UNIVERSITAT POLITÈCNICA DE VALÈNCIA

_ TELECOM ESCUELA TÉCNICA VLC SUPERIOR DE INGENIERÍA DE TELECOMUNICACIÓN

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería de Telecomunicación

Enmascaramiento de voz en oficinas abiertas

Trabajo Fin de Grado

Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación

AUTOR/A: Marti Canet, Evarist

Tutor/a: Diego Antón, María de

Cotutor/a externo: FUSTER CRIADO, LAURA

CURSO ACADÉMICO: 2021/2022

Resumen

El propósito de este trabajo consiste en el estudio, análisis y posterior desarrollo de un sistema capaz de mitigar ruidos ambiente que, en entornos de trabajo, hospitales, etc, puede ser perjudicial para el normal desarrollo de las actividades laborales. Este sistema se centra principalmente en el enmascaramiento de las conversaciones o voces que el usuario es capaz de entender. Estas conversaciones pueden afectar a la concentración de los trabajadores y también puede verse reducida la propia privacidad de esta. La solución propuesta se basa en el uso de sonidos que puedan enmascarar la señal vocal reduciendo su inteligibilidad mediante el empleo de sonidos enmascaradores que pueden ser ruidos de banda ancha o sonidos de la naturaleza. El sistema implementado consta de un conjunto de micrófonos y altavoces que capturan las señales de voz de fondo, las procesan y emiten una respuesta acorde a ciertos parámetros para enmascarar las conversaciones de alrededor y reducir completamente su inteligibilidad, intentando siempre mantener un volúmen controlado en los altavoces para no perturbar el entorno de trabajo .De forma dinámica se calcula el nivel de sonido enmascarador empleando el parámetro STOI para calcular la inteligibilidad de las señales de voz.

Resum

El propòsit d'aquest treball consistix en l'estudi, anàlisi i posterior desenvolupament d'un sistema capaç de mitigar sorolls d'ambient que, en entorns de treball, hospitals, etc, pot ser perjudicial per al normal desenvolupament de les activitats laborals. Aquest sistema s'ha centrat principalment en el camuflament de les conversacions. Aquestes conversacions poden afectar a la concentració dels treballadors i també pot veure's reduïda la pròpia privacitat d'aquesta. La solució proposada es basa en l'ús de sons que poden enmascarar la señal de veu reduïnt la seua intel·ligibilitat mitjançant sons enmascaradors que poden ser sorolls de banda ampla o sons de la naturalesa. El sistema consta d'un conjunt de micròfons i d'altaveus que capturen les senyals de veu de fons, les processen i emeten una resposta d'acord a certs paràmetres per a enmascarar les conversaciones d'alrededor i reduir completament la seua intel·ligibilitat, intentant sempre mantenir uns nivells mínims necessaris de volum en els altaveus per no pertorbar l'entorn de treball. S'ha fet ús d'un parámetre nomenat STOI que calcula la intel·ligibilitat de les senyals de veu.

Abstract

The purpose of this work consists of the study, analysis, and subsequent development of a system capable of mitigating background noise that, in work environments, hospitals, etc, can be detrimental to the normal development of work activities. This system mainly focused on masking conversations around you with pleasant, natural sounds. These conversations can affect on the concentration of the workers, and somebody can also reduce privacy. The proposed solution consists of a system of microphones and loudspeakers that capture that capture the background voice signals, process them and emit a response according to specific parameters to mask surruonding conversations and reduce their intelligibility. Always trying to maintain the minimum necessary volume levels in the loudspeakers so as not to disturb the work environment. A parameter called STOI is used, which calculates the intelligibility of the voice signals.

Índice

I

Capítulo 1. Introducción

1.1 Introducción general al proyecto a desarrollar

Este proyecto surgió con intención de dar respuesta a la necesidad de respetar y proteger tanto la privacidad de una conversación entre dos o más individuos, como el confort cuando estos se encuentran en ciertos espacios abiertos donde se requiere un nivel de privacidad como oficinas abiertas, sucursales bancarias o salas y habitaciones en hospitales. En los mencionados entornos puede darse el caso de que se esté hablando sobre temas que no incumben al resto de trabajadores en una oficina, o se necesite concentración para que un trabajador realice su trabajo sin la necesidad de escuchar voces de fondo que le desconcentren de sus labores, o se esté hablando sobre ciertos detalles financieros de un cliente en el mostrador de la sucursal y cuyos detalles no deben ser escuchados por el resto de clientes de la entidad bancaria. Incluso el diagnóstico de un paciente que no debe ser conocido por el resto de personas que se encuentran en una sala o habitación de hospital. Es por ello que en este proyecto propone el desarrollo de un sistema que, capturando mediante elementos como micrófonos las señales de voz de la conversación que se desea mantener privada o ininteligible para los demás, este sea capaz de generar una señal acústica agradable en la posición donde se encuentren los individuos ajenos a la conversación, para así reducir por completo la inteligibilidad de una manera confortable. Esta señal acústica es nada menos que un sonido enmascarador. En la figura 1 se muestra un esquema del sistema propuesto, en donde se muestra cómo será el sistema para obtener una idea más clara y concisa:

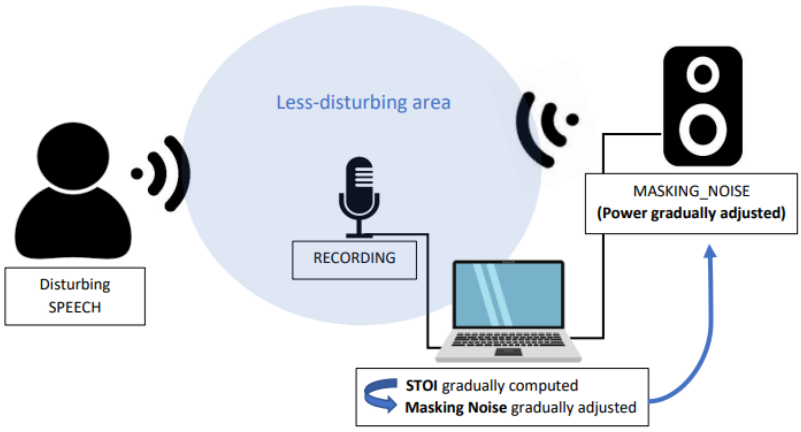

Fig 1. Esquema del sistema de enmascaramiento de voz

El micrófono etiquetado como recording se sitúa en el área que se quiere controlar y en la que quiere reducirse la inteligibilidad del habla generada en otra zona, el sistema emplea esta señal para calcular el STOI y, con el valor de dicho parámetro generar un sonido enmascarador que será emitido por el altavoz para contrarestar dicha señal de voz y conseguir que sea ininteligible para el usuario. Se pueden encontrar algunas soluciones comerciales que abordan el mismo problema. Las más sencillas de ellas tan solo consiguen enmascarar el sonido y reducir por completo la inteligibilidad sin tener en cuenta lo molesto que pueden llegar a ser los niveles de sonido generado. Algunos ejemplos de las soluciones comerciales más completas son: Sonet Qt (https://cambridgesound.com/products/sonet-qt/) , Soft-dB (https://www.softdb.com/sound-masking/) , Lonestar Acoustics (https://lonestaracoustics.com/) o Atlas IDE (https://www.atlasied.com/speech-privacy-solution).

1.2 Cronograma del trabajo

En este apartado vamos a ver cada una de las fases que se fueron siguiendo desde que se eligió este tema como mi proyecto de final de carrera a desarrollar hasta el montaje final que se realizó y el testeo práctico del funcionamiento del sistema. Aunque se van a abordar puntos que se verán en capítulos posteriores, se va a realizar de una manera muy breve con el único objetivo que el lector obtenga un esquema claro de qués lo que se va a ir explicando y ampliando en los próximos capítulos. También es objetivo clarificar cómo se aborda un proyecto de este estilo. A continuación, se va a ir desarrollando subapartado tras subapartado en este capítulo las tareas que se fueron llevando a cabo siguiendo la cronología empleada. Repito, de una manera muy breve. Es probable que el orden cronológico de las tareas difiera en algunos puntos del orden que se sigue en los capítulos. Esto se debe a que al realizar el proyecto, uno se va dando cuenta de que falta extenderse en algún punto anterior o incluso realizar algún estudio de pasos previos puede resultar interesante. En la memoria se sigue un orden de los capítulos para que el lector tenga una idea total muy clara sobre el proyecto, desde los cimientos en los que se apoya hasta los aspectos más técnicos del sistema que se ha construido, formando una especie de pirámide invertida, es decir, yendo de los aspectos más amplios y generales hasta los detalles más técnicos. Se muestra un pequeño diagrama con los 4 puntos principales que se han tenido que desarrollar. Tras la figura sigue una breve explicación de cada uno en distintos subapartados:

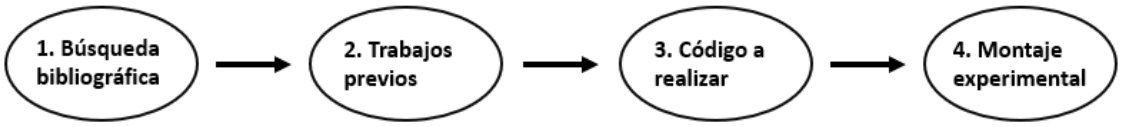

Fig 2. Cronograma seguido en el proyecto

1.2.1 Búsqueda bibliográfica

La búsqueda bibliográfica fue un primer paso muy guiado por parte de la tutora y cotura que me han ayudado a realizar este TFG. Se me proporcionó una serie de documentos y artículos de revistas científicas cuyas referencias están presentes en la bibliografía de la memoria. Como los aspectos que se trabajan en este proyecto son bastante novedosos, no hay disponible gran cantidad de documentos ni otros elementos bibliográficos. A mi se me presentó dos artículos de revista distintos. En uno se habla de un parámetro que cobra gran importancia durante todo el proyecto. Los autores de dicho artículo publicaron de manera abierta tanto el documento del artículo como el código de MATLAB para calcular dicho parámetro a partir de unos *inputs* que se requieren. Dicho documento lo podemos ver referenciado en [1].

Por otra parte también se me proporciona otro artículo, publicado por investigadores y profesores de la propia UPV en donde se realiza un estudio de la posible aplicación de dicho parámetro para aplicaciones de *masking sounds* y la relación del parámetro y distintos sonidos de enmascaramiento que se puedan emplear. Para ello se realiza también un test a varias personas. Podemos ver dicha referencia en [2].

Finalmente, durante el desarrollo de este proyecto, se hacen uso de distintas técnicas y entender la teoría es fundamental para una correcta aplicación de dichas técnicas, mi trabajo por esta parte fue investigar más a fondo y encontrar alguna clase de elemento bibliográfico que me pudiese ayudar a entender completamente cualquier aspecto, técnica o idea que pudiese llevar a cabo durante el trabajo. Aunque como es de esperar cuando se realiza un trabajo de esta índole, la búsqueda de información es lo primero que se realiza, pero no se termina de hacer hasta que se termina todo el TFG, debido a que se van descubriendo y cambiando ideas o aspectos aplicados al proyecto a medida que se profundiza.

1.2.2 Trabajos previos

En anterioridad en esta memoria se ha dicho que lo que se pretende realizar en este proyecto es algo novedoso, para el desarrollo del sistema se ha partido de artículos y trabajos previos que sientan las bases de lo que va a ser el futuro sistema. Uno de ellos es el comentado en el subapartado anterior, en donde se dice que junto con el artículo, se entrega un código MATLAB para el cálculo del parámetro, conocido como STOI. Este código ha sido un trabajo previo a este proyecto y que se ha tenido que integrar en el código principal. Como dicho código, también se me proporcionaron por parte de la cotutora de este proyecto una serie de subrutinas que pueden ser necesarias para realizar ciertos cálculos durante la ejecución del código principal del sistema. Este es trabajo previo que se me entregó, mi función en este paso fue estudiar a fondo dichas subrutinas y pensar cómo las voy a ir integrando en el código principal.

Finalmente, una serie de trabajos previos que se tuvieron que realizar anterior al montaje del prototipo son una serie de test a distintos individuos que más adelante se explican detalladamente y que siguen la línea de investigación acerca del STOI y los sonidos enmascaradores que se realiza en [2], pero aplicados al entorno y desarrollo del prototipo que voy a realizar.

1.2.3 Código a realizar

El código forma una parte fundamental del trabajo a realizar en este proyecto, aunque desgraciadamente es la parte que se mantiene más oculta para el lector de esta memoria o para el público en general que vaya a utilizar el sistema que se desarrolla en este trabajo. Aunque explicar la rutina principal y subrutinas del código y todos los otros códigos no es un objetivo en este apartado de la memoria, lo que se pretende es dar una idea de cómo está estructurado dicho código y, cuáles son las tareas fundamentales que debe realizar este código para llevar a cabo el correcto funcionamiento del sistema. Cabe resaltar que todo el software del sistema está desarrollado en MATLAB. A continuación se presenta un diagrama con las tareas principales que debe realizar el software del sistema para su funcionamiento, paso a paso y, de una manera muy general para que el lector tenga una idea de las funciones que realiza el software sin la necesidad de que este conozca por completo el código:

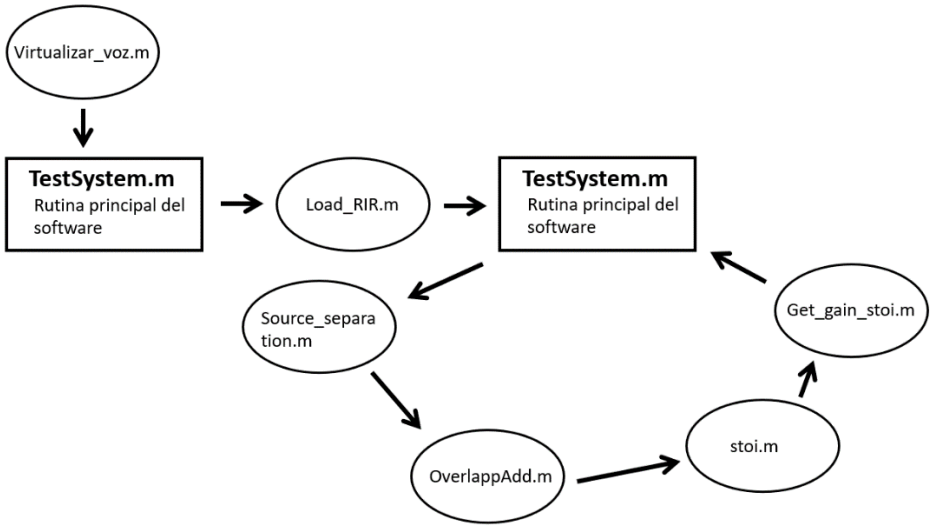

Fig 3. Diagrama de flujos del software empleado para el funcionamiento del sistema

Aunque más adelante se va a estudiar con mucha mayor profundidad cada proceso que interviene en el propio sistema, se va a explicar de manera breve el funcionamiento del sistema con los pasos que se muestran en el anterior diagrama:

- Virtualizar_voz.m: Esta función que antecede a la rutina principal es la encargada de generar las llamadas funciones de observación, un tipo de variable que actúa a modo de filtro y es necesaria para llevar a cabo lo que se conoce y, en capítulos posteriores se estudiará, proceso de virtualización.

- TestSystem.m: Es la rutina principal del código, en ella se desarrolla casi la totalidad de los procesos de procesado de las señales, extracción de la información más relevante. etc. En esta primera aparición lo que realiza es la declaración de las constantes y variables más importantes así como los búfferes de las señales que se vayan capturando.

- Load_RIR.m: Esta función es la encargada de cargar las respuestas impulsivas calculadas y almacenadas cuando se realizó el montaje del sistema. Son necesarias para el cálculo de ciertas operaciones de procesado de las señales.

- TestSystem.m: La segunda vez que aparece la rutina principal en el diagrama es para indicarnos que genera un bucle que termina cuando se apaga el sistema o ya no existe más sonido ambiente que enmascarar o conversaciones externas cuya privacidad deba respetarse. Dentro de este bucle se realizan las operaciones más importantes que desempeñan el papel más fundamental para el funcionamiento del sistema.

- Source_separation.m: Realiza la resta acústica para obtener la señal de voz limpia para su posterior comparación.

- OverlappAdd.m: El solape suma es la técnica que se utiliza a modo de convolución en dominio frecuencial y en teimpo real.

- Stoi.m: Esta función es la encargada del cálculo del parámetro STOI en cada frame.

- Get_gain_stoi.m: A partir del valor del parámetro STOI calculado anteriormente, esta función genera una ganancia a aplicar por el sistema de altavoces que generan el sonido enmascarador

Como se ha comentado anteriormente, la finalidad de contar el código en este capítulo introductorio es para crear una idea general de lo que se va a ir contando durante los próximos capítulos. Efectivamente se va a contar y trabajar con mucho más detalle cada uno de los aspectos e ideas empleadas en el código, siguiendo por supuesto una evolución de cómo desde la idea más primitiva se llegó hasta el desarrollo final de dicha técnica o función junto con el desarrollo y posterior conclusión.

1.2.4 Montaje a realizar

El montaje del sistema se realizó en el laboratorio de audio del insituto del GTAC (Grupo de Tratamiento de Audio en Comunicaciones), que forma parte del iTEAM (Insituto de Telecomunicaciones y Aplicaciones Multimedia). El objetivo de este subapartado es nombrarlo como punto de gran importancia en el cronograma del trabajo, ya que es la parte en donde se prueba de manera física el funcionamiento del sistema a nivel práctico y se obtienen los resultados de este en tiempo real, tal y como funcionaría si se instalase como solución tecnológica en uno de los entornos para los que fue pensado este sistema. Más adelante se muestran tanto los dispositivos y elementos que van a componer este sistema como una explicación muy detallada sobre cómo se debe realizar el montaje, cómo interactuan los distintos elementos entre sí, fotografías de este, etc.

1.3 Breve descripción de los próximos capítulos

El objetivo de este apartado es únicamente dar a conocer de manera general qué es lo que el lector va a ver en los próximos capítulos. No se abordará en ningún punto, tan solo se nombrará lo que se o se trabaja en cada capítulo. Ahora pasamos a resumir brevemente qué se estudiará en cada capítulo:

- Capítulo 2: En este capítulo se van a nombrar y explicar las técnicas, ideas y aspectos teóricos en los que este proyecto se basa. Forman una parte fundamental del sistema. Es necesario conocer lo que se cuenta en este capítulo porque durante el desarrollo del sistema, tanto en sus primeras fases como en el montaje final, se hace uso de los conocimientos que en este capítulo se describen.

- Capítulo 3: En este capítulo se trabajan los aspectos y técnicas más prácticas que también sientan los cimientos del prototipo. También se puede ver en este capítulo cómo se van a aplicar algunos de los conceptos teóricos vistos en el anterior capítulo de manera práctica en el sistema. Se va a describir dos estudios en forma de test en los que se llegan a conclusiones de cómo aplicar dichos conocimientos teóricos al sistema.

- Capítulo 4: En este capítulo ya se entra en detalle sobre el desarrollo del prototipo y de los primeros pasos. Lo que se pretende realizar en este tema es desarrollar completamente el sistema en un entorno puramente simulado, es decir, a través de la herramienta de MATLAB. Cobra una gran importancia realizar primero el sistema en un entorno de simulación porque es fundamental para darse cuenta de cómo debe estructurarse el sistema, si se necesitan algoritmos adicionales y, además permite escalar el sistema en complejidad. Otra función muy importante es simular cómo el medio en que se encuentra puede afectar al rendimiento o prestaciones del sistema.

- Capítulo 5: Una vez en el capítulo 4 se han llegado a los conocimientos necesarios y se ha estudiado el sistema en profundidad para optimizar al máximo sus prestaciones, lo que se pretende en este capítulo es ver que el sistema deba de funcionar sea cual sea el entorno en el que se encuentre. La causalidad y la no causalidad pueden tener grandes efectos en el sistema. Es por ello que en este capítulo se desarrolla un estudio para conocer cómo debe comportarse el sistema para que rinda con muy buenas prestaciones sea el sistema causal o no causal. Aquí se sigue manteniendo el sistema en simulación.

- Capítulo 6: Este capítulo ya aborda el desarrollo y construcción del sistema en un entorno real, haciendo uso activamente de componentes y elementos físicos. Lo que se pretende es observar si el sistema sigue manteniendo las prestaciones que se observaron en simulación. En este tema final del trabajo se desarrollan nuevas técnicas necesarias para completar el sistema y que funcione en tiempo real y con las mismas prestaciones que se obtuvieron en simulación. Se evoluciona en complejidad y finalmente se realiza un estudio de los resultados obtenidos y las prestaciones.

Capítulo 2. Aspectos teóricos que se trabajan en el proyecto

2.1 Captación y enmascaramiento de sonidos

El enmascaramiento de sonido es una técnica por el cual se genera un sonido, normalmente ruido blanco o rosa, con el objetivo de enmascarar otro sonido que está presente en el mismo momento. En nuestra aplicación intentaremos enmascarar el ruido ambiente. Basado en el enmascaramiento auditivo, dista de ser una técnica de control activo del ruido como lo es la cancelación de sonido, el enmascaramiento también es capaz de reducir parcial o totalmente la percepción de dicho sonido no deseado. Es aplicado en una zona concreta con el objetivo de mejorar la percepción acústica. En el caso del proyecto se traduce en la reducción de la inteligibilidad de una conversación presente en el entorno con el objetivo de mantener la privacidad, consiguéndose así un enfoque más individual y mejorando la productividad en entornos de trabajo como oficinas abiertas.

El enmascaramiento del sonido significa un control sobre los ruidos presentes en un entorno desarrollado. A continuación se expone una figura en donde se puede ver de una manera más esquemática la definición de enmascaramiento de sonido:

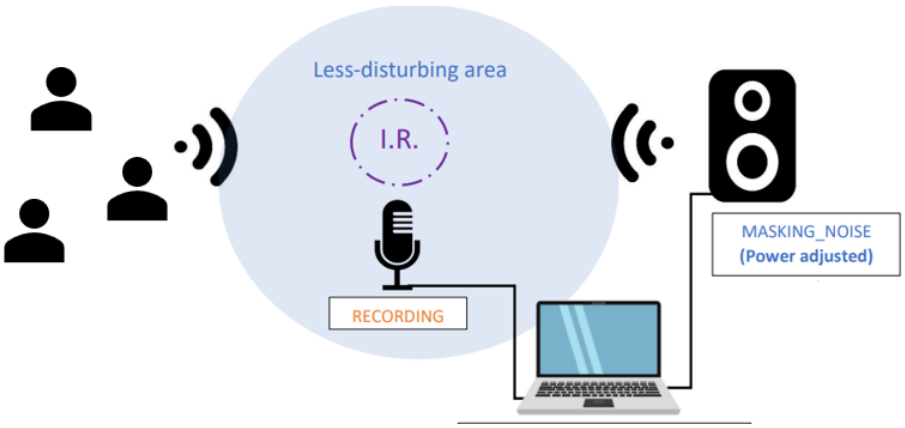

Fig 4. Esquema del enmascramiento de sonido en el sistema

En esta figura podemos observar que área azul etiquetada como *Less-disturbing area* es la zona en donde se desea enmascarar el ruido ambiente provocado por las conversaciones de los individuos presentes a la izquierda. En dicha área tenemos un micrófono que capta dicho ruido ambiente y este es procesado mediante ordenador, para una posterior generación de sonido de enmascaramiento emitido por el altavoz de la derecha para reducir la perpepción del ruido molesto provocado por individuos hablando en este caso. Como se viene explicando en esta memoria desde el inicio, no solo es el reducir la percepción del ruido, sino más bien reducir la inteligibilidad de dichas conversaciones presentes alrededor de la zona de trabajo.

2.2 El parámetro STOI

Como bien se ha explicado anteriormente, el objetivo fundamental es reducir por completo la inteligibilidad de una conversación privada en los puntos en donde se encuentren otros individuos ajenos a dicha conversación. Es por ello que el sistema debe analizar mediante ordenador la inteligibilidad de la conversación en cada instante de tiempo para poder generar una respuesta acorde al nivel de inteligibilidad. Aquí es donde se presenta un parámetro, conocido como STOI (*Short-Time Objective Intelligibility Measure*), que muestra la correlación existente entre la señal

de voz pura estimada por el sistema y la señal capturada por los distintos micrófonos presentes en el entorno. Este parámetro se propuso durante el 2011 y lo podemos ver en [1] (Taal, 2011).

Como el STOI, han surgido otros parámetros que intentan medir la inteligibilidad de las señales de voz de otras formas. El hecho que en este proyecto se haya decantado por el uso del STOI se debe a dos factores. El primero es que es uno de los más actuales que existen hoy en día, se puede leer una breve introducción a la historia de los parámetros que miden la inteligibilidad en el primer apartado de [1] (Taal, 2011). El segundo factor y más importante de los dos, es que existe una función de MATLAB totalmente libre, creada y distribuida por los autores del parámetro STOI, por lo que simplifica el trabajo a realizar. Basta con entender cómo funciona dicho código.

El parámetro STOI está basado en tramas de la señal de voz muy cortas, de unos 386 ms. El algoritmo va realizando el cálculo del STOI comparando dos tramas de dicha longitud y además con un solape de un 50% entre tramas contiguas. Entonces busca la correlación entre las tramas de las dos señales que compara y genera un valor de STOI que computa la inteligibilidad de la conversación presente en el entorno. A continuación se presenta un pequeño diagrama de lo que se viene explicando en este apartado:

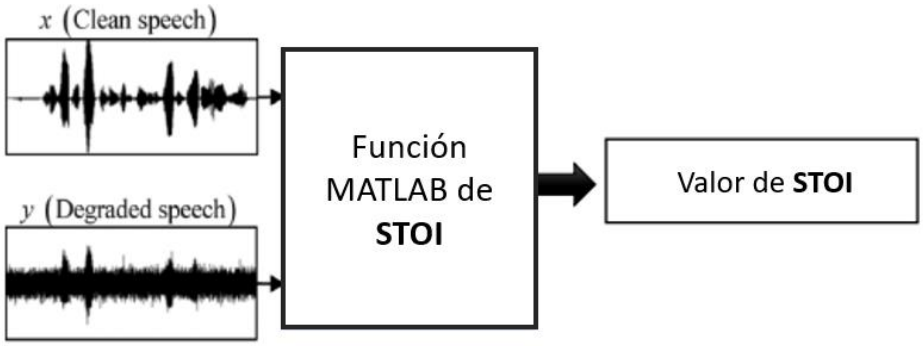

Fig 5. Diagrama de bloques de la generación del parámetro STOI

En la figura, las señal denotada como *X (Clean Speech)* hace referencia a la señal de voz pura que sirve de referencia y la denotada como *Y (Degraded Speech)* hace referencia a la suma total de señales, sonidos y ruidos presentes en el entorno y capturados por una de las partes *hardware* del sistema, es decir, los micrófonos.

2.3 Umbral STOI

En cuanto al valor de STOI que computa la inteligibilidad, el código genera un valor de STOI que se mueve en una escala entre 0 y 1. Siendo 1 la inteligibilidad más absoluta y 0 lo totalmente opuesto. Entre medias existen unos rangos que se mueven entre inteligibilidades muy malas e inteligibilidades muy buenas.

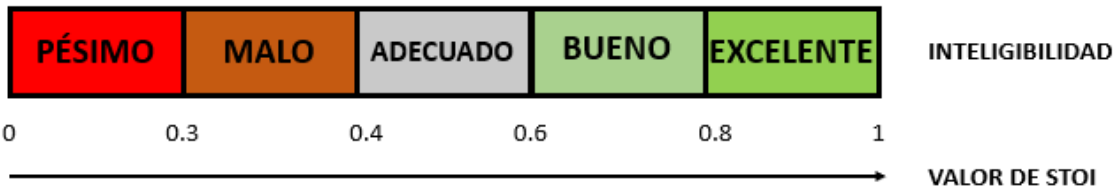

Fig 6. Rango de valores del STOI

La anterior figura muestra un pequeño gráfico en donde se resume brevemente entre qué valores de STOI se considera que la inteligibilidad de la conversación que se analiza es buena o mala, se puede ver de manera más detallada en (Taal, 2011). En dicho artículo también se explica que existe un valor de STOI en el cual, para valores superiores a este la señal de voz que nos llega se puede entender en mayor o menor medida y que para valores inferiores la conversación es totalmente ininteligible. A este valor se le conoce como umbral del STOI. Dicho valor umbral es de gran importancia conocerlo, pues cualquier sistema que se implemente cuyo objetivo sea reducir la inteligibilidad, como es el caso de este proyecto, debe generar una respuesta acústica que reduzca el valor de STOI en el punto en donde se encuentra el individuo que no debe entender nada de la conversación a este valor como mínimo. En nuestro caso, dado que en los entornos en donde el prototipo del sistema va a instalarse se requiere también de un estado de confort para no resultar molesto, es vital conocer este valor pues la misión del sistema es que el STOI converja a este valor.

2.4 Técnica del micrófono remoto

La técnica del micrófono remoto o virtualización del micrófono consiste en la estimación de la respuesta impulsional auditiva que se tendría desde una fuente de sonido hasta un punto cualquiera calculado gracias a unos micrófonos auxiliares fijos, conocidos como micrófonos de monitorización. Con un conocimiento del entorno en donde se va a desarrollar el sistema mediante estimación de las respuestas imulsionales, se puede conseguir el efecto de tener físicamente un micrófono en la posición deseada. Aunque realmente este micrófono no exista en dicha posición. Por ello se le conoce como micrófono virtual o punto de virtualización [3](Danielle Moreau, 2008). Respecto a la referencia anterior, en dicho artículo se comenta la técnica de virtualización o micrófono remoto para *active noise control* o lo que es lo mismo cancelación de ruido, pero este caso tiene la novedosa particularidad que se está utilizando para enmascaramiento de señales de voz.

El efecto de virtualización lo vamos a conseguir mediante un algoritmo. Este algoritmo genera una matriz [P*M x V] llamada función de observación. Las dimensiones de esta función de observación se forman mediante los valores de P, M y V que hacen referencia a:

- P: número de coeficientes o muestras que se desean para la función de observación
- M: número de micrófonos de monitorización
- V: número de micrófonos virtuales, que en nuestro caso es siempre 1.

Cuando se implanta el sistema por primera vez en un entorno, al igual que se instalan los micrófonos de monitorización, también se instala un micrófono en la posición en donde se va a virtualizar. Se deben capturar en todos los micrófonos una señal que se emite a través de las fuentes de sonido para posteriormente correlarla junto con las señales capturadas por los micrófonos de monitorización y así obtener la función de observación.

La función de observación transforma la señal recibida por los micrófonos de monitorización en lo que recibirían los micrófonos virtuales. A lo largo de la memoria, también se refiere a la función de observación simplemente como filtro, ya que se usa un modelo FIR de orden P para modelar dicha función de observación. Se calcula también la solución con mínimos cuadrados.

La respuesta impulsiva que se obtendría desde cualquiera de las fuentes que emitan señales acústicas hasta el micrófono virtual se consigue filtrando la respuesta hacia el micrófono de monitorización junto con la comentada función de observación. Si tuviéramos más de un micrófono de monitorización, la respuesta virtual se obtendría como suma de las respuestas de cada uno de los micrófonos de monitorización filtradas por la función de observación.

Finalmente comentar que alrededor del punto que se quiere virtualizar existe una zona cuyo margen de error es asumible como para dejar una ligera libertad de movimiento alrededor de dicho

punto, por tanto se ha creado una *less-disturbing area* alrededor del oído, siendo el punto donde mayores prestaciones se consiguen el propio punto donde se encuentra el micrófono de virtualización.

Capítulo 3. Elementos prácticos necesarios

3.1 Test del umbral de STOI

El oído y aparato auditivo de cada humano tiene ciertas diferencias respecto al de otro humano. Tanto la edad, como el género, como la genética de cada persona hace que este sentido sea distinto entre diferentes personas. También lo es el umbral de STOI de cada persona. No todos tenemos un umbral de STOI igual, hay quienes son capaces de entender ciertas conversaciones mientras que para otros son totalmente ininteligibles. Aunque en un oído sano y joven, las diferencias entre los umbrales de STOI no son tan acusadas como lo podrían ser entre una persona jóven y una persona muy anciana, cuyo sistema auditivo ha perdido muchas prestaciones.

Fijar un umbral de STOI para cada individuo en cada entorno donde el sistema de este proyecto sea requerido es tarea imposible. La solución pasa por estimar empíricamente mediante un test con diferentes jurados entre los que se logre una paridad en cuanto a género y rango de edades, para obtener un valor medio de umbral de STOI que pueda englobar lo mejor posible todas las posibles opciones. Este test es un programa desarrollado en MATLAB en donde se presentan distintas pruebas. En cada una de las pruebas hay un sonido distinto que enmascarará una señal de voz que se escucha desde el inicio de cada prueba. Mediante unos botones de (+) y (-) se aumentará o reducirá el nivel de señal del sonido enmascarador respectivamente. La idea consiste en ir ajunstando dicho nivel hasta que la señal de voz que se escuchaba deje de entenderse completamente sin la necesidad de llegar a niveles molestos o dañinos para el individuo. Un buen punto de referencia podría ser el momento en que puedes escuchar la voz, pero dejas de entender las palabras. Es entonces cuando la prueba finaliza y dicho valor de STOI es calculado. Así es como se calcula cuál es el valor del umbral de STOI para cada sonido de enmascaramiento y, para cada individuo entrevistado. En la siguiente imagen se deja una captura de la interfaz del programa que obsevan los jurados entrevistados en esta prueba.

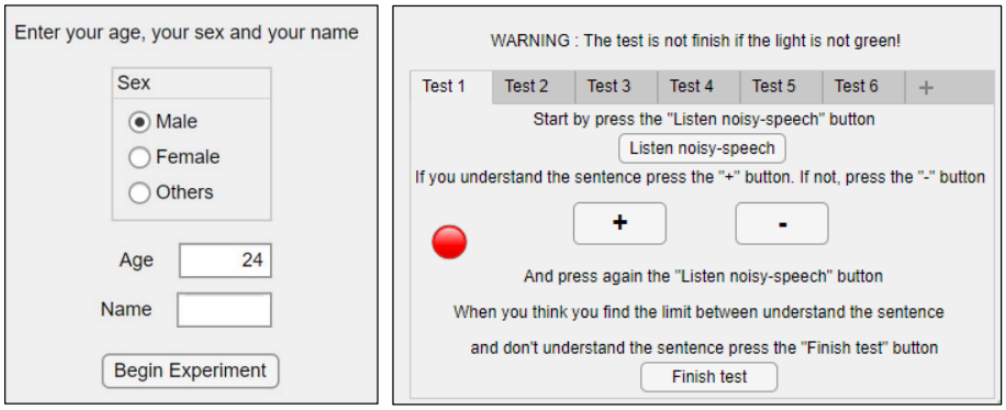

Fig 7. Interfaz gráfica del test de umbral del STOI

Hay que aclarar una cuestión importante. Para este test se han usado 6 tipos distintos de sonidos, filtrados para eliminar altas frecuencias y cuyo RMS, en el caso de señales de audio el RMS hace referencia a la sonoridad media dentro de una ventana, sea idéntico en todos los casos, para evitar que haya sonidos con valores de energía mucho más grandes que otros. Se realiza de este modo para que todos los sonidos tengan iguales condiciones en términos de potencia o energía y no sean condicionantes en la elección por parte del jurado. En apartados posteriores puede que se realicen estudios con otros sonidos distintos, siempre dependiendo de la naturaleza de lo que se esté analizando en cada caso.

Volviendo a los resultados del test, tras encuestar a unas 20 personas, vamos a representar los valores de STOI para 6 tipos distintos de sonidos. Se representará mediante un gráfico de caja y bigotes, ya que con este tipo de gráfico podemos ver la varianza en el STOI calculado a partir de los resultados de las pruebas. Los sonidos enmascaradores presentes en el test son los siguientes: *Babble Noise* que corresponde con el sonido que se escucharía en un bar lleno de gente, *Violin Noise* se corresponde con un fragmento de una pieza tocada con violín, *Pink Noise* o ruido rosa, *Water Noise* que se asemeja al sonido que hace el agua al bajar por un sumidero, *Relax Noise* es un fragmento de un sonido relajante perteneciente a un vídeo de música relajante de YouTube y finalmente, *m5 Noise* que se corresponde con un ruido rosa filtrado en donde a mayores frecuencias la pérdida energía en cada componente es mayor. La pendiente en la densidad espectral de potencia es de -5dB.

A continuación se representa el gráfico con los valores de STOI para cada tipo de sonido:

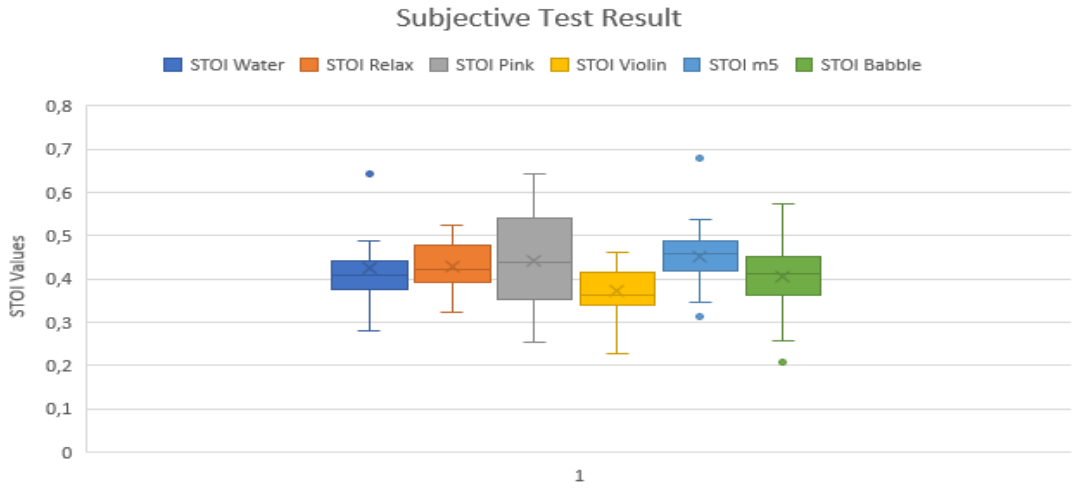

Fig 8. Gráfico de caja y bigotes con los resultados del test de umbral del STOI

A primera vista podemos ver que el valor medio del umbral de STOI para cada tipo de sonido está entre los 0.35 y 0.45, un valor bastante razonable según (Fuster, L., 2021). Otro aspecto a tener en cuenta es la desviación estándar que presentan los distintos sonidos. En general, la desviación típica para sonidos de enmascaramiento como el ruido rosa suelen ser menores como se pueden observar en [2], en parte se debe al tipo de jurado que se ha entrevistado para la realización del test. Mientras en [2] se entrevistó a expertos del campo de la acústica, en este test el objetivo a sido testear a usuarios convencionales, con el objetivo de estudiar los valores umbral para el público general ya que cualquier persona podría ser usuaria en un futuro de este tipo de sistemas de enmascaramiento de señales de voz para preservar la privacidad y un estado de concentración en puestos de trabajo. Por ende, es razonable que la desviación típica de todos los sonidos sea mayor.

Otra cuestión interesante a comentar es que el umbral del ruido de pendiente de -5dB es ligeramente superior que el resto. Este sonido no concentra tanta energía en frecuencias altas, por lo que su densidad espectral se asemeja más al rango del habla humana, especialmente en varones. Es por ello que el valor umbral para este ruido de enmascaramiento es mayor que el resto.

3.2 Estimación de los caminos de propagación en el laboratorio

Este proceso es de gran importancia en el desarrollo del proyecto y es fundamental explicar cómo se modelan los caminos o respuestas impulsionales desde las diferentes fuentes de sonido, voz y ruido enmascarador, hasta los diferentes puntos de medición. Para tanto el proceso de simulación y estudio inicial del entorno en que se basa el trabajo como para el prototipo en tiempo real, el

obtener previamente las respuestas impulsionales es elemental para filtrar y convolucionar tales respuestas con las distintas pistas de audio para emular al máximo el efecto que se produciría en un entorno real.

En este apartado se pretende explicar cómo se han obtenido las respuestas impulsionales y se comentará cada una de las 4 diferentes respuestas, de sendas fuentes a sendos micrófonos, que se han estudiado en el proceso de simulación. Para obtener la respuestas, se requiere de dos procesos. Uno físico que consiste en realizar el montaje adecuado, en la posición adecuada, de los elementos vitales que conforman el ecosistema del proyecto, micrófonos y altavoces que realizarán la función de fuentes de ruido y voz. El otro proceso consiste en emular mediante software la respuesta impulsional hacia cada punto de estudio basándose en algoritmos. Se muestran a continuación los equipos más importantes que componen este sistema:

Fig 9. Altavoces JBL y micrófonos Behringer ECM8000

Fig 10. Tarjeta de audio STUDIO-CAPTURE y cables XRL

La tarjeta de audio es la encargada de controlar tanto los micrófonos como los altavoces, recibiendo la señal que se captura a través del micrófono y enviando la señal emitida por los altavoces. También es el nexo entre el IDE de MATLAB y el resto del equipos que componen el sistema. A través de MATLAB y mediante las entradas de la tarjeta, el sistema mapea todas las respuestas impulsivas que más tarde serán utilizadas para posteriores análisis y para el propio funcionamiento del sistema. Tanto los altavoces como los micrófonos están conectados con la tarjeta de audio mediante los cables XML.

Dado que al extrapolar este sistema a numerosos casos según la posición que ocupan tanto los micrófonos de monitorización como las fuentes de voz y ruido, se pueden dar casos de causalidad y no causalidad de las respuestas a la hora de realizar la virtualización. Es por ello, que se ha decidido estudiar las respuestas impulsionales que se obtendrían en escenarios diferentes,

especialmente los que puedan ocasionar causalidades y no causalidades. El objetivo es obtener resultados robustos e independientes de la ubicación de los micrófonos y altavoces.

También se ha realizado el proceso de virtualización para las respuestas impulsionales en las que interviene la fuente de voz. La razón de proceder de esta manera es por filtrar también la señal de *speech* con el objetivo de comparar las gráficas de valores de STOI y ganancias para comparar con las obtenidas haciendo uso de la respuesta impulsional hasta el punto físico ideal.

Otra cuestión importante es el *reshape* que se ha tenido que realizar de las 3 respuestas impulsionales para que la función *getObservation* sea capaz de generar correctamente la función ideal para virtualizar. El propósito del *reshape* es reordenar las muestras de las respuestas impulsionales para que la función *getObservation* reciba los datos de manera que interprete que hay un altavoz y tres micrófono. No es una solución única. Durante la evolución del proyecto, se van puliendo aspectos en el código para facilitar y optimizar la estima de respuestas.

La función *getObservation* va a ser la encargada de generar la función ideal que virtualizará la respuesta impulsional de virtualización.

3.3 Test de preferencia de sonidos

Se ha implementado un test para determinar qué sonidos de enmascaramiento son preferibles para escuchar en entornos de trabajo. Como bien se ha ido mencionando a lo largo de esta memoria, el sistema debe ser capaz de enmascarar las conversaciones que hay alrededor para que se vuelvan ininteligibles manteniendo el confort de los usuarios. Es por ello que elegir qué sonidos para el enmascaramiento va a utilizar el sistema es una tarea fundamental. Aunque más adelante en el proyecto veremos un estudio sobre las prestaciones a la hora de enmascarar distintos sonidos en cuanto a su naturaleza, es de vital importancia conocer los sonidos que, en general, el usuario prefiere escuchar cuando se encuentra en un entorno de trabajo por ejemplo, uno de los entornos pensados para este prototipo. Para ello se ha diseñado un test de pares muy simple en donde la gente va a elegir qué sonido prefiere sobre otro.

El test va a contar con 6 sonidos de distinto tipo, los que se probaron en [2] salvando alguna diferencia. Van desde sonidos “muy molestos” a sonidos “relajantes”. Los sonidos que componen este test son:

- *Babble Noise:* Este sonido corresponde a conversaciones entre varias personas en las que no se entiendo lo que dicen. Un murmullo constante.
- *Pink Noise:* Es un ruido caracterizado porque su densidad espectral es inversamente proporcional a la frecuencia. Su percepción es similar al ruido blanco, salvo que la particularidad que los diferencia es que las componentes en alta frecuencia del ruido rosa son mucho menores en cuanto a energía.
- *Water Noise:* Este es el sonido que emite el agua al caer por un sumidero, o el sonido que emite el agua fluyendo a través de un arroyo.
- *Violin Noise:* Como su nombre indica, es el sonido de una pieza tocada con un violín. La pieza es de un *tempo* bastante lento y con notas largas.
- *Relax Noise:*Es una combinación de sonidos de la naturaleza junto con instrumentos musicales y que suele utilizarse como música relajante para descansar o estudiar.
- *M5 Noise:* Es el ruido rosa filtrado con una pendiente de -5dB.

Estos sonidos son comparados entre sí en un total de 18 rondas. En cada ronda se comparan 2 de los sonidos y el jurado debe escoger qué sonido prefiere escuchar en un entorno de trabajo, en donde la concentración y el confort son clave para el desarrollo normal de sus labores. Los sonidos van alternándose y cambiando de orden a la hora de escucharlos para que la consistencia en la elección de los participantes siempre sea la más alta posible. Es decir, si se escoge un sonido como más confortable sobre otro, que al cambiarlos de orden el primero que se escogió siga siendo la eleccion en la otra ronda. O si un sonido A se prefiere sobre uno B, y dicho sonido B se prefiere

sobre uno C, por ende el sonido A se prefiere sobre el C. El test trata de trabajar estos aspectos de consistencia y repetitividad en la elección de los sonidos. Para que ningún factor que no sea lo agradable que le parezca al participante un sonido u otro influya en el resultado de la prueba.

A la izquierda podemos apreciar que el participante debe escribir algunos datos personales. La única razón por la cual se pregunta es, como en el test del umbral, obtener la mejor paridad posible en cuanto a género y rango de edades de los participantes del test. A la derecha se observa la interfaz que se encuentra el jurado al realizar el test. Podemos ver que hay dos botones, el A y el B, cuando se pulsa uno de estos botones se reproduce uno de los sonidos previamente comentados y el participante marca su preferencia en las casillas que hay encima de los botones. Al finalizar el test, se observan unos porcentajes referidos a repetitividad y consistencia, y son indicadores de la aleatoriedad en la elección del jurado en cuestión. Valores muy bajos significa que se han marcado las casillas aleatoriamente y por tanto, dicho test sería descartado.

Tras la realización del test a un grupo de 20 individuos, veamos cómo han quedado los resultados globales:

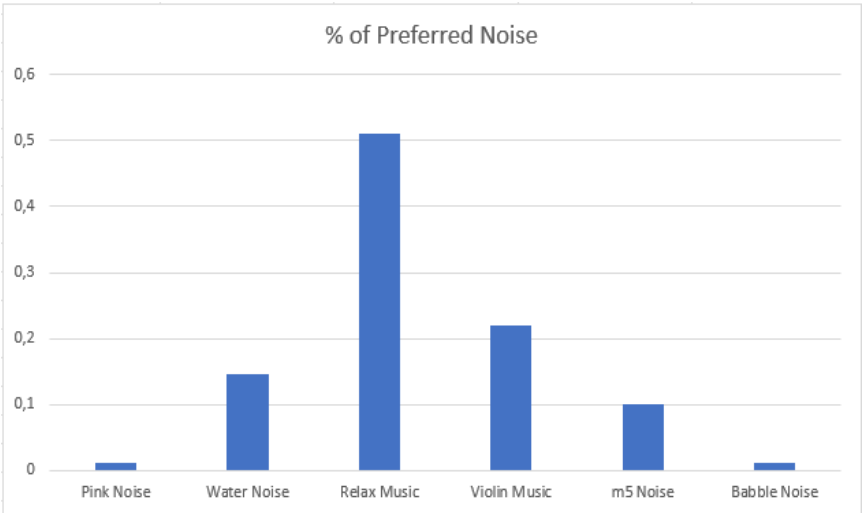

Fig 11. Probabilidad de elección de sonidos enmascaradores

Esta gráfica nos muestra la probabilidad de elección de cada sonido. Podemos observar que hay un sonido que predomina sobre el resto de manera abrupta. La música relajante es el sonido preferido por los participantes en cuanto a sonido que desearían escuchar en su entorno de trabajo, por sus características, su ritmo lento y suave. La probabilidad de escoger dicho sonido es superior que la probabilidad de ecoger cualquiera de los otros en conjunto, es decir, más de un 51%. Le sigue el sonido del violín con un 22% que también es bastante relajante y adecuado para un entorno de concentración como es el trabajo según los jueces del test. En cuanto al ruido rosa y al tumulto, son los peores sonidos para concentrarse según los participantes. Tan solo representan un 1.2% cada uno del total. Finalmente comentar una diferencia interesante que se puede observar en este estudio. El hecho de filtrar las altas frecuencias del ruido rosa como se ha filtrado con el ruido de pendiente de -5dB hace que a la gente le resulte más placentero a la hora de escuchar en su entorno de trabajo, pasando del 1.2% a un 10%.

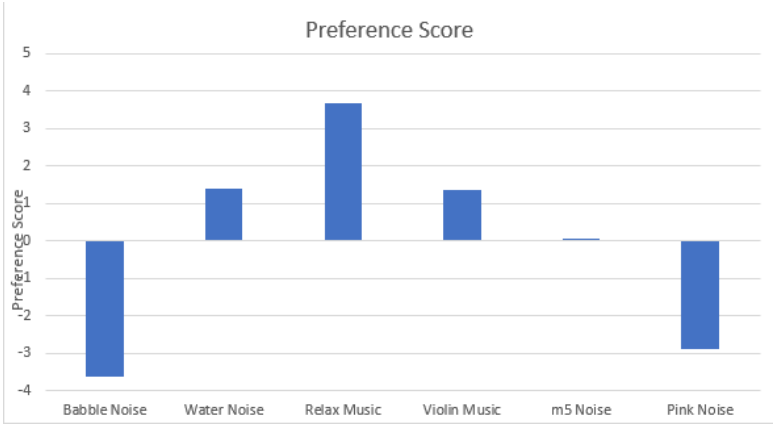

Fig 12. Preferencia de sonidos enmascaradores

Esta gráfica nos muestra la preferencia de cada sonido. El orden de preferencia es igual que si viésemos la gráfica de probabilidad de elección de sonidos salvando que en esta se puede apreciar de manera más clara que entre los peores sonidos para entornos de trabajo según la gente, *Babble Noise* y *Pink Noise*, el ruido del tumulto de la gente es menos preferido que el ruido rosa. En cuanto a la concordancia entre las elecciones entre los distintos participantes, el test arroja un 55%, siendo un 100% de concordancia que la elección de todos los participantes ha sido la misma y un 0% que ningún participante coincide con ningún otro. Podemos calificar los resultados del test como fiables en dicho aspecto.

Capítulo 4. Desarrollo del prototipo en simulación y de los distintos pasos

El objetivo del prototipo en simulación es simular y estudiar todo lo que se implementará en el prototipo real. La intención es controlar y entender todos los aspectos que involucra el prototipo, como son el STOI, una parámetro que se va a utilizar como herramienta para calcular la inteligibilidad *[1] (Cees H. Taal, 2011),* las ganancias que se van a aplicar a la fuente de ruido así como la intelegibilidad del speech tras la emisión del ruido enmascarador por parte de la propia fuente de ruido. En una fase inicial del proceso, se requieren los caminos de propagación hacia los distintos micrófonos se han modelizado previamente mediante un algoritmo LMS. En cuanto a la señal de voz, se ha modelizado como un campo lejano, de manera que se perciban con niveles de amplitud similares en casi todos los puntos.

Para un mayor análisis detallado del problema, se ha dividido en 3 pasos, denotados como *Paso 1*, *Paso 2* y *Paso 3.* A su vez, el segundo paso se ha subdividido en otros dos pasos más, *2A* y *2B.* La idea de dividir el proceso de desarrollo del código de simulación es estudiar de manera más profunda la intelegibilidad final de la voz en diferentes puntos aplicando distintas posiciones para varios micrófonos, en donde se va a estudiar cómo se ve afectado el valor del STOI en cada punto o micrófono al aplicar una ganancia a la fuente de ruido que se calcula a partir del STOI en un punto. Como el propósito final es observar cómo el ruido enmascara la voz haciendo que la conversación de fondo sea muy difícil de entender desde la posición en donde se encuentra la cabeza y oídos de una persona en su puesto de trabajo, se va a estudiar siempre desde la posición que ocuparía el oído de una persona sentada en su puesto de trabajo. Para una mayor comparación entre el efecto ideal que se conseguiría teniendo un micrófono en la posición del oído humano y la que se consigue con la técnica del micrófono remoto, también llamada virtualización, se ha optado por colocar un micrófono en dicha posición, para realizar la comparación entre ideal y real.El micrófono ideal se llama micrófono ideal y el real de virtualización. Aunque se irán dando detalles de esta última idea a medida que se avance con el estudio y desarrollo del prototipo.

El objetivo del primer paso es sentar las bases de lo que va a ser el prototipo, primero mediante simulación para después implementar en tiempo real. Analizando los elemenos más sencillos que intervienen en el diseño del sistema. Se va a estudiar el efecto sobre el "micrófono de monitorización", que hace referencia al micrófono que se encuentra en el ordenador del entorno de trabajo o, en la parte de delante del oficinista. Este micrófono está pensado para que sea un micrófono conectado al PC de trabajo o que sea el propio micro que llevan integrados algunos monitores e incluso podría ser un *wearable* o micrófono de solapa en el propio usuario.

4.1 Paso 1: Estudio del funcionamiento del parámetro STOI, de la ganancia y del sistema.

El objetivo principal de este paso que antecede al verdadero propósito del estudio del entorno de simulación es estudiar y observar cómo se comporta el código stoi.m *[1] (Cees H. Taal, 2011),* cómo genera este los valores del STOI en función del sonido que capta el micrófono de control y del sonido generado por la fuente de voz. Otro aspecto importante de estudio es el efecto de la ganancia que se aplica a la fuente de ruido y que es generada a partir del código llamado get_gain_stoi.m.

El código stoi.m es el encargado de calcular, como se ha comentado anteriormente, el valor del STOI en cada trama a partir del sonido (voz + ruido enmascarador) captado por el micrófono de control junto con la voz generada a partir de la fuente de voz. Mediante distintos procesos comentados en *[1] (Cees H. Taal, 2011),* este es capaz de generar un valor de STOI en donde 1 significa inteligibilidad total y 0 es lo opuesto.

Un punto importante a comentar es el hecho de cómo se ha gestionado la información almacenada de todos los sonidos almacenados. Dado que el código stoi.m no trabaja muestra a muestra porque

sería inviable, ya que se requeriría de un tiempo de análisis muy grande. El análisis de la inteligibilidad de una conversación humana analizada muestra a muestra como la almacena MATLAB es impracticable, debido a que se necesitaría analizar como unidad mínima un vocablo, sílaba o palabra corta, que ocupan cientos y cientos de muestras, para analizar la inteligibilidad. En ese modo, lo ideal es que se trabaje con tramas de varias muestras. En el caso del código stoi.m y del paso1.m, se opta por trabajar con búfferes de 1s o lo que es lo mismo, 10 veces el tamaño del bloque (4096 muestras y, un solape de 9 bloques). Con 1s ya se puede analizar el STOI de una manera más razonable a efectos del habla humana.

En cuanto al análisis del *Paso 1*, el objetivo como se ha comentado anteriormente es visualizar cómo evolucionan los parámetros de STOI, ganancia en este caso de control y las diferentes pistas de audio que componen todo el entorno del sistema, como son el propio sonido de la voz sin enmascarar, el ruido de enmascaramiento y el sonido que detectaría finalmente el micrófono de control (voz+ruido de enmascaramiento). Para una mejor comprensión de la estructura del problema, vamos a visualizar el siguiente esquema:

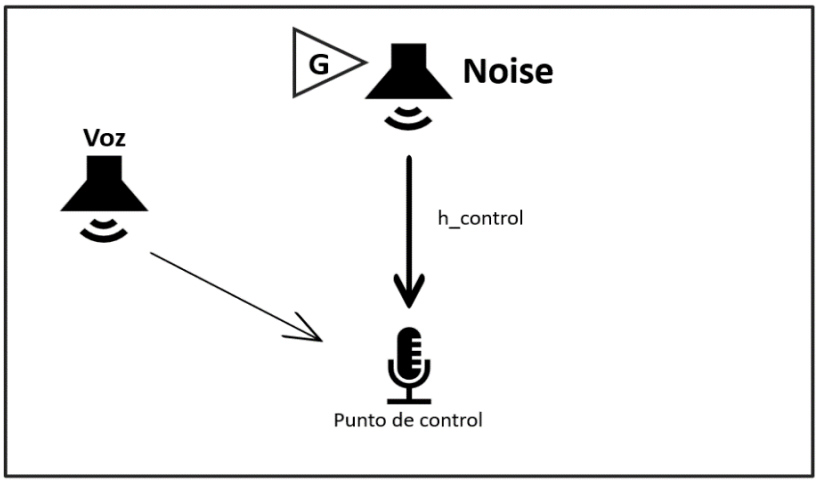

Fig 13. Esquema de los elementos que componen el primer Set-up

Una mayor comprensión al inicio del desarrollo y explicación del sistema en esta memoria es fundamental, por ello se presenta también a continuación otro esquema no tan conceptual de lo que es el sistema:

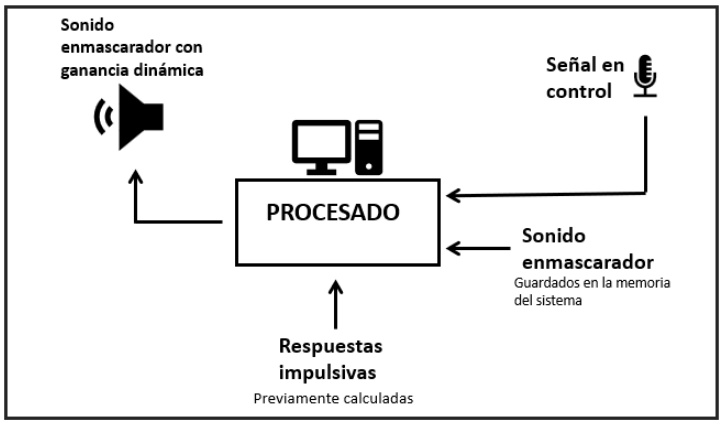

Fig 14. Diagrama del sistema

Uno de los objetivos principales en todos los pasos es calcular el nivel del parámetro STOI para calcular la ganancia dinámica del sonido enmascarador. El micrófono de control, al igual que los

micrófonos que se van a implementar en pasos posteriores, recopilan la suma aditiva de la señal de voz junto con el sonido enmascarador. Sin embargo, el código para calcular el STOI precisa tanto del sonido capturado por el micrófono como por el speech previo al enmascaramiento, algo que el micrófono es incapaz de capturar. La solución que se plantea es estimar la señal del speech sabiendo que el sistema conoce el modelado del camino de propagación, el sonido de enmascaramiento y la ganancia que aplica a la fuente de ruido. A continuación se representa matemáticamente cómo se va a estimar la señal de speech:

$$y_M = s(n) + [h * (n(n) \cdot G_M)] \quad (1.1.1)$$

$$\hat{s} = y_M - [\hat{h} * (n(n) \cdot G_M)] \quad (1.1.2)$$

$$stoi_{value} = stoi(\hat{s}, y_M) \quad (1.1.3)$$

Siendo y_M la señal capturada por el micrófono, h y $\hat{h}$ el modelado de los caminos de propagación y el modelado estimado respectivamente y finalmente, $s(n)$ y $\hat{s}(n)$ las señales de voz y voz estimada respectivamente. Una vez calculado todo lo necesario, el programa ya es capaz de calcular el valor del STOI.

Una vez analizado cómo se calcula el parámetro STOI, debemos de enfocarnos en cómo va a afectar a la ganancia que se va a aplicar en cada momento a la fuente de ruido. El sistema debe ser capaz de asignar un cierto valor de ganancia en función del STOI cada 100ms aproximadamente para que este converja en el valor umbral prefijado en base a los resultados de los test realizados y comentados en capítulos anteriores. En dicho estudio se consideran valores entre 0.35 y 0.4 como valores umbrales en los que la inteligibilidad de la voz se pierde casi por completo.

El propósito de esta parte es analizar qué valores de ganancia se deben aplicar a la fuente de ruido en cada instante para que el parámetro de STOI converja al valor umbral asignado. También se debe tener en cuenta que debido a que el proyecto está pensando para aplicarse en entornos de trabajo, se debe tener en cuenta que la ganancia no debe ser excesivamente grande y su crecimiento debe ser gradual puesto que se debe de mantener un entorno adecuado para el trabajo. Por este motivo el código get_gain_stoi.m, que calcula la ganancia en base al valor de STOI, introduce un factor de olvido para evitar que se apliquen cambios muy bruscos de ganancia.

El funcionamiento del código get_gain_stoi.m es bastante simple. La idea es que se aplique una cierta ganancia a la fuente de ruido para que este enmascare la señal de speech y consiga reducir el valor del STOI hasta que finalmente converja en el umbral fijado. Para valores de STOI muy alejados del umbral, se deben aplicar incrementos de ganancia más elevados que cuando rondan valores cercanos al umbral. Por eso, se debe aplicar unos valores de incremento de ganancia distintos según si el valor de STOI se encuentra en un rango o en otro.

4.1.1 Análisis de las granancias de get_gain_stoi.m en función de la potencia del sonido enmascarador

La ganancia adicional que debemos aplicar en cada rango de valores del STOI para una convergencia más rápida en el umbral de STOI debe ser asignada en función de la potencia del "sonido" enmascarador. Esto se debe a que el sonido cuya potencia es muy alta, su convergencia al umbral STOI es muy rápida pero el volúmen que la fuente de ruido aplica para contrarestar la señal de voz es demasiado alto y molesto para entornos de trabajo.

Las siguientes pruebas y observaciones que debemos realizar es cuáles van a ser esos umbrales de potencia que van a determinar los valores adicionales de ganancia de get_gain_stoi que se van a

aplicar. Lo más recomendable para realizar esta investigación es haciendo uso del código test_STOI_sim.m, que hace referencia a lo que hemos denominado en primera instancia como paso 1, ya que la estructura es más simple. Solo disponemos de una ganacia a calcular, de un único punto de control y de una respuesta impulsional previamente estimada.

En la siguiente tabla se va a representar la potencia de la señal medida desde el micro de control para los diferentes *Masking sounds* :

Masking Noises	***Power Values (dB)***
Babble Noise	-36.1030
Music Noise	-36.4852
Wind Bird Noise	-36.8745
Pink Noise	-37.8277
Forest Noise	-55.5248
Natural Water Noise	-37.1353
Driving In The Rain Noise	-32.1548
Whistle Bird Noise	-41.2502

Tabla 1. Valores de potencia en dB de los sonidos enmascaradores

Hay que tener en cuenta que estos valores de potencia hacen referencia a un montaje en particular, aunque nos podemos hacer una idea de las características que va a tener cada señal y que, aunque se refiera un montaje, no va a haber una gran diferencia entre otros. El de mayor potencia va a seguir siendo de mayor potencia y probablemente más molesto y, el menos potente va a seguir siendo menos potente. No tiene que ver con la ganancia con la que se emite a través del conjunto de altavoces, sino la potencia del archivo de audio donde se almacena el sonido enmascarador.

Vemos que hay algunos sonidos cuya potencia es muy alta, por ejemplo *Driving In The Rain.* El sonido que provocan los coches al pasar por la carretera un día lluvioso puede ser efectivo a la hora de reducir completamente la inteligibilidad de una conversación, cuyo objetivo es mantener la privacidad en espacios abiertos como podría ser una oficina, pero puede resultar también molesto cuando estamos trabajando y necesitamos concentrarnos.

Una vez realizado este primer estudio sobre las diferentes potencias de los *Masking Noises* y cómo nos pueden llegar a afectar en nuestro entorno de trabajo, pasaremos a ver cómo evoluciona el parámetro STOI a lo largo de la señal de voz cuando enmascaramos el ruido con distintos sonidos y cómo de rápido tiende al umbral de inteligibilidad. La idea principal es asignar diferentes rangos de variaciones de la ganancia y los intervalos en donde se aplican en función del grado de convergencia y de la potencia del sonido enmascarador. Sabemos que los sonidos cuya potencia es mayor suelen ser más molestos para las personas cuando estamos trabajando.

Para sonidos cuya potencia es mayor, vamos a aplicar valores de variación de ganancia menores en donde las mayores variaciones se apliquen en un intervalo de valores de STOI más alejado de lo que lo haríamos con sonidos con baja potencia. En estos últimos, la variación de ganancia que se le aplica a los intervalos de valores de STOI más alejados del umbral fijado van a ser mayores.

Para estudiar la convergencia del parámetro STOI en las mismas condiciones, sin tener en cuenta la potencia, se van a estudiar los casos más extremos y algún caso que se ajuste al término medio, cuando el algoritmo asigna la misma variación de ganancia.

Veamos cómo evolucionan los valores de STOI para los casos más extremos: *Driving In The Rain* y *Forest Noise*, y para casos medios: *Music Noise y Natural Water Noise*:

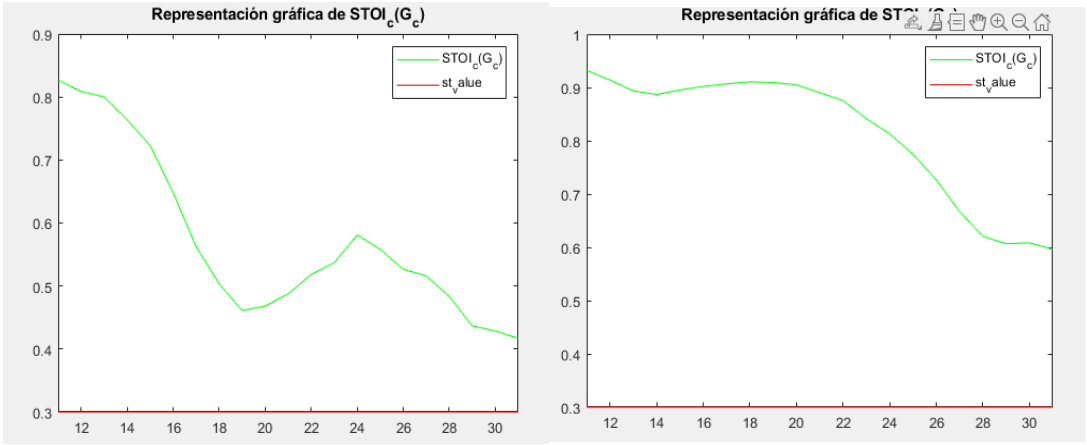

Fig 15. Valores de STOI para *Driving In The Rain* (izquierda) y *Forest Noise* (derecha)

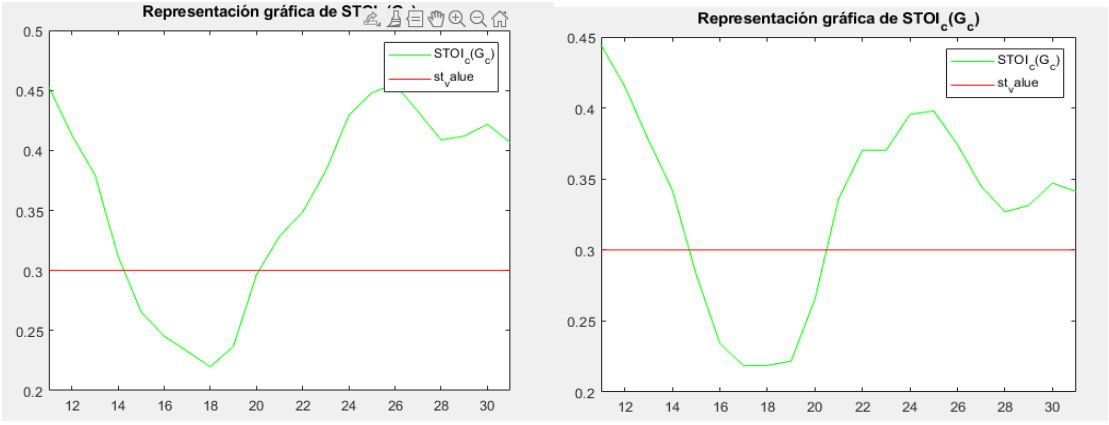

Fig 16. Valores de STOI para *Music Noise* (izquierda) y *Natural Water Noise* (derecha)

Apreciamos que en algunos casos, al aplicar diferentes sonidos enmascaradores observamos que el algoritmo es incapaz de hacer que el parámetro STOI se acerque al objetivo o umbral de inteligibilidad, denotado en las gráficas como *st_value*, a medio plazo. En algunos casos debido a que tanto las variaciones de ganancia como el intervalo de valores de STOI no están ajustadas a los valores óptimos o cercanos a este para que se den las condiciones de convergencia en estos casos. La solución pasa por aplicar distintas variaciones de ganancia y distintos intervalos donde aplicarlos para cada sonido enmascarador, según el nivel de potencia de los sonidos enmascaradores. Dado el número de sonidos, de ahora en adelante nos vamos a quedar con los siguientes para objeto de estudio: *Forest Noise*, *Music Noise* y *Wind Bird Noise.*

Una cuestión que se percibe al estudiar la forma de las gráficas de los diferentes sonidos de enmascaramiento es el hecho de que muy pocas veces el valor de STOI queda por debajo , en gran medida, del umbral definido (tan solo cuando el valor de STOI es próximo al valor umbral a converger y en cuyo caso la variación tanto por arriba como por abajo es idéntica). Por tanto podríamos dejar unos valores de variación de ganancia genéricos para cualquier tipo de sonido, indiferentemente de la potencia que posean. Se puede dar el caso en momentos en los que se dan silencios entre palabras, pero suele darse en ínfimos intervalos de tiempo, en los que el sistema tampoco es capaz de obtener valores de STOI inferiores a 0.3 durante períodos de tiempo a considerar. En cuanto a los valores superiores, se han establecido 4 diferentes intervalos, según el nivel de potencia de los sonidos de enmascaramiento, en donde se han declarado diferentes valores tanto de variación de ganancia como de rangos a aplicar según el valor de STOI, así como ganancias máximas y mínimas que el sistema puede soportar. Dicha configuración queda de la siguiente forma:

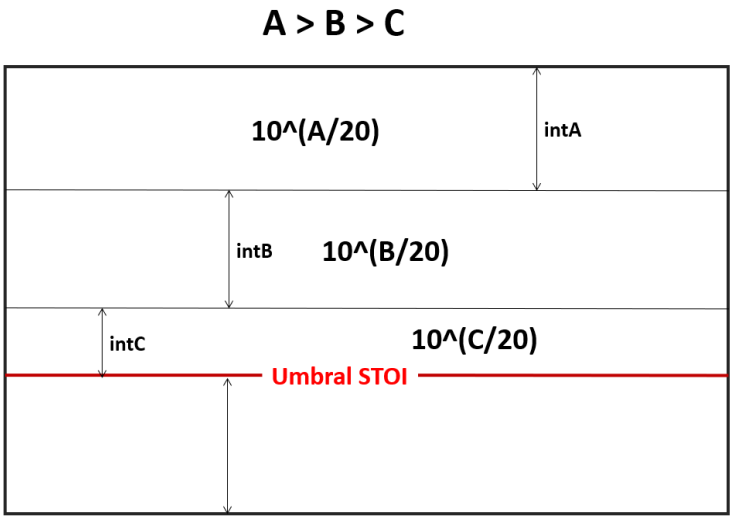

Fig 17. Representación de la variación en ganancia según el rango de STOI

En la figura 17 podemos apreciar una serie de letras e intervalos. Las letras A, B y C hacen referencia a las ganancias, en lineal, que se aplican en cada intervalo de niveles de STOI. Los distintos intervalos denotados como intA, intB e intC hacen referencia al rango de nivel de STOI en donde se aplica cierto valor de ganancia. Tras esta representación esquemática se encuentra el algoritmo *get_gain_stoi* que calcula la ganancia a aplicar en cada momento según el valor del parámetro STOI. A parte de generar una variación de la ganancia mayor o menor en función del STOI, también presenta un factor de olvido, que consigue que no hayan variaciones abruptas de la ganancia que generen repentinos golpes de sonido muy fuertes que puedan distraer al sujeto que se encuentre en su área de trabajo. Este factor de olvido tiene como objetivo que si en el entorno pasa de haber silencio a tener una conversación molesta o que se requiera de privacidad, el sistema actúe gradualmente hasta enmascarar en pocos segundos totalmente la señal de voz, quedando dicha conversación ininteligible. A continuación vemos cómo trabaja el sistema visualizando y analizando los valores y señales que se muestran (*NOTA* para los ejemplos siguientes se está estudiando haciendo uso de la señal de sonido enmascarador *MusicNoise*):

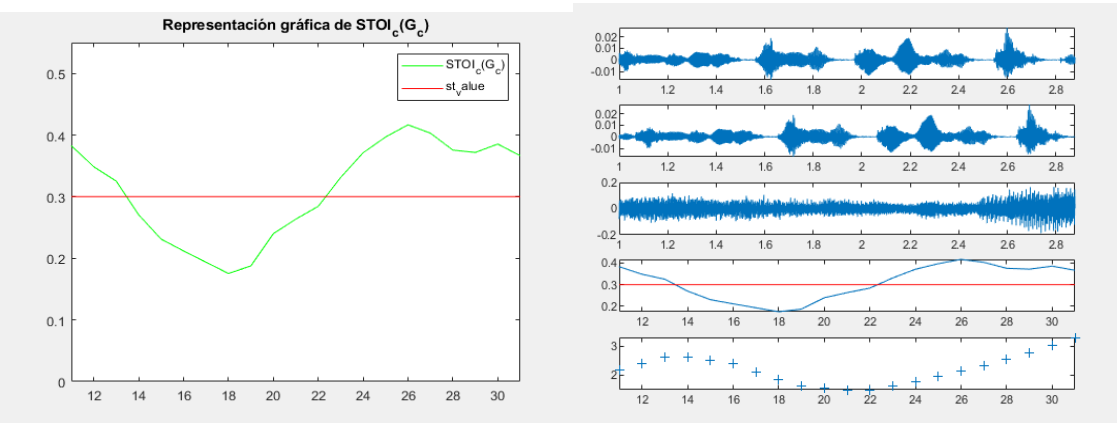

Fig 18. Valores de STOI (izq) y gráfica de las señales y ganancias que se obtienen a lo largo de la simulación (dcha)

La gráfica izquierda presenta buenos resultados ya que observamos que a los pocos segundos de duración, el sistema consigue que el valor de STOI se mantenga cercano al umbral (*st_value*=0.3). Por otra parte, en la figura de la derecha observamos cómo el sistema rinde con buenas prestaciones,

no llegando a saturarse ni tampoco emitiendo con una gran ganancia como podemos ver en la gráfica inferior de la derecha. Las señales digitales se encuentran bien acotadas en valores correctos. Una observación que se realiza es que si aumentamos ligeramente el valor del umbral de STOI a un cierto valor en donde siga siendo ininteligible, podemos apreciar que el sistema converge de la misma manera con casi la mitad de la ganancia precisada por la fuente. Podemos verlo a continuación:

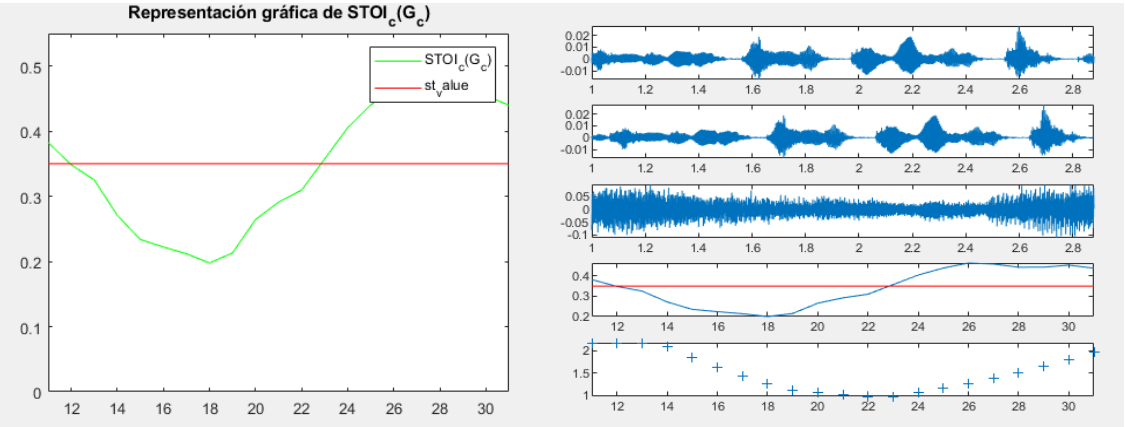

Fig 19. Valores de STOI y señales que se obtienen al aumentar ligeramente el umbral STOI

Ahora podemos comprobar como en la figura izquierda, hay menos puntos en los que el sistema se sature. La saturación del sistema se observa cuando en la gráfica inferior de la derecha, todos los valores de ganancia quedan en la parte superior de la gráfica formando una línea recta la mayor parte del tiempo. Paralelamente veremos como el STOI en ese período en donde la ganancia satura no llega a estabilizarse alrededor del umbral del STOI, sino que lo hace en un valor de STOI superior, no reduciendo por completo la inteligibilidad de las señales de voz del entorno. Aumentando ligeramente el valor umbral de STOI sin que este tenga grandes repercusiones en la inteligibilidad final vemos que la ganancia puede reducirse hasta casi la mitad. Otra posible medida podría haber sido que el factor de olvido sea muy alto, pero hubiese generado picos puntuales de saturación, generando paupérrimos valores de calidad de experiencia para las personas.

Para terminar el estudio de la parte del sistema encargado de estimar la ganancia, se explica a continuación cómo se ha calculado la potencia. La fórmula matemática utilizada para el cálculo de la potencia de un ruido consiste en el sumatorio del cuadrado de todas las muestras que lo componen al cuadrado y dividido entre el número total de muestras. La expresión se presenta a continuación:

$$P = \frac{\sum_{i=1}^{N} |x_i|^2}{N} \qquad (1.1.4)$$

Siendo N el número total de muestras del array que almacena la señal en cuestión.

4.2 Paso 2: El micrófono ideal

El paso que sucede consiste en aplicar un segundo micrófono que se sitúan en uno de los oídos del *dummy* de pruebas que simula una persona sentada en su puesto de trabajo en una oficina. La idea es simular el efecto auditivo que va a experimentar una persona sentada y de cómo se va a enmascarar la señal de speech para que a la persona le sea muy difícil entender el hilo de la conversación.

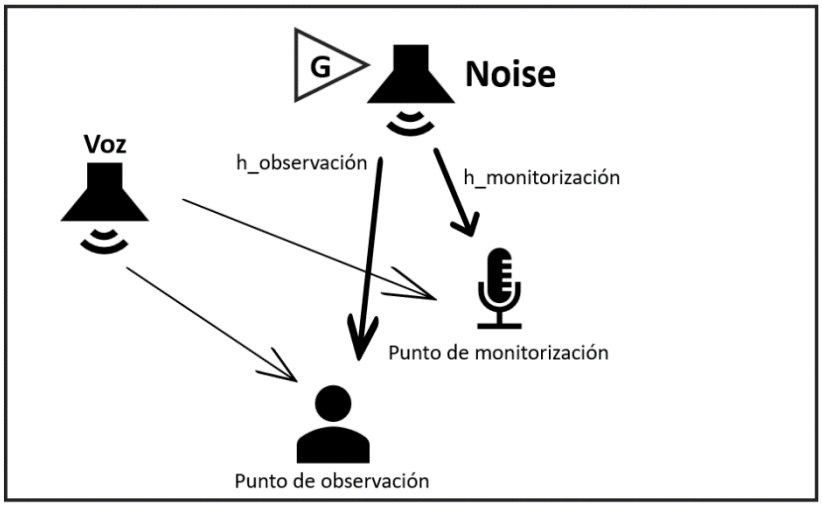

Fig 20. Esquema de los elementos que componen el segundo Set-up

En este paso, el estudio de la inteligibilidad se centra en el micrófono ideal, es decir, el micrófono que se va a colocar simulando la posición del oído de la persona. Por ello, el objeto de estudio en esta parte será obtener los valores de STOI y ganancia a aplicar a la fuente de ruido en la posición de un oído del usuario. Cabe destacar que la posición del micro ideal está más alejada que la ocupada por el micro de control (que a partir de ahora pasará a llamarse de monitorización) estudiado en el paso anterior. Es lógico pensar que para enmascarar la voz, dadas sus características, la ganancia a aplicar a la fuente de ruido debe ser mayor puesto que la distancia entre la fuente y el micro virtual es mayor. Las ecuaciones a tener en cuenta son las siguientes:

$$y_{Mo} = s(n) + [h_{Mo} * (n(n) \cdot G_{Mo})] \quad (1.2.1)$$

$$y_{ideal} = s(n) + [h_{ideal} * (n(n) \cdot G_{ideal})] \quad (1.2.2)$$

$$\hat{s} = y_{Mo} - \left[\widehat{h_{Mo}} * (n(n) \cdot G_{Mo})\right] \quad (2.3)$$

$$\hat{s} \approx y_{ideal} - [h_{ideal} * (n(n) \cdot G_{ideal})] \quad (1.2.4)$$

$$stoi_{Mo} = stoi(\hat{s}, y_{Mo}) \quad (1.2.5)$$

$$stoi_{ideal} = stoi(\hat{s}, y_{ideal}) \quad (1.2.6)$$

Ahora intervienen dos ganancias distintas, la que se calcula con el STOI del micrófono de monitorización y la que se calcula a partir del STOI en la posición del nuevo micrófono. Las aproximaciones en la ecuación (1.2.4) se deben a que conociendo la señal que se captura en el micrófono y la ganancia y sonido enmascarador, se estima la señal original de voz. El propósito es el estudio de todos los parámetros y valores en todos los puntos objeto de estudio. Anteriormente visualizábamos el STOI en MATLAB como un array, en donde en cada posición de dicho array almacenábamos el valor del STOI en cada instante analizado. Para este nuevo paso, se necesita

conocer el valor de STOI en las dos posiciones. Otra cuestión importante es que en este punto, al igual que estamos estudiando la evolución de la señal de voz junto con el sonido enmascarador capturado en cada uno de los micrófonos, también disponemos de dos ganancias diferentes generadas tanto en la posición del micrófono de monitorización como en la posición del micrófono ideal. Por esta razón, se plantean en total 4 valores de STOI diferentes:

$STOI_{monitorización}(G_{monitorización})$	$STOI_{monitorización}(G_{ideal})$
$STOI_{ideal}(G_{monitorización})$	$STOI_{ideal}(G_{ideal})$

Tabla 2. Los distintos STOIs según la ganancia que se aplica en este segundo Set-up

En cuanto al propio código de MATLAB, generar el código del paso 2 ha sido muy similar al del paso 1. Se ha tenido que duplicar todas las variables y funciones declaradas ya que ahora tenemos captura de muestras en dos micrófonos situados en dos lugares distintos. La parte más complicada de implementar es la parte del código en donde se calculan las expresiones desde *(2.1)* hasta *(2.6)*.

Dado que la posición de más interés es la posición del oyente , lo mejor es estudiar y visualizar los parámetros de STOI en el micro ideal cuando se aplican las ganancias de monitorización y ideal respectivamente. Vamos a visualizar una gráfica en donde podemos observar claramente cómo el valor de STOI en la posición del oyente, cuando se aplica la ganancia de monitorización que es siempre menor que la ideal, queda por debajo del valor de STOI cuando a la fuente de ruido se le aplica la ganancia de la posición del oído:

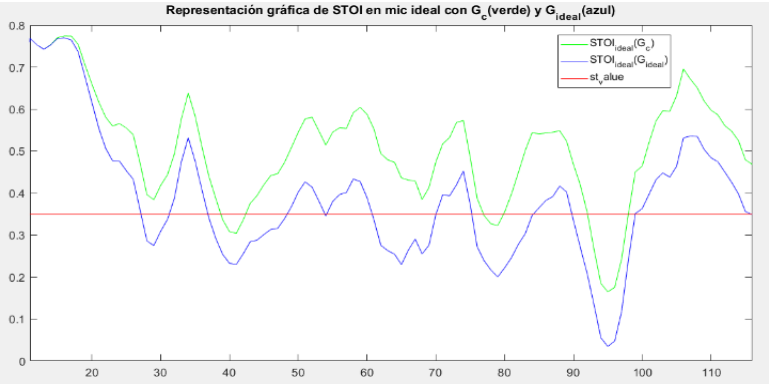

Fig 21. Valores de STOI en el micrófono ideal aplicando la ganancia calculada en monitorización y en el punto ideal

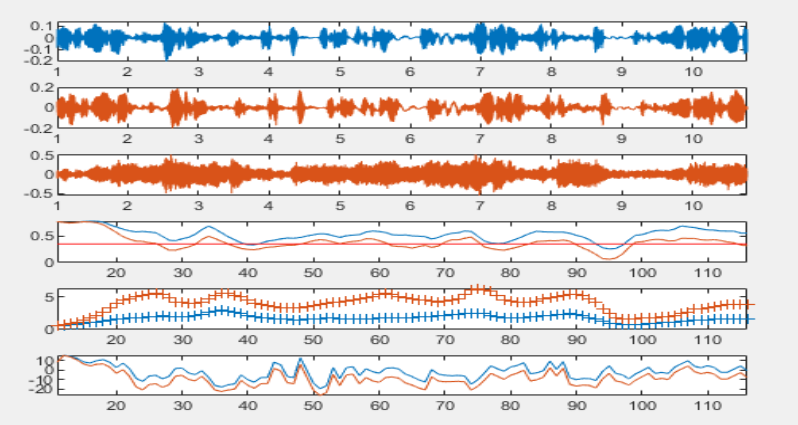

Fig 22. Representación de las señales obtenidas y ganancias en el micrófono ideal para cada ganancia aplicada

Si visualizásemos la gráfica de valores de STOI, pero esta vez para el micrófono de monitorización, observaríamos que el efecto a nivel gráfico sería similar, los valores de STOI cuando aplica ganancia ideal serían inferiores, pero la experiencia sería pésima, ya que al ser una ganancia superior y la medición realizada con un micrófono más próximo a la fuente de ruido, el ruido enmascarador es emitido con demasiado volúmen como para conciliar un agradable entorno de trabajo. De esta última idea cabe destacar una conclusión. La subjetividad también tiene un papel importante en este experimento, por lo tanto el hecho de mantener una QoE es tan importante como el poder enmascarar la voz para que sea ininteligible y mantener el STOI cercano al umbral asignado.

4.3 Paso 3: El proceso de virtualización

El último paso es el que le da sentido práctico al proyecto. Consiste en modelar el recorrido entre la fuente de ruido y la posición del oído humano debido a que en este paso el montaje y el entorno que se simula se tiene que parecer lo más posible al entorno real que se va a dar cuando el proyecto evolucione al siguiente objetivo que es la simulación en tiempo real para entornos de trabajo. En aplicaciones finales no vamos a disponer de un micrófono dentro del oído de una persona por lo que se modelará el camino hasta el propio oído haciendo uso del proceso de virtualización. Aunque en el montaje del setup sí se colocará un micrófono lo más cerca físicamente de donde se ubicará el oído del usuario ya que se debe estimar la respuesta impulsional para la posicion donde virtualizas. A continuación se presenta un esquema conceptual para clarificar mejor la idea sobre virtualizar el camino de propagación.

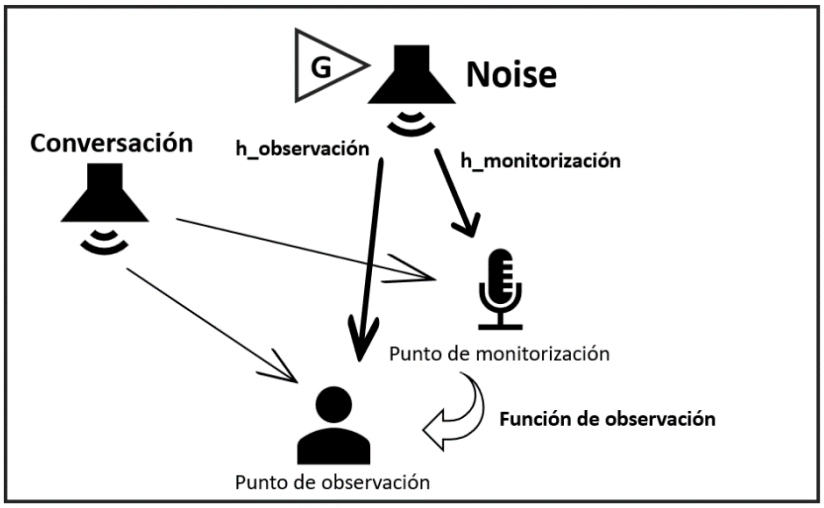

Fig 23. Esquema de los elementos que componen el tercer Set-up

Como podemos observar en el esquema, el modelado desde la fuente de ruido hasta el punto en donde se sitúa el punto ideal se va a estimar mediante la conocida como función de ideal. Con la respuesta impulsional hacia el micrófono de control y posterior filtrado con la función de observación, podemos llevarnos esta respuesta al punto que se quiere virtualizar. h_{total} es el resultado final de esta respuesta impulsional. A este punto se le conoce como de virtualización y corresponde con el efecto real que va a percibir el usuario del sistema. Por ello llamar al micrófono físico que se encuentra en el oído del *dummy*, porque es el efecto ideal que se percibiría si este micrófono ideal existiese en el sistema. En nuestro caso siempre se deja a modo de comparación ya que gran parte de este trabajo requiere de una evaluación analítica puramente teórica.

En este tercera etapa del desarrollo y simulación del entorno de trabajo teóricamente solo vamos a disponer de un solo micrófono aunque, debido a que nos encontramos en una primera etapa de estudio y simulación, vamos a mantener activo el micrófono que se encuentra en los oídos del *dummy* con el objetivo de compararlo con los parámetros y la respuesta obtenida con el proceso de virtualización. Cuanto más se parezcan, mejor se estará realizando la virtualización. Cabe destacar que cuanto más lejos se encuentre el micrófono ideal, la respuesta obtenida mediante el proceso de virtualización será menos correlada con la obtenida con el micrófono ideal.

A continuación se van a enumerar las expresiones matemáticas que se van a seguir en este tercer paso de la etapa de simulación, respetando el orden de ejecución con el que se van a llevar a cabo:

$$y_{Mo} = s(n) + [h_{Mo} * (n(n) \cdot G_{Mo})] \quad (3.1)$$

$$\hat{s} = y_{Mo} - \left[\widehat{h_{Mo}} * (n(n) \cdot G_{Mo})\right] \quad (3.2)$$

$$\widehat{y_{ideal}} = \hat{s} + \left[\widehat{h_{Mo}} * (n(n) \cdot G_{Mo})\right] * f_{obs} \quad (3.3)$$

Finalmente, la estimación del modelado de los caminos:

$$h_{total} = \widehat{h_c} * f_{obs} \approx \widehat{h_{ideal}} \quad (3.4)$$

$$y_{virt} = \hat{s} + \left[\widehat{h_{ideal}} * (n(n) \cdot G_{virt})\right] \approx y_{ideal} \quad (3.5)$$

$$stoi_{virt} = stoi(\hat{s}, y_{virt}) \approx stoi_{ideal} \quad (3.6)$$

En cuanto al código del paso 3, hay que realizar una implementación de código en MATLAB bastante importante para el cálculo del proceso de virtualización. El código implementa un ruido blanco gaussiano *AWGN* con el objetivo de que este convolucione con los caminos modelados tanto hacia el micrófono de monitorización como ideal. Después calcularemos la nueva variable función de ideal f_{obs} haciendo uso de una nueva función llamada *getObservation.m*. Finalmente se filtra mediante un filtro dicha variable junto con el $h_{monitorización}$ para obtener el h_{total}, que será muy similar al h_{ideal}.

Para terminar el estudio de este tercer paso de la etapa de simulación se va a representar tanto las respuestas obtenidas como las señales tras el filtrado así como los valores del parámetro STOI a lo largo de la simulación obtenidos en cada instante de tiempo:

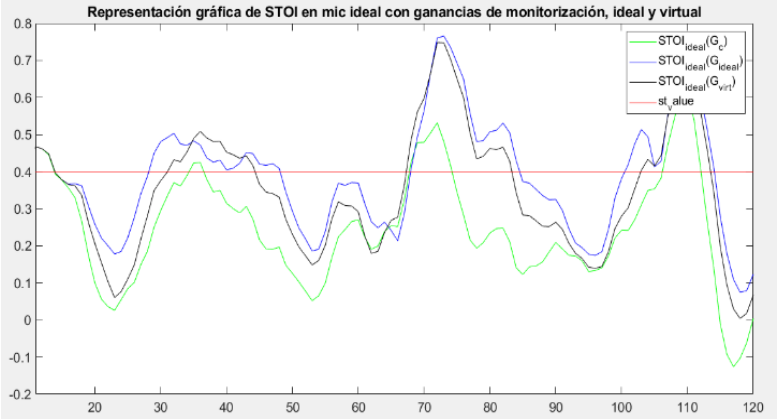

Fig 24. Valores de STOI en cada micrófono aplicando su respectiva ganancia

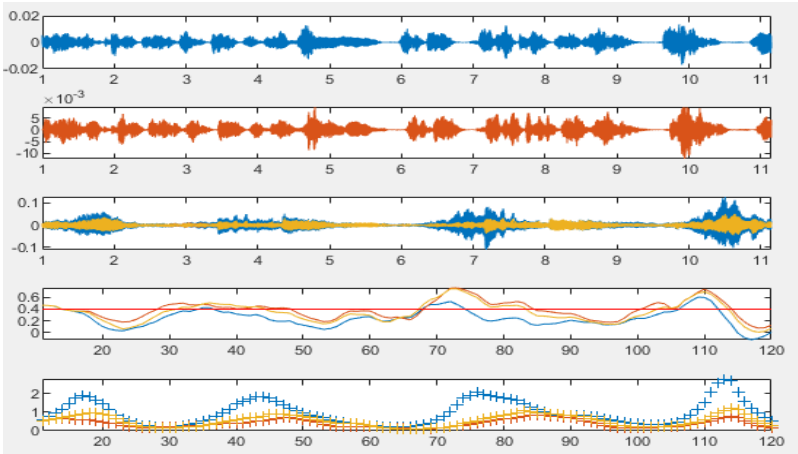

Fig 25. Señales y ganancias obtenidas en cada punto y aplicando su respectiva ganancia

Los valores de STOI son los calculados en el punto ideal pero aplicando en cada caso una de las 3 ganancias que se obtienen a partir del STOI en cada punto, es decir se calcula cada ganancia según el STOI particular en cada punto pero luego se representa el STOI que se obtiene en el punto donde se encuentra el micrófono ideal si se utilizase cada vez una de las 3 ganancias anteriormente comentadas. Como se puede observar en la gráfica de la figura 24, los parámetros de STOI tanto en el punto ideal como en virtualización son muy similares a lo largo de la simulación, por lo que la estimación realizada mediante el proceso de virtualización se acerca mucho al objetivo. Si el punto ideal estuviese más cercano al micrófono de monitorización, la estimación se asemejaría mucho más al punto ideal y por tanto, los valores de STOI tanto en ideal como en virtualización serían mucho más parecidos.

En cuanto a la gráfica de la derecha, lo más destacable es la diferencia de ganancias que se calculan en cada instante, en donde podemos observar una mayor diferencia entre los valores de ganancia que se aplican a la fuente de ruido. Los valores de ganancia calculados a partir del parámetro STOI tanto en ideal como mediante el proceso de virtualización son muy similares, una buena señal de que dicho proceso se ha realizado correctamente.

Para terminar este apartado, vamos a analizar la tercera gráfica de la figura 25, que corresponde a la representación de la señal captada en cada punto, es decir, el conjunto de las señales de voz y ruido una vez se ha aplicado la ganancia. La diferencia entre lo que ha capturado el micrófono ideal y lo generado mediante virtualización es muy pequeña, prácticamente se superponen ambas. Mientras, estas difieren de la señal capturada en monitorización, como era de esperar, pues no se encuentra en el mismo punto físico.

El siguiente paso consiste en analizar las respuestas impulsionales generadas a partir de las tomas de muestras del laboratorio. Se pretende realizar un estudio de las respuestas impulsionales tomadas de un sistema causal primero, y no causal después, analizando en cada caso la forma que se consigue de cada respuesta y compararla con la que se espera conseguir en el punto de monitorización para ver si hay un correcto proceso de virtualización.

Capítulo 5. Causalidad y No Causalidad en el sistema

5.1 Configuración causal

En cuanto a un sistema causal, es básicamente lo que se ha tratado en apartados anteriores durante el proceso de simulación. La causalidad o no causalidad la determinan la posición de los transductores en el sistema. En dicha configuración, la fuente de sonido enmascarador queda tras el micrófono ideal y el punto que se quiera virtualizar, en una posición más cercana a estos que los puntos de monitorización. La fuente de ruido se modela como un punto más alejado de todos los elementos anteriormente mencionados, pero físicamente más cercana a los puntos de monitorización que el ideal y el virtualizado. Si el sistema debe generar una respuesta o como se ha mencionado, función de observación para poder generar dicha respuesta emulando el camino de propagación hacia el punto a virtualizar, debe llegar antes la señal a los micrófonos de monitorización.

Como ya se trabajó en el último paso de simulación al modelizar la respuesta impulsional total hacia el punto ideal desde el punto de monitorización, para este apartado se ha aumentado el número de puntos de monitorización a tres y también se han desplazado dentro de lo posible siempre cumpliendo con el objetivo de causalidad del sistema. La representación gráfica de las respuestas impulsionales conseguidas se visualizan a continuación:

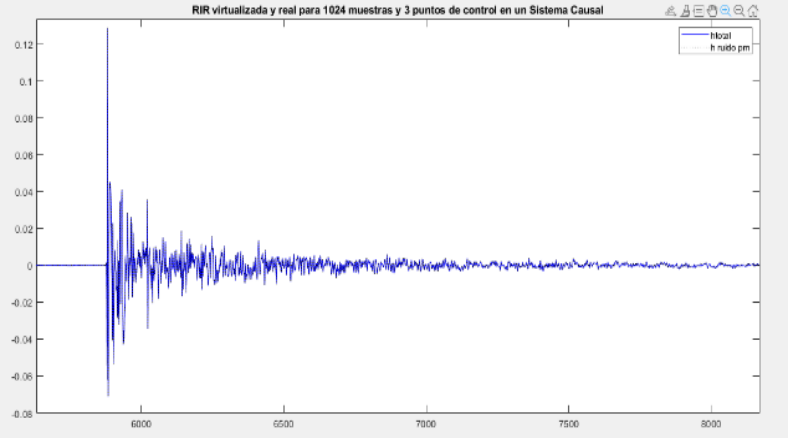

Fig 26. RIRs para 1024 muestras y 3 micrófonos de monitorización en un sistema causal

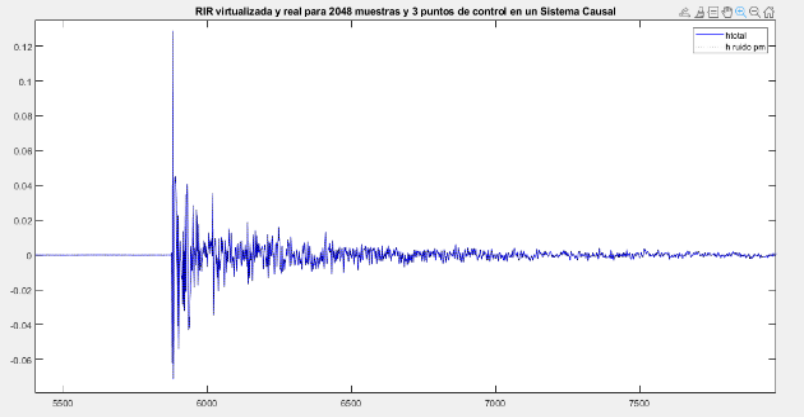

Fig 27. RIRs para 2048 muestras y 3 micrófonos de monitorización en un sistema causal

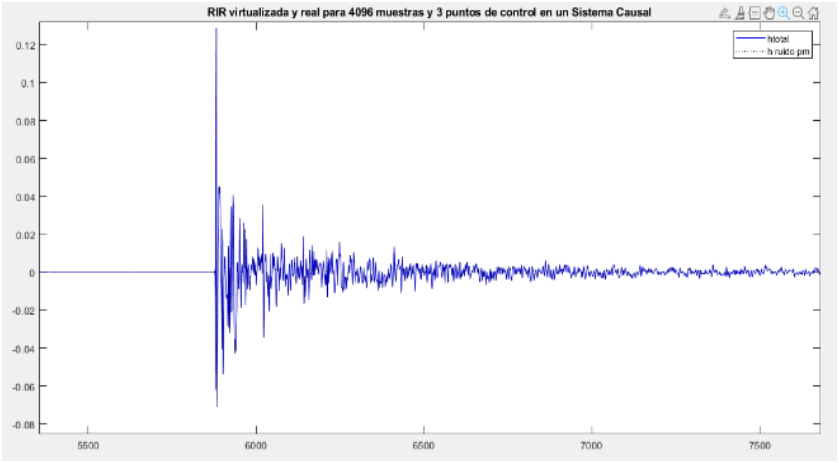

Fig 28. RIRs para 4096 muestras y 3 micrófonos de monitorización en un sistema causal

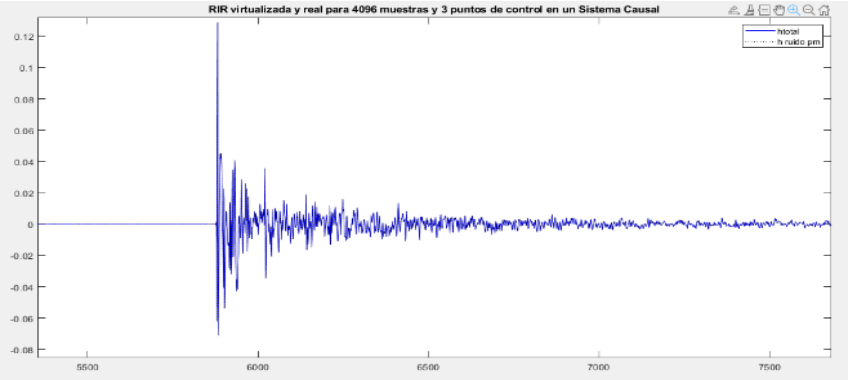

Fig 29. RIRs para 8192 muestras y 3 micrófonos de monitorización en un sistema causal

Las cuatro gráficas representan tanto la respuesta impulsional total que se consigue al virtualizar (en color azul), como la teórica que se conseguiría en el punto ideal (puntos negros), la cual es muy difícil de observar puesto que ambas se superponen .Se comprueba por tanto que, para tres puntos de monitorización y modificando la longitud del filtro, podemos conseguir un resultado satisfactorio en todos los casos, aunque nos faltaría conocer la validación subjetiva del usuario, pero no es esa una cuestión a abordar en este apartado.

En cuanto a realizar la virtualización en un solo punto, es básicamente lo que se hizo el el paso 3 como se ha explicado anteriormente, siendo indiferente el punto de monitorización que se elija, ya que el montaje en el laboratorio los puntos de monitorización eran bastante próximos entre sí. Así que se opta por realizar el mismo estudio usando solo 2 puntos de control y observando cómo evoluciona el proceso de virtualización. Para ello se ha optado por el caso más crítico que sería utilizar los dos puntos de control más distantes físicamente entre sí. El resultado gráfica de las respuestas impulsuionales conseguidas es el siguiente:

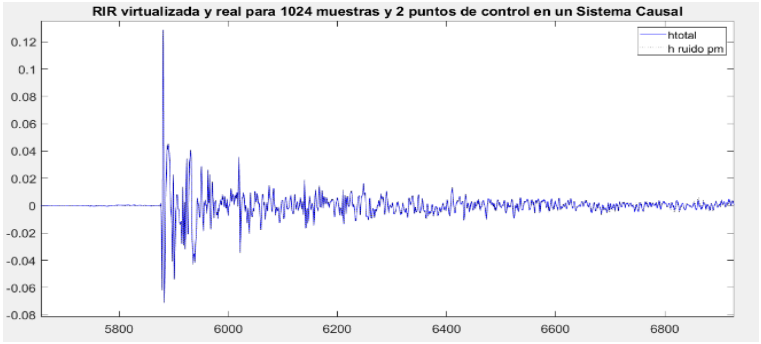

Fig 30. RIRs para 1024 muestras y 2 micrófonos de monitorización en un sistema causal

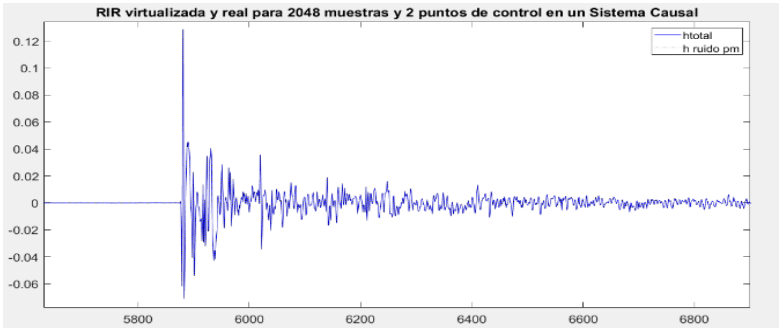

Fig 31. RIRs para 2048 muestras y 2 micrófonos de monitorización en un sistema causal

Fig 32. RIRs para 4096 muestras y 2 micrófonos de monitorización en un sistema causal

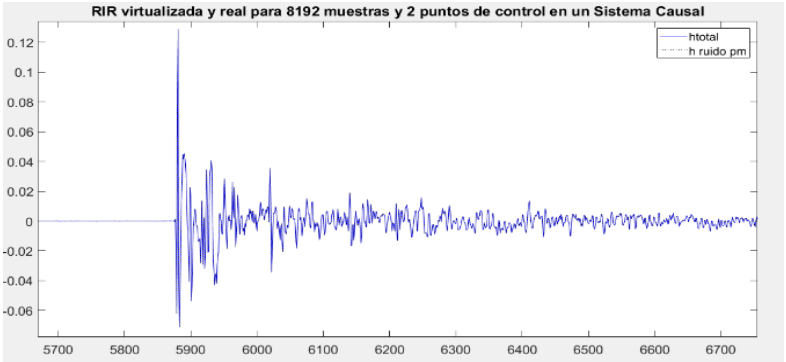

Fig 33. RIRs para 8192 muestras y 2 micrófonos de monitorización en un sistema causal

Se observa que no hay mucha diferencia en cuanto a la respuesta impulsional de 2 puntos de monitorización con las respuestas impulsionales representadas al inicio de este apartado, por lo que se decir que la estimación mediante virtualización de la respuesta impulsional hacia el punto ideal es muy buena desde cualquier punto de monitorización y haciendo uso de cualquier punto de monitorización de entre los disponibles.

Finalmente como análisis final de este sistema causal, vamos a visualizar el resultado que se obtendría al contar con solo un micrófono de monitorización en este sistema:

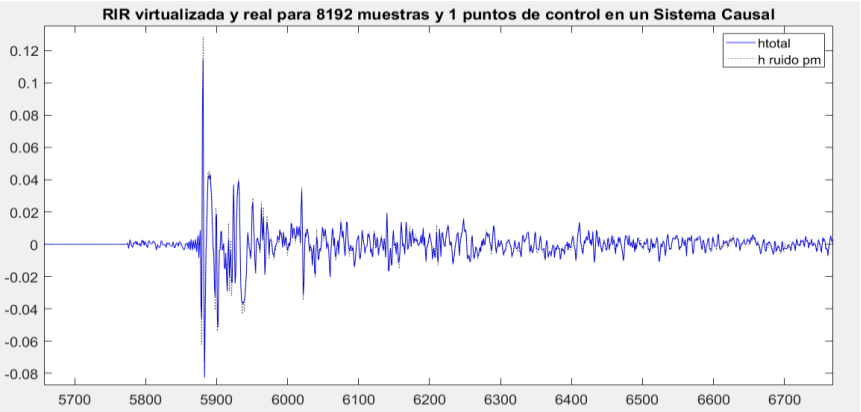

Fig 34. RIRs para 8192 muestras y 1 micrófono de monitorización en un sistema causal

Podemos observar cómo la respuesta impulsiva que podemos conseguir mediante virtualización se parece bastante a la que se obtiene de manera ideal haciendo uso de un micrófono ideal. Por tanto, en un sistema causal con tan solo un punto de monitorización el sistema es capaz de rendir adecuadamente, teniendo en cuenta la validación aplicada.

5.2 Configuración no causal

La funcionalidad de este proyecto de investigación debe ser extrapolable casi a cualquier combinación posible de posiciones en donde se pueda encontrar tanto los puntos de monitorización y ideal como la fuente de ruido. Y el resultado debe ser satisfactorio en cada uno de ellos. Así que el estudio del esquema cuando la fuente de ruido se encuentra mucho más cercana al punto ideal que a los puntos de monitorización resulta muy interesante porque hasta ahora no se ha estudiado qué pasa cuando el sistema presenta no causalidades, una situación que podría darse en entornos de trabajo normales. Esta configuración se podría dar si dos personas mantienen una conversación justo al lado del usuario y el oído de este capta la señal de voz antes que los micrófonos de monitorización del sistema.

El esquema y la distribución de los distintos elementos en este objeto de estudio se presentan a continuación:

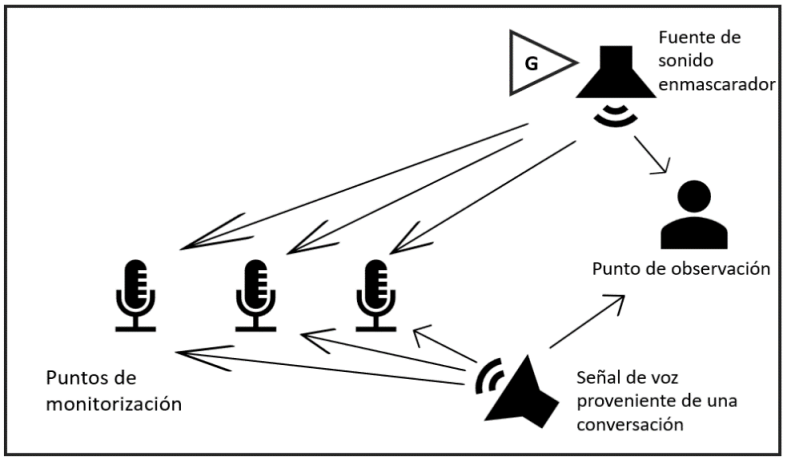

Fig 35. Configuración del sistema no causal con 3 micrófonos de monitorización

Se puede observar claramente que la fuente de ruido está mucho más cercana al punto ideal que a los de monitorización y, entre la fuente y los distintos puntos hay obstáculos que pueden interferir en la respuesta impulsional. El sistema necesita un tiempo de procesado mayor al de propagación acústica, al no ser así es esto lo genera no causalidades en el sistema, ya que la respuesta virtualizada está más adelantada en el tiempo que la respuesta hacia los micrófonos de monitorización

En este apartado vamos a representar las respuestas impulsionales totales fruto de la virtualización y teóricas que se conseguirían en el punto ideal. Se va a tener en cuenta la duración del filtro en cada caso porque puede resultar interesante estudiar el efecto de utilizar filtros más cortos o largos que otros y ver el efecto que pueden tener sobre la respuesta impulsional total. A continuación se representan las respuestas impulsionales total (en azul) y teórica (punteado en negro) para filtros de 1024, 2048, 4096 y 8192 muestras respectivamente:

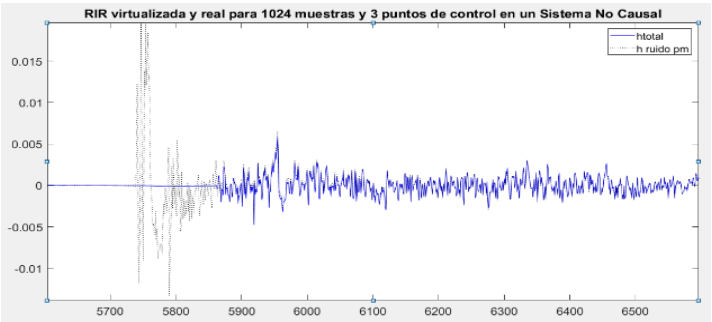

Fig 36. RIRs para 1024 muestras y 3 micrófonos de monitorización en un sistema no causal

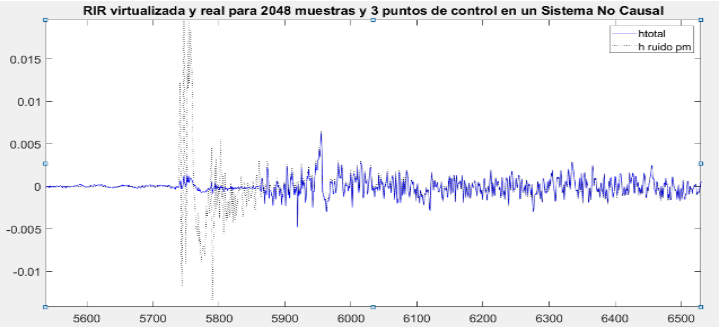

Fig 37. RIRs para 2048 muestras y 3 micrófonos de monitorización en un sistema no causal

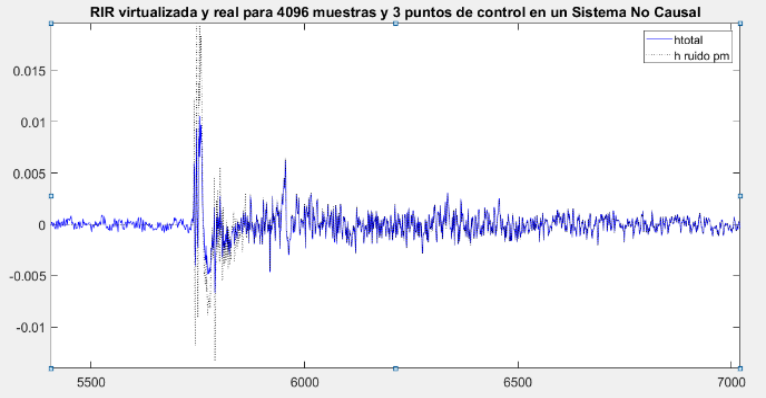

Fig 38. RIRs para 4096 muestras y 3 micrófonos de monitorización en un sistema no causal

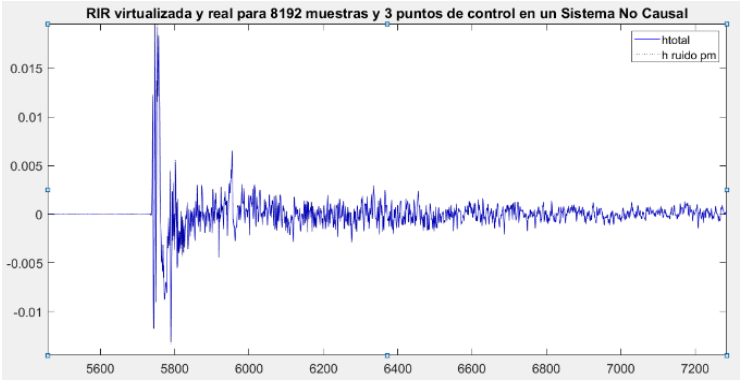

Fig 39. RIRs para 8192 muestras y 3 micrófonos de monitorización en un sistema no causal

Observando las cuatro gráficas presentes podemos concluir que la longitud del filtro sí es relevante cuando nos encontramos en una situación de no causalidad. Las gráficas de la parte superior, que corresponden a los filtros de 1024 y 2048 observamos cómo la respuesta impulsional virtualizada no presenta el primer pulso de mayor amplitud tal y como ocurre en la respuesta del punto ideal. Es solo ruido retardado un número de muestras tras el pulso inicial, el de mayor amplitud. Sin embargo, en la gráfica inferior izquierda, que corresponde al filtro de 4096 muestras se puede apreciar que la respuesta ya se asemeja bastante a la original que se obtendría en el punto ideal. Finalmente, la última gráfica que nos queda por comentar y analizar coincide con la del filtro de 8192 muestras. En ella podemos observar cómo se superponen ambas gráficas, dándonos a entender que para un filtro de esas dimensiones, el sistema ya es capaz de virtualizar correctamente con 3 puntos de monitorización la respuesta que se obtendría en el punto ideal. La hipótesis que obtenemos de este apartado es que para sistemas en los que se generen no causalidades, la mejor solución para una correcta virtualización es utilizar filtros con un gran número de muestras. En este caso, el máximo que disponemos que es de 8192.

Una buena opción para calcular el error introducido entre las respuestas impulsionales, la estimada a partir de la virtualización y la que se obtendría teóricamente en el punto ideal, es mediante figuras de mérito como el error cuadrático medio MSE como el *non-parametric partial mean* o NPM ,es decir métricas estadísticas que se emplean para medir el error o diferencia entre señales o respuestas de este estilo. En el caso del MSE, vamos a trabajar con la señal filtrada por un ruido blanco gaussiano que generamos durante el proceso para calcular la respuesta impulsional resultante de virtualizar. Esto se debe a que al calcular el MSE necesitamos calcular el error en cada frecuencia para obtener un resultado más satisfactorio en caso de que una de las componentes a cierta frecuencia no haya podido ser virtualizada correctamente. Para el caso del NPM, trabajaremos con la respuesta previa al filtrado con el ruido blanco. Como las respuestas para longitudes de muestra

de 1024 y 2048 disciernen bastante de la que se obtiene en el punto ideal, vamos a comprobar tan solo los que más se le parezcan, como son las respuestas obtenidas para 4096 y 8192 muestras. Para los siguientes ejemplos también se va a proceder de la misma manera, si es que hay respuestas que se perciban similares a las teóricas. Los resultados obtenidos en este experimento son los siguientes:

Número de muestras	MSE (dB)	NPM (dB)
4096	-3.10	-3.59
8192	-35.62	-36.52

Tabla 3. Valores de MSE y NPM para 4096 y 8192 muestras en 3 puntos de monitorización para la configuración no causal

Como podemos observar, la similitud entre las respuestas impulsionales para 8192 muestras es abrumadora. No tanto para las respuestas de 4096 muestras, cuyos valores son cercanos a 0 dB. En parte se debe que a que la principal componente de la respuesta virtualizada es de bastante menor amplitud, como podemos observar en la gráfica mostrada anteriormente.

Los valores de MSE que hemos obtenido anteriormente resultan de filtrar las respuestas impulsionales por un ruido blanco gaussiano. Una buena oportunidad de estudio sería estudiar el valor de MSE de las respuestas impulsionales filtradas con distintos sonidos enmascaradores. Para ello, se han obtenido los ruidos enmascaradores de *MusicNoise* filtrado a 5kHz y, *PinkNoise*, *WindBirdNoise* y *WaterNoise* filtrados a 8kHz. Además se ha añadido un ruido de pendiente a -5dB también filtrado a 8kHz. La razón de haber filtrado todos a 8kHz excepto el *MusicNoise* a 5kHz se debe a que la naturaleza del sonido de un violín presenta componentes a altas frecuencias con mucho más baja energía que el resto. Finalmente, se presentan los resultados del MSE para cada sonido enmascarado filtrado para 4096 y 8192 muestras:

Ruido Enmascarador	MSE (dB) para 8192 muestras	MSE (dB) para 4096 muestras
Ruido de pendiente	-47.87	-4.14
PinkNoise	-44.83	-4.62
MusicNoise	-44.66	-2.49
WindBirdNoise	-44.96	-4.11
WaterNoise	-43.93	-5.44

Tabla 4. Comparativa entre valores de MSE para los nuevos sonidos de enmascaramiento en 3 micros de monitorización

Otro objeto de estudio de gran interés consiste en testear cómo va a rendir un sistema que presente no causalidades con un número menor de puntos de monitorización, es decir, con tan solo 1 o 2 puntos de monitorización. En el paso 3 del apartado que llamamos de simulación testeamos la virtualización con solo 1 punto de monitorización como ya se ha venido mencionado a lo largo de este documento. La cuestión viene que los sistemas en los que se probó eran todos causales, así que ahora es de especial interés testearlo en este sistema para valorar las prestaciones que pueda tener este sistema. Se ha testeado con el punto de control más a la derecha del puesto de trabajo, la misma posición que ocupaba el punto de control cuando se realizó el estudio del paso 3. A continuación se presenta la gráfica para 8192 muestras en el filtro, el óptimo:

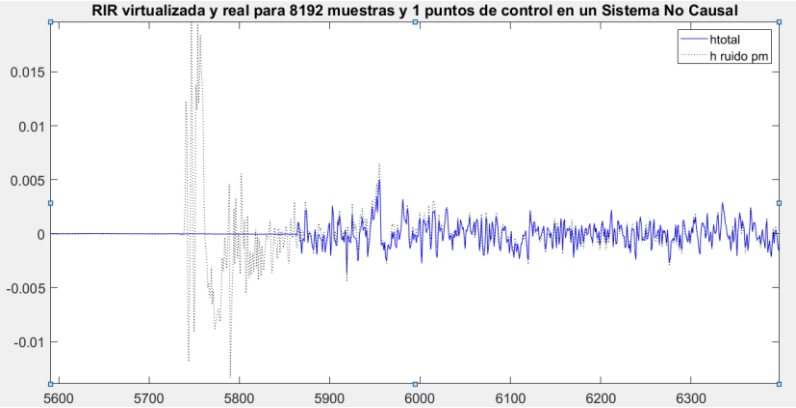

Fig 40. RIRs para 8192 muestras y 1 micrófono de monitorización en un sistema no causal

En este caso podemos observar cómo aumentando el número de muestras del filtro, no podemos obtener una respuesta virtualizada similar a la obtenida en el punto ideal. Se debe a que cuando se presentan no causalidades, el sistema no es capaz de virtualizar correctamente. Tampoco es viable hacer uso de filtros más grandes que de 8192 muestras, ya que el tiempo de procesado se eleva considerablemente, aunque en términos de simulación no resulte ser un problema crítico. En conclusión, cuando aparecen no causalidades en el sistema, obtener el punto ideal mediante virtualización resulta imposible. La solución pasa por estimar dicha respuesta con varios puntos de monitorización.

Para zanjar finalmente este apartado, vamos a obtener las respuestas impulsionales cuando se hace uso de al menos 2 puntos de monitorización, los que se encuentran a los extremos del puesto de trabajo. Las respuestas impulsionales son las siguientes:

Fig 41. RIRs para 1024 muestras y 2 micrófonos de monitorización en un sistema no causal

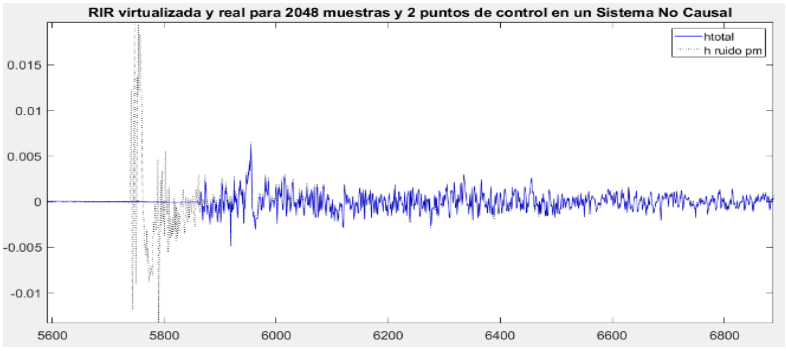

Fig 42. RIRs para 2048 muestras y 2 micrófonos de monitorización en un sistema no causal

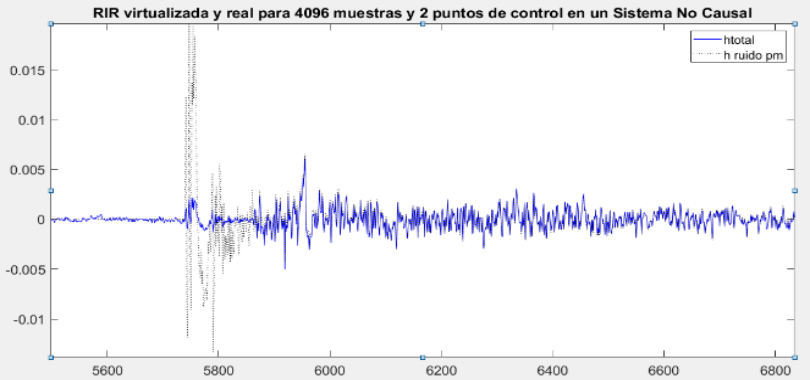

Fig 43. RIRs para 4096 muestras y 2 micrófonos de monitorización en un sistema no causal

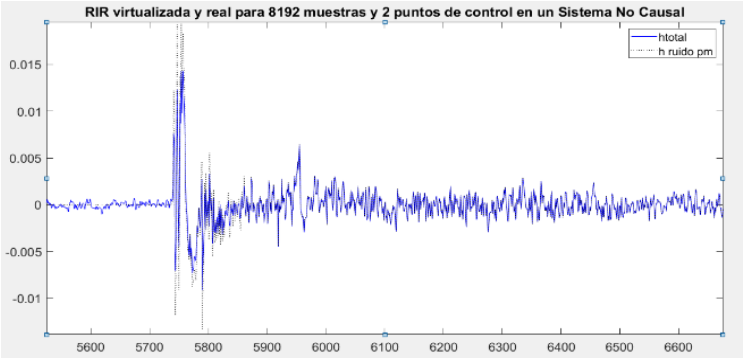

Fig 44. RIRs para 8192 muestras y 2 micrófonos de monitorización en un sistema no causal

Se puede observar en las gráficas, que para solo 2 puntos de monitorización (los más distantes entre sí) la única respuesta que se asemeja es la obtenida en el filtro de mayor longitud. El problema que ocurre es que como se puede ver la respuesta impulsional virtualizada del filtro de 8192 muestras no llega a ser exactamente como la teórica. Sucede algo muy similar con la respuesta obtenida con el filtro de 4096 muestras para 3 puntos de monitorización distintos. Las prestaciones que ofrece no son del todo buenas y utilizar filtros más grandes tampoco es una buena opción. Así que, dada la naturaleza del sistema, hacer uso de tan solo 2 puntos de monitorización en sistemas no causales no resulta ser muy buena opción si se dispone de filtros de menor número de muestras.

En cuanto al valor del error entre las respuestas impulsionales, analizaremos por tanto solo la que corresponde a la de 8192 muestras:

Número de Muestras	MSE (dB)	NPM (dB)
8192	-5.71	-6.41

Tabla 5. Valores de MSE y NPM para 8192 muestras en 2 puntos de monitorización para la configuración no causal

La conclusión final a la que se llega en este apartado al haber visto y comentado los anteriores casos es que para este sistema no causal, como mínimo se debe hacer uso de al menos 3 puntos de monitorización para obtener resultados satisfactorios en cuanto a la virtualización de la respuesta impulsional del punto ideal.

Finalmente, para el cálculo del MSE cuando tenemos 2 puntos de monitorización, tan solo tenemos en cuenta como hemos estudiado anteriormente el caso que tenemos 8192 muestras:

Ruido Enmascarador	**MSE (dB) para 8192 muestras**
Ruido de pendiente	-8.30
PinkNoise	-8.06
MusicNoise	-8.46
WindBirdNoise	-6.17
WaterNoise	-8.45

Tabla 6. Comparativa entre valores de MSE para los nuevos sonidos de enmascaramiento en 3 micros de monitorización

Vemos que los valores de MSE tampoco son muy buenos, demostrando que, aunque sea la única opción en que se asemejan, la estimación se aleja bastante para considerarse buena.

Capítulo 6. Desarrollo del prototipo en tiempo real

Hasta este punto se había trabajado con un sistema formado por 2 altavoces, en donde se estimaban las respuestas impulsionales para estudiar el comportamiento del sistema en un entorno puramente simulado a través de MATLAB. En dicho entorno, se trabajó en todo momento en el dominio temporal. Realizarlo de este modo ha servido para obtener la idea de cómo el sistema captura y procesa la información y es capaz de dar como resultado una respuesta que cumple con los objetivos de inteligibilidad establecidos. También se ha conseguido una idea clara de qué características y distribuciones de los componentes que forman el sistema son mejores o peores en cuanto a rendimiento y resultado final. Pero la forma con la que se ha trabajado dista en gran medida a cómo se va a trabajar cuando el sistema funcione en tiempo real, viene condicionado con los dispositivos con los que se deben emplear. A continuación se va a explicar en detalle en que se ve afectado este cambio de paradigma hasta ahora estudiado.

6.1 Uso de las funciones de observación

El correcto uso de las llamadas funciones de observación, unas respuestas que al convolucionar con las respuestas impulsionales filtradas se obtiene la respuesta de virtualización, se tiene que usar la señal capturada por el micrófono de monitorización. También existe la posibilidad de utilizar las señales grabadas en los micros. Para conseguir dichas funciones de observación en el dominio frecuencial no necesitamos específicamente las respuestas al impulso, sino lo que graban los distintos micrófonos. De la fuente de ruido a los micrófonos sí podemos obtener las RIR porque se conoce la señal de ruido de enmascaramiento que va a emitir el altavoz y también se pueden obtener la señal de ruido en cada micrófono. Por otra parte, no conocemos las respuestas al impulso de la fuente de voz, así que usaremos a partir de ahora la señal grabada por los micros para obtener la función ideal para dicha voz y poder usarla con el sistema de enmascaramiento.

6.2 Análisis de las señales en el dominio frecuencial

El objetivo a partir de ahora trabajar en el dominio frecuencial reside es que se puede analizar cada componente frecuencial por separado para obtener una estimación y, finalmente, una respuesta mucho más precisa. También porque es mucho más eficiente para el hardware del sistema trabajar con muestras separadas en frecuencia. Otro punto a favor de este nuevo paradigma reside en que, como se ha explicado en el párrafo anterior, la forma de obtener las respuestas impulsivas ha cambiado, así que trabajar en frecuencia facilita bastante en cuanto a eficiencia del código y del número de operaciones. La convolución y el filtrado de las señal en tiempo se convierten en simples operaciones aritméticas como sumar y multiplicar vectores. Esta nueva forma de operar nos conduce también a realizar ciertos cambios en las ecuaciones y fórmulas empleadas en el proceso de simulación en dominio tiempo, es decir, de la (1.3.1) a la (1.3.6).

6.3 Virtualización de la voz

Otro cambio de paradigma importante en esta nueva forma de operar con el sistema reside en el hecho de que, a diferencia de antes que con la función ideal filtrábamos con las respuestas impulsionales de los micrófonos de monitorización para conseguir la respuesta virtualizada, ahora lo que vamos a emplear para virtualizar junto con la función ideal es la propia señal de voz. El hecho de que el sistema debe emplearse en entornos abiertos y dinámicos como unas oficinas, en donde las señales de voz que queremos enmascarar pueden venir de cualquier dirección, hace pensar que virtualizar la señal de voz sea la opción más lógica si se desea que el sistema sea capaz de enmascarar la voz si esta proviene desde cualquier punto y, también si se diese el caso de que el sistema cambie en causalidad. A continuación se representa un esquema gráfico con todas las

respuestas impulsionales que se encuentran en el sistema y una nomenclatura adecuada para una posterior explicación detallada de las nuevas fórmulas del nuevo paradigma del sistema:

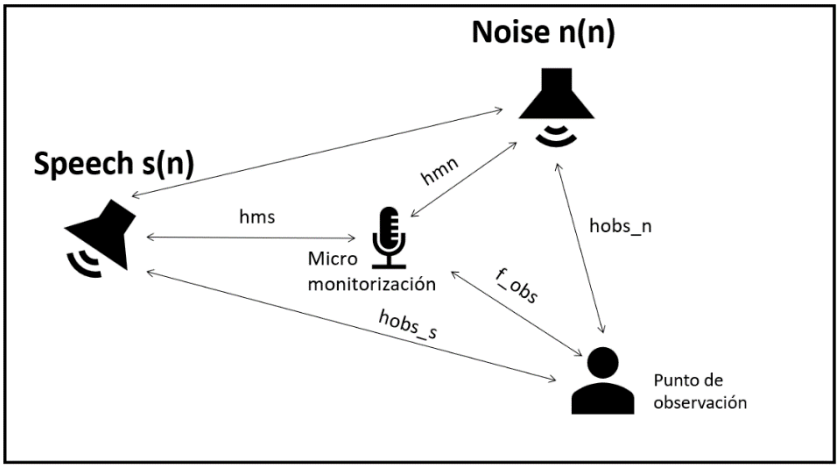

Fig 45. Esquema de los elementos que intervienen en este nuevo paradigma junto con las RIRs existentes

Para cuando el sistema presente más de un punto de monitorización, el esquema será igual, pero replicado tantas veces como puntos de monitorización adicionales presente. Una vez representado el esquema del sistema con esta nomenclatura, pasamos a la explicación teórica y representación de las ecuaciones que intervienen en cada punto.

La primera etapa que interviene en la simulación del sistema en tiempo real es el set-up, en donde se realiza el montaje del propio sistema, colocando en su posición tanto los micrófonos como los altavoces que generarán las señales primarias. En las primeras pruebas del nuevo sistema en dominio frecuencial, primero se trabaja en un entorno puramente simulado como se hacía anteriormente en los distintos estudios realizados cuando se trabajaba en el dominio temporal, pero con la particularidad del cambio de dominio. Se procede así para una mayor comodidad a la hora de realizar el estudio del sistema con el nuevo paradigma y obtener finalmente el algoritmo con las operaciones pertinentes que de como resultado una respuesta satistfactoria. También para una mejor comparativa de las respuestas, se sigue manteniendo el micrófono ideal, para comparar los resultados obtenidos con el proceso de virtualización (real), con los que capturaría el oído humano en dicha posición ideal.

En esta etapa previa, calcularemos las respuestas impulsionales desde el altavoz de ruido, el único que podemos caracterizar acústicamente. Es decir, estimaremos las respuestas $\widehat{h_{obs,n}}$ y $\widehat{h_{m,n}}$

El siguiente proceso consiste en grabar la voz con los micrófonos para calcular la función ideal para la fuente de voz. Dicha función ideal se va a obtener mediante las siguientes expresiones:

$$y_m(n) = s(n) * h_{ms} \qquad (2.1.1)$$

$$y_{ideal}(n) = s(n) * h_{ideal} \qquad (2.1.2)$$

Si nos fijamos , las anteriores expresiones hacen referencia a las señales que seran capturadas en cada uno de los micrófonos antes de enmascarar la señal. La (2.1.2) se puede capturar porque mantenemos el micrófono en dicha posición en la fase de desarrollo del prototipo para comparar con los resultados del punto virtualizado.

Una vez obtenemos la función ideal, el algoritmo del sistema empieza a funcionar y calcular las respuestas. A partir de ahora la expresión (2.1.1) pasa a ser la siguiente:

$$y_m(n) = s(n) * h_{ms} + \left[h_{mn} * \left(n(n) \cdot G_{ideal}(n)\right)\right] \quad (2.1.3)$$

Para conseguir la señal total que se capturaría en la posición del punto ideal, vamos a ultilizar la anterior expresión (2.1.3), donde eliminaremos el ruido estimado en el micrófono de monitorización, ya que de esta sí conocemos la señal:

$$\widehat{n_m}(n) = \widehat{h_{mn}} * \left(n(n) \cdot G_{ideal}(n)\right) \quad (2.1.4)$$

$$\widehat{s_m}(n) = y_m(n) - \widehat{n_m}(n) \quad (2.1.5)$$

Con la expresión (2.1.5) obtenemos la señal de voz estimada. Con dicha señal y la función ideal, nos llevaremos la señal de voz estimada a la posición del micrófono ideal:

$$\widehat{s_{ideal}}(n) = \widehat{s_m}(n) * h_{virt} \quad (2.1.6)$$

Por otra parte, el ruido de enmascaramiento en la posición del oyente se puede conseguir como:

$$\widehat{n_{ideal}}(n) = \widehat{h_{obs\_n}} * \left(n(n) \cdot G_{ideal}(n)\right) \quad (2.1.7)$$

La señal que se captura en el punto ideal se puede estimar sumando las señales obtenidas en (2.1.6) y (2.1.7):

$$\widehat{y_{ideal}}(n) = \widehat{n_{obs}}(n) + \widehat{s_{ideal}}(n) \quad (2.1.8)$$

La inteligibilidad en el punto ideal y el valor de la ganancia $G_{obs}(n)$ que se aplicará en la fuente de ruido se obtiene de con la función de STOI:

$$stoi_{ideal}(n) = stoi\left(\widehat{s_{ideal}}(n), \widehat{y_{ideal}}(n)\right) \quad (2.1.9)$$

6.4 Primera etapa del sistema en tiempo real

Una vez explicada la parte teórica del sistema, pasamos a la parte más práctica. En esta parte del desarrollo en tiempo real, se va a seguir trabajando con las respuestas impulsionales tal y como se realizaba anteriormente, tanto en la parte en donde se calcula la propia función ideal como en el código del algoritmo del sistema. La diferencia que se va a tener en el algoritmo del sistema cuando se trabaje en el tiempo real a cuando se trabaje en dominio frecuencial pero en entorno simulado, es decir simulando en MATLAB, es cómo se van generando las señales capturadas por los distintos puntos del sistema, ya que en tiempo real, la señal que aparece en la fórmula (2.1.3) se captura directamente desde uno de los puntos de monitorización.

Una de las grandes diferencias en cuanto al algoritmo y sus pertinentes operaciones de dentro del sistema entre el dominio temporal y frecuencial es cómo se generan las distintas señales a raíz de las señales que se conocen, es decir, las grabadas por los micrófonos y las que podemos convolucionar con las respuestas impulsionales. En el dominio temporal, se operaba tal y como se va demostrando en las expresiones de la (1.3.1) a la (1.3.6), exactamente las mismas operaciones. Pero en el dominio frecuencial las convoluciones temporales pasan a ser multiplicaciones entre dos señales, si a ello le sumamos el hecho de que en el nuevo sistema en tiempo real no disponemos de

la totalidad de la señal desde el inicio, sino que solo disponemos de tramas que dependen de la frecuencia de muestreo y de la longitud de dichas muestras, nos damos cuenta que debemos realizar dichos cálculos de otra manera. Es por ello que hacemos uso del algoritmo *OverlappAdd*, o solape suma. Este algoritmo recibe tramas de la señal de voz que se ha estimado como se explica en (2.1.4) y (2.1.5) y, la multiplica por el impulso que se requiera en cada caso. Tras esta operación, parte del resultado lo guarda en la señal que devuelve y otra pequeña parte la guarda en una variable que añadirá a la siguiente iteración cuando reciba la siguiente trama, es decir, la solapará.

6.4.1 Análisis de los resultados

Como se ha mencionado anteriormente, previo al testeo del sistema en tiempo real dentro del laboratorio, se ha realizado un estudio simulado en el entorno de MATLAB con el objetivo de comprobar si los estudios de las respuestas y causalidad realizados en apartados anteriores siguen cumpliéndose en el nuevo paradigma y qué otras nuevas características para estudiar nos podemos encontrar. Aunque se trata de un entorno simulado, en este apartado se realizan teniendo en cuenta el sistema en el nuevo paradigma. Es la antesala al prototipo real.

Lo primero, vamos a realizar el estudio cuando el sistema es causal, ya que es siempre más fiable. En este nuevo estudio, al igual que el anterior, también se llega a la firme conclusión de que el sistema es capaz de virtualizar el micrófono y, además, generar una respuesta que enmascara completamente la señal de voz, consiguiendo unos valores de STOI sobre el umbral de inteligibilidad.

A continuación se presenta una captura con las gráficas de los valores de STOI junto con las gráficas de las señales que se obtienen en los puntos más relevantes del sistema:

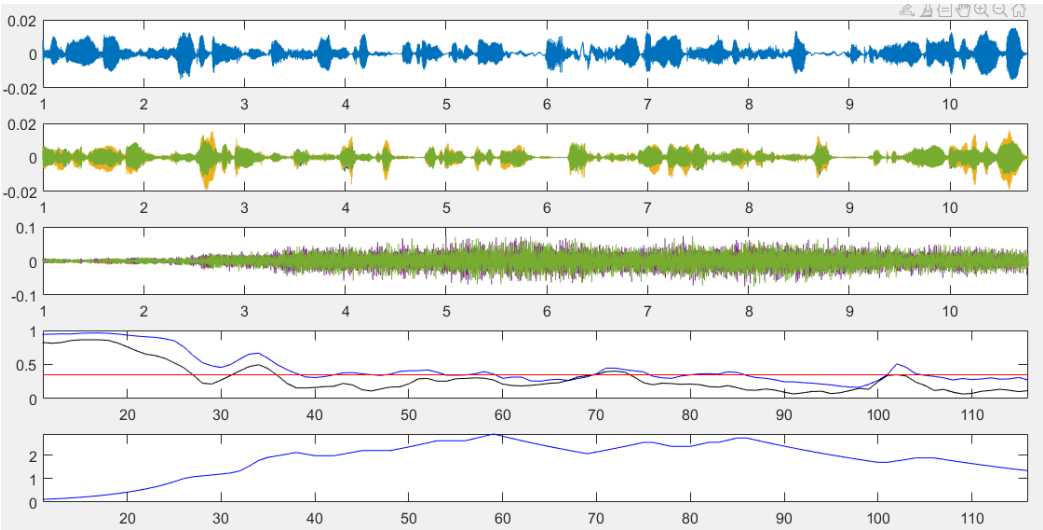

Fig 46. Gráficas dominio frecuencial para sistema causal y 8192 muestras

Como estamos en el estudio del sistema causal, los micrófonos de monitorización quedan físicamente más cercanos a la fuente de voz que el punto ideal y su correspondiente punto a virtualizar, por tanto es lógico pensar que los niveles de STOI para estos 3 micros de monitorización será mayor que en el resto de puntos. Efectivamente podemos comprobarlo en la cuarta gráfica contando desde arriba de la imagen. Esta gráfica corresponde con los valores de STOI. La línea azul hace referencia a los valores captados por el micrófono central de los tres que hay de monitorización, mientras que la línea negra hace referencia a los valores de STOI obtenidos en el proceso de virtualización. Los valores de STOI del punto ideal no aparecen en esta gráfica porque la línea es coincidente con la de los valores en virtualización, lo que significa que la virtualización es excelente. Cabe resaltar que se ha utilizado un filtrado con 8192 puntos. También se presenta

una buena oportunidad para estudiar el efecto que tiene el uso de filtros de distinta longitud y comprobar si en este nuevo paradigma, el sistema es capaz de obtener respuestas impulsivas virtualizadas lo suficientemente precisas para realizar una buena virtualización. En la siguiente tabla se muestran los distintos valores de MSE y NPM para distintos valores de muestras del filtro:

Número de muestras	MSE (en dB)	NPM (en dB)
8192	-69.02	-68.90
4096	-49.49	-49.32
2048	-38.80	-38.54
1024	-28.03	-28.16

Tabla 7. Valores de MSE y NPM para distinto número de muestras en configuración causal

Obtener unos valores tanto de MSE como de NPM inferiores a unos -25 dB significa que el sistema virtualiza bien. Por tanto, para un sistema causal podemos considerar la longitud del filtro que deseemos.

Otro objeto interesante para ser estudiado es cuántos micrófonos de monitorización hacen falta para realizar una buena virtualización del punto ideal. Para realizar el siguiente estudio se va a utilizar 8192 muestras. A continuación se presenta una tabla con los valores de MSE y NPM obtenidos cuando se hace uso de uno, dos y tres micrófonos de monitorización:

Número de micrófonos de monitorización	MSE (en dB)	NPM (en dB)
1	-19.32	-19.29
2	-47.33	-47.23
3	-69.02	-68.90

Tabla 8. Valores de MSE y NPM para 1, 2 y 3 micrófonos de monitorización con 8192 muestras en configuración causal

Lo que los valores de la tabla anterior nos muestran es que a partir de 2 micrófonos de monitorización podemos obtener una virtualización del micrófono ideal muy buena. Con un solo micrófono de monitorización, la virtualización que se realizaría no sería del todo buena. Unos valores de -19 dB no distan mucho de una correcta virtualización, pero tampoco son buenos resultados como plantear un sistema con un solo micrófono de monitorización.

Pasamos a estudiar el efecto que tendría si el sistema presentase una no causalidad. Cuando se realizó el anterior estudio en el sistema simulado en dominio temporal, se observó que el correcto funcionamiento del sistema se vuelve más crítico cuando el sistema se vuelve no causal. La configuración en el laboratorio ahora es distinta, el micrófono ideal junto con el punto que se desea virtualizar ahora están mucho más cercanas a la fuente de voz que los micrófonos de monitorización, lo que como resultado en la gráfica del STOI obtendremos que los valores del micrófono de monitorización central quedarán por debajo que los que se van a obtener en el punto ideal y virtualización, que en el mejor de los casos estarán muy solapadas. La siguiente imagen muestra la gráfica de las señales capturadas en los distintos puntos junto con los valores de STOI y ganancia cuando se hace uso de 8192 muestras en el filtro:

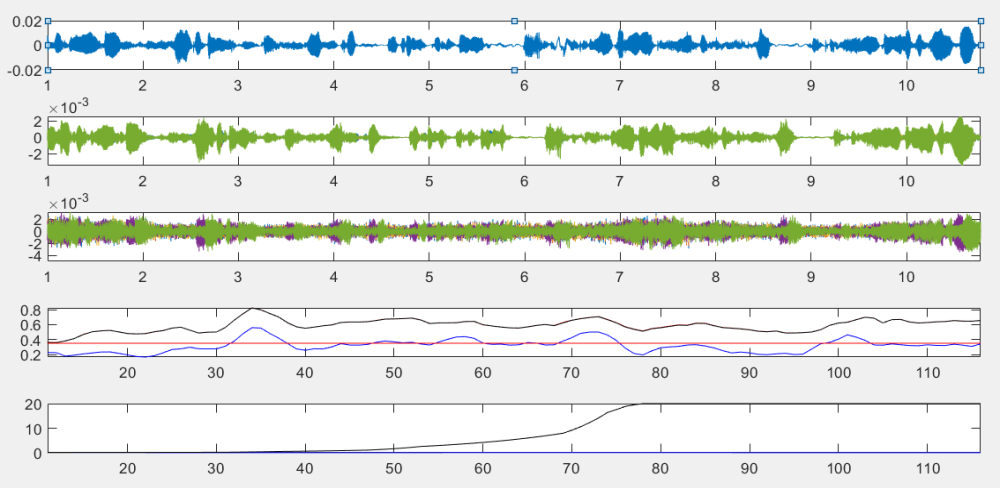

Fig 47. Gráficas dominio frecuencial para sistema no causal y 8192 muestras

Una cosa que llama la atención en la gráfica de las ganancias es que el sistema se ha saturado, entregando un máximo de ganancia a la fuente de ruido enmascarador, esto se debe a la propia configuración del sistema, ya que ahora el micrófono virtualizado se encuentra más cerca de la fuente de voz que de la de ruido, por lo tanto, debe generar mayores ganancias para intentar reducir el STOI. También se puede observar en la gráfica del STOI que los valores en el punto ideal quedan lejos del umbral. La conclusión que se obtiene es que necesitamos aumentar el máximo de la ganancia de saturación del sistema. Como 5 no es un valor muy alto podemos probar en aumentarlo unas unidades más. A partir de 10, podemos obtener valores razonablemente buenos, aunque el volúmen del ruido es bastante alto ya que empieza a rozar lo molesto y, si consideramos que el sistema está pensado para entornos de trabajo, no parece que tampoco sea una buena opción aumentar el valor de la ganancia de saturación mucho más de 5 o 7.

Siguendo con la línea de estudio que se ha seguido en el sistema causal, vamos a representar los valores de MSE y NPM que se obtienen en este sistema no causal, también para distintos número de muestras:

Número de muestras	MSE (en dB)	NPM (en dB)
8192	-36.14	-35.97
4096	-6.01	-5.82
2048	-1.39	-1.30
1024	-1.04	-0.98

Tabla 9. Valores de MSE y NPM para distinto número de muestras en configuración no causal

Podemos observar algo muy particular, en el sistema no causal ,como ya se ha visto en el estudio realizado en el anterior dominio, el número de muestras de la función ideal es mucho más crítico para este sistema no causal. Lo que lo hace particular es la diferencia entre tener 8192 muestras y 4096. La diferencia con el paradigma anterior es que para tres puntos de monitorización, tan solo nos vale la solución con 8192 muestras en la función ideal.

Veamos a ver qué ocurre cuando no se tienen tres sino dos micrófonos de monitorización. Como vimos anteriormente, para el sistema no causal el uso de reducir el número de puntos de monitorización también resultaba un sistema muy crítico, dejando de funcionar correctamente si no se usaba un número grande de muestras. A continuación se presentan los valores de MSE y NPM

para tres, dos y un micrófono de monitorización. Para este caso también se hace uso solo de 8192 muestras en la función ideal.

Número de micrófonos de monitorización	MSE (en dB)	NPM (en dB)
1	-0.91	-0.78
2	-7.96	-7.79
3	-36.14	-35.97

Tabla 8. Valores de MSE y NPM para 1, 2 y 3 micrófonos de monitorización con 8192 muestras en configuración no causal

Podemos observar claramente que en el sistema no causal, tan solo nos es válida la opción de tener tres micrófonos de control.

Como conclusión final de este apartado, para el sistema ya en dominio frecuencial, es mucho más crítico el hecho de no tener tres puntos de monitorización. La diferencia en dB entre la solución con tres puntos de monitorización y la que tiene dos es demasiado grande y, para el sistema no causal, inviable su uso. Por tanto, el hecho de implementar el sistema con tres micrófonos en monitorización es un hecho imperativo si se desea que el sistema rinda adecuadamente. También es evidente la diferencia en dB entre hacer uso de 8192 muestras o 4096. Para el sistema causal se podría utilizar perfectamente una función ideal con una longitud de 4096 muestras, pero dado que el set up de este sistema está pensado para entornos abiertos de oficina, nunca vamos a tener la certeza de que nuestro sistema presente una causalidad o no causalidad, así que para evitar problemas vamos a utilizar la solución con 8192 muestras.

6.5 Prototipo en tiempo real. Montaje y testeo del prototipo en el laboratorio

Una vez se ha comprobado que la simulación del sistema en dominio frecuencial funciona perfectamente, es hora de pasar a la parte final de este proyecto, la implementación del sistema en tiempo real, dentro del laboratorio emulando un espacio abierto con los distintos equipos presentes. El montaje del sistema es también muy importante, teniendo en cuenta donde colocar los distintos micrófonos que van a intervenir como también los altavoces que van a servir como fuente de ruido de enmascaramiento y voz.

Para intentar emular al máximo el entorno que se encontraría una persona en una oficina abierta, pero teniendo en cuenta que las pruebas van a ser realizadas dentro del laboratorio, se ha optado por utilizar una mesa de trabajo, colocando los tres micrófonos de monitorización justo enfrente de la mesa, y un poco más altos que la propia mesa, emulando como si en esa posición quedaran los micrófonos del ordenador que se tendría en frente al sentarte en el pupitre. Se ha decido utilizar una cabeza de maniquí, para que el montaje sea lo más parecido posible a la realidad. Otro detalle que se ha tenido es que, solo por motivo de comparación final para observar si el sistema funciona correctamente, se ha optado por conservar el punto de observacion aunque en la realidad el sistema si se implementase en un entorno, este no estaría presente. Este micrófono ideal se ha colocado lo más cerca de la cabeza del dummy para simular que el punto a virtualizar se va a situar lo más cerca posible a la oreja de la persona que está sentada. Finalmente, los altavoces se han colocado uno detrás del maniquí y otro a un lado de los mircófonos de monitorización para poder simular en cada momento que el sistema sea causal o no causal dependiendo de cuál de los altavoces sea la fuente de ruido o la de voz.

Una vez montado y colocado todo en su sitio, la configuración es la siguiente:

Fig 48. Montaje definitivo del sistema en el laboratorio del GTAC (iTEAM)

Hay un cambio de paradigma como es evidente en esta parte respecto a la anterior. Se trata de en la parte del código, cómo se consiguen las diferentes señales. En los apartados previos a este, se obtenían las señales que se capturan en los distintos micrófonos filtrando con la respuesta impulsional en frecuencia y realizando el solape suma. Pero en el sistema en tiempo real, tenemos disponibles las señales capturadas por los micrófonos en cada momento. Todas las señales de voz se obtienen realizando la resta acústica tal y como se menciona en (2.1.5).

Otro factor a tener en cuenta es el propio hecho a llevar a cabo el sistema de enmascaramiento y el proceso de virtualización hacia el punto ideal es el hecho de que se cuenta con elementos reales como micrófonos, altavoces y tarjetas de audio, con las imperfecciones en mayor y menor medida que estos equipos reales puedan acarrear. Es por ello que no siempre se van a conseguir resultados iguales, puede que unos más satisfactores que otros. Pero el propósito general de este proyecto a sido que el sistema planteado cumpla con sus objetivos una gran mayoría de las veces. Uno de los efectos adversos a los que nos podemos enfrentar es que las tarjetas de audio no siempre sean capaces de realizar las operaciones acústicas correctamente. En parte debido a que al trabajar en tiempo real y mediante tramas y solapados, puede que en una de esas tramas y solapes una o dos muestras se adelanten o atrasen respecto a la orginal causando resultados extraños. Este tipo de acontecimientos son bastante fáciles de detectar y su frecuencia es muy puntual.

A continuación vamos a realizar un pequeño análisis tanto de los resultados obtenidos cuando el sistema es causal como cuando el sistema no es causal, tal y como se viene realizando a lo largo de esta memoria. Otro aspecto que vamos a tener en cuenta es la ganancia que se utiliza en el altavoz de ruido enmascarador y, que se calcula a partir del STOI en un punto en concreto para poder observar cómo afecta este parámetro al sistema. Para agilizarlo se ha optado por estudiar cuando se hace uso de la ganancia calculada a partir del STOI en el punto central de monitorización o G_{m2} y cuando se hace uso de la ganancia obtenida a partir del STOI en el punto de virtualización o G_v. Hay que destacar que en el sistema real una vez implementado en su entorno de trabajo se va a hacer uso de la segunda ganancia comentada, pues el sistema está pensado para que se enmascare

adecuadamente en el punto en donde la persona va a estar colocada, es decir, el punto de virtualización. El hecho de utilizar ahora la ganancia G_{m2} es para comparar resultados.

La primera gráfica que vamos a estudiar hace referencia a los resultados obtenidos cuando en un sistema causal, aplicamos la ganancia G_{m2}:

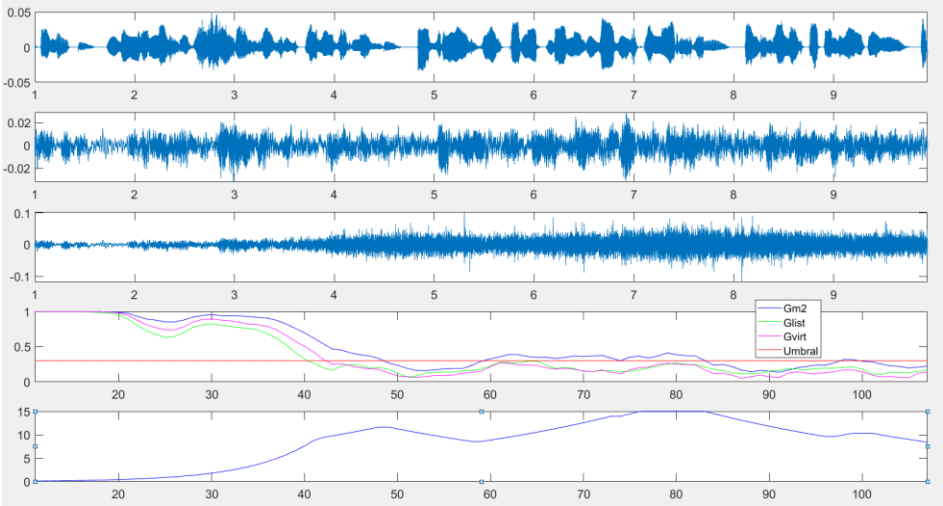

Fig 49. Gráficas obtenidas aplicando G_{m2} en un sistema causal

Las señales representadas en la segunda y tercera gráfica contando desde arriba corresponden con la señal de voz tras realizar la resta acústica que se explica en (2.1.5) y la señal que se captura con el micrófono de monitorización número 2 respectivamente. Se han elegido estas porque son las de mayor interés para este estudio. Podemos observar que la señal de la primera gráfica, que corresponde con la señal de voz que se emite por el altavoz, y la segunda gráfica tienen una forma similar, aunque retrasadas un número de muestras. Lo que podemos afirmar que la resta acústica (2.1.5) se realiza correctamente. Tampoco vemos que la tercera gráfica sea una señal que diverja y las amplitudes de las tres gráficas superiores tienen unos valores lógicos. La señal que se captura por el micrófono debe siempre tener mucho más amplitud ya que se compone de la suma de la señal de voz junto con la del ruido enmascarador multiplicado por la ganancia. Se puede ver en (2.1.3). En cuanto a los valores de STOI, es decir la penúltima gráfica, observamos que el sistema converge. Es otro indicador de que el sistema funciona correctamente. Además podemos ver que el STOI que se obtiene en el punto de monitorización siempre tiene valores superiores a los otros dos que se estan representando, STOI en punto ideal y de virtualización. Se debe a que como nos encontramos con un sistema causal, tanto los puntos ideal como de virtualización quedan más cerca de la fuente de ruido que los micrófonos de monitorización, lo que se traduce en que los micrófonos más cercanos capturan el ruido enmascarador con mayor amplitud, facilitando el enmascaramiento de la voz. Este mismo efecto también se podrá observar siempre que el sistema sea causal. También observamos que los valores de STOI en ideal y en virtualización son muy similares, lo que nos indica de que el proceso de virtualización se realiza satisfactoriamente.

Finalmente, observamos que los valores de STOI en ideal y virtualización quedan bastante por debajo del umbral de STOI. Nos puede hacer una idea de que, aunque se esté enmascarando la voz perfectamente y sea totalmente ininteligible, puede que sea demasiado molesto para trabajar. El sistema, debido al entorno en donde está pensado instalarse, debe encontrar un equilibrio entre enmascarar la voz y ser apto para desarrollar una actividad laboral en una oficina. Esta es la casuística a tener en cuenta de por qué es más apropiado hacer uso de la ganancia en virtualización G_v.

Tras este estudio vamos a comentar los resultados obtenidos cuando el sistema es causal y se hace uso de la ganancia en el punto de virtualización:

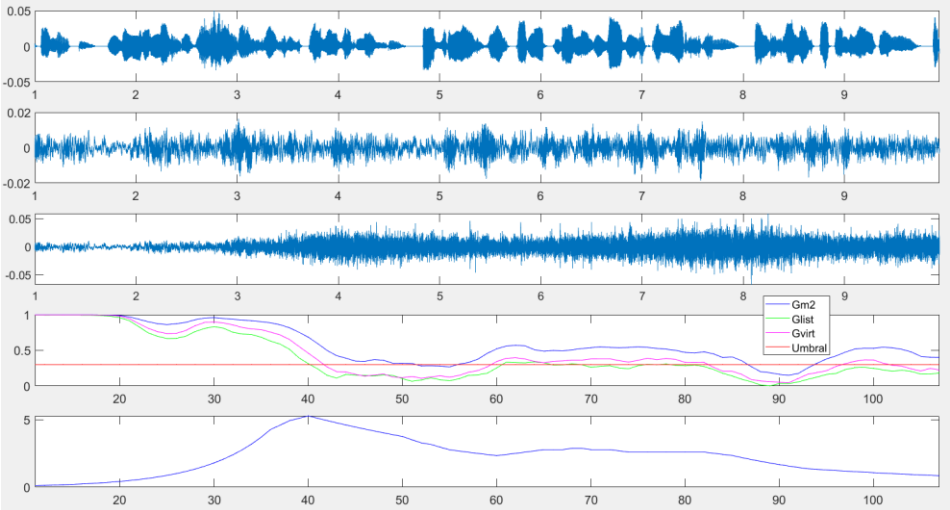

Fig 50. Gráficas obtenidas aplicando G_v en un sistema causal

Lo primero que comprobamos son las gráficas de las señales acústicas de la imagen. Analizamos como se ha hecho anteriormente que la forma de la segunda es similar a la primera, pero desplazada un número de muestras y que la tercera gráfica no diverge y la amplitud de esta es mucho mayor que la de la primera y segunda gráfica. La diferencia más sustancial la vemos en las dos gráficas finales. En la gráfica de los valores de STOI se observa que como el sistema es causal, los valores de STOI calculados en el punto de monitorización quedan por encima de los restantes, pero añadiendo la diferencia de que en comparación con los obtenidos cuando se aplica la ganancia G_{m2}, estos valores son más altos que los de la gráfica STOI cuando se aplica G_{m2}. Como el punto a virtualizar físicamente se encuentra más cerca a la fuente de ruido, no es necesario aplicar una ganancia tan elevada para que la señal de voz sea enmascarada completamente en ese punto. Se puede comparar en la gráfica de abajo que hace referencia a los niveles de ganancia que se aplican en la fuente de ruido en cada caso. Los de la gráfica de G_v son menores que los de la gráfica cuando aplica G_{m2}. A nivel auditivo también se puede comprobar si se escuchan las señales que se obtienen en cada punto. Se percibe un ruido con mucho más volúmen en el punto de virtualización que en el punto de monitorización cuando el sistema es causal, lo que da origen a que la inteligibilidad en el primer punto es más baja que en el segundo. Se llegan a las mismas conclusiones analizando objetivamente los resultados, pero las conclusiones son similares tras escuchar los sonidos resultantes.

En cuanto al sistema no causal, ocurre el efecto contrario que se comenta en las gráficas anteriores en cuanto a valores de STOI en cada punto y ganancias. Ahora la fuente de ruido queda más cerca de los micrófonos de monitorización que del punto de virtualización. Se espera que ahora, los valores de STOI en ideal y virtualización sean superiores que los obtenidos en monitorización. Y podemos verlo a continuación:

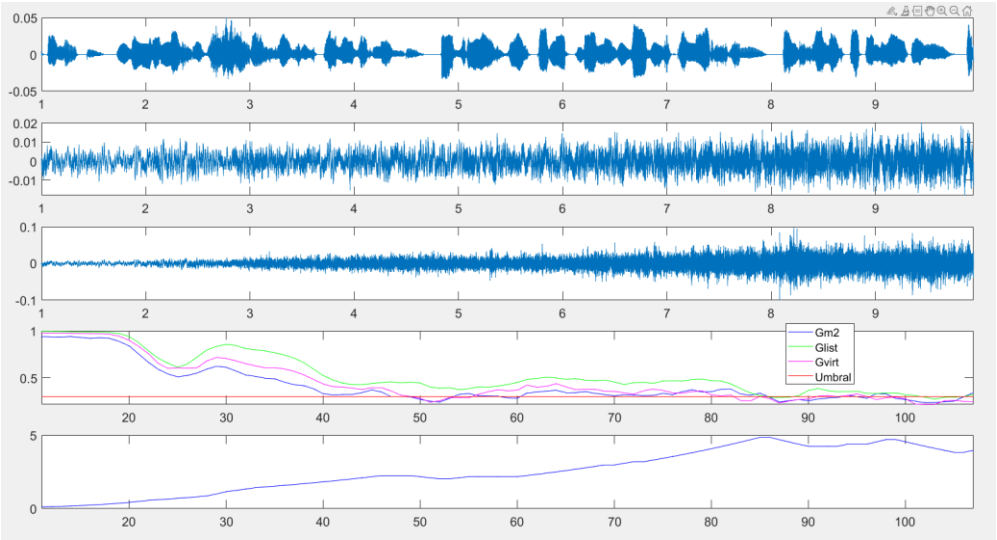

Fig 51. Gráficas obtenidas aplicando G_{m2} en un sistema no causal

También ocurre el efecto contrario en las ganancias, como ahora los micrófonos de monitorización se encuentran mucho más cercanos a la fuente de ruido, se obtienen ganancias menores cuando se aplica G_{m2} que cuando se aplica G_v:

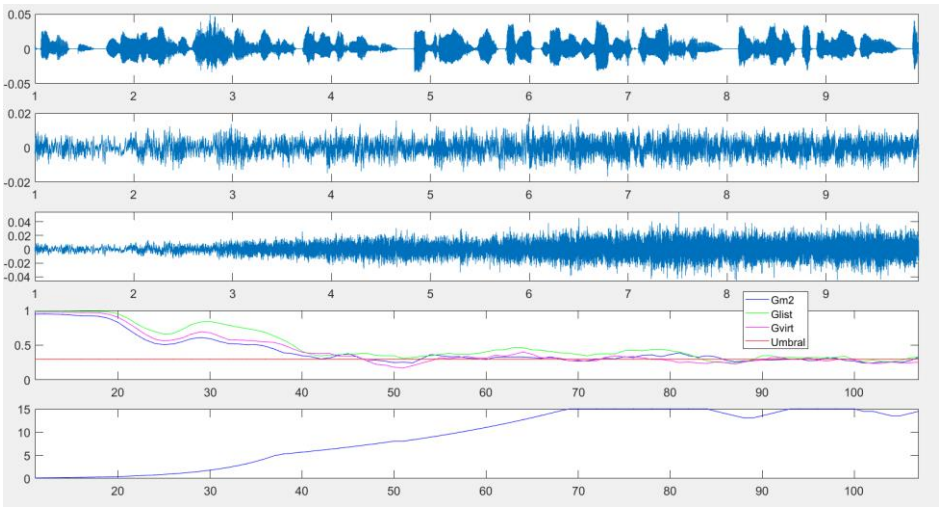

Fig 52. Gráficas obtenidas aplicando G_v en un sistema no causal

Si realizamos el análisis acústico de las señales para un sistema no causal, es decir escuchándolas, también llegamos a la misma conclusión y uno se puede dar cuenta de que efectivamente sucede el efecto contrario al sistema causal.

Finalmente comentar que se podría realizar el estudio haciendo uso de otros ruidos enmascaradores, pero se llegan a las mismas conclusiones. La dos diferencias que se observan es que se observa es que el sistema converge antes o después dependiendo de la naturaleza del ruido enmascradador elegido y, que la ganancia a aplicar en cada ejemplo varía dependiendo de factores como la pontencia o naturaleza del sonido.

Capítulo 7. Bibliografía

[1]. *An Algorithm for Intelligibility Prediction of Time–Frequency Weighted Noisy Speech.* **Cees H. Taal, Richard C. Hendriks, Richard Heusdens, and Jesper Jensen.** 2011, IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING, VOL. 19, NO. 7, págs. 1-12.

[2]. *SUBJECTIVE ANALYSIS OF SPEECH PRIVACY USING SPEECH MASKING IN OPEN-PLAN OFFICES.* **Laura Fuster, Maria de Diego, Gema Piñero, Alberto Gonzalez and Miguel Ferrer.** 2021, Annual Congress of the International Institute of Acoustics and Vibration (IIAV), págs. 1-8

[3]. *A Review of Virtual Sensing Algorithms for Active Noise Control.* **Danielle Moreau, Ben Cazzolato, Anthony Zander and Cornelis Petersen.** 1, Adelaide : Algorithms, 2008, Vol. 1. 1.