

# Hito\_3\_Echo\_Cancellation

January 8, 2025

## 1 Echo Cancellation

### 1.1 Introducción

Para este hito se ha tratado de implementar un código que evite la retroalimentación del audio, es decir, que cuando el audio emitido por los altavoces de un dispositivo vuelva al micrófono del mismo, este no sea reenviado de vuelta al interlocutor que lo produjo o que en caso de hacerlo no viaje una tercera vez.

### 1.2 Solución al problema

De las tres posibilidades que se nos ha explicado, se ha elegido la primera (“s(t) delay and substract solution”).

A continuación se explica paso a paso cómo funciona la implementación.

1. Se empieza enviando un único pulso al buffer.
2. Después, se recoge del buffer y se inicia un hilo para calcular en segundo plano el retraso del eco y poder atenuarlo.
3. Para calcular el retraso (d) y la atenuación necesaria (a), se ha creado un método llamado “estimate\_delay\_and\_attenuation”. Este método se encarga de lo siguiente:
  - Concatena el chunk enviado y recogido del buffer para procesamiento.
  - Se tiene que realizar la correlación entre el chunk recibido y el concatenado, para así ver cuánto tiempo tarda en recibirse el eco por la retroalimentación. Se tiene que hacer para ambos canales, y tras ello se realiza una media para así conseguir un retraso medio por canal.
  - Se calcula el pico más alto de la atenuación de la señal enviada y la recibida, y tras ello se divide la última con la primera, consiguiendo así la atenuación.
  - Por último, al retraso se le suma el tiempo de buffering que se he haya elegido (sea el predefinido, 150 ms, u otro).

### 1.3 Puntos claves del código

- Dentro del código se utiliza un filtro de paso bajo (véase entre las líneas 33-40). La función de dicho filtro en nuestro código consiste en reducir el ruido de las señales que van transmitiéndose entre los dispositivos que participan en la transmisión.
- También se hace uso de un hilo para realizar el cálculo de la atenuación y el retraso para que de esta forma no se vea interrumpido el procesamiento en tiempo real.

## 1.4 Resultados de ejecución

Dependiendo de las características de los dispositivos usados los resultados varían en gran medida. Se han realizado múltiples pruebas entre los miembros del equipo, el mayor caso de éxito se da cuando los integrantes Mary y Manuel se comunican, ya que tienen una comunicación aceptable aunque ocasionalmente si ocurren errores en el audio. Además, en este caso, cabe destacar que si parte del audio vuelve al dispositivo no se genera retroalimentación indefinida.

En las pruebas con Giovanna, ella escucha de manera aceptable al resto de miembros pero por lo general su audio sale gravemente perjudicado, haciendo que el receptor no pueda interpretarlo.

Por parte de Oliver no se han podido realizar tantas pruebas como con el resto de miembros debido a que un transformador que alimenta las comunicaciones de su casa se ha incendiado. Por otro lado, se ha podido verificar que la comunicación con Mary fluctúa, ya que ella casi siempre logra escuchar de manera aceptable a Oliver pero sólo lograba interpretar aproximadamente tres cuartos del audio.

Finalmente, se llega a la conclusión de que dependiendo del sistema operativo en el que se esté realizando la ejecución y el dispositivo que se esté utilizando, los resultados pueden llegar a variar. Esto posiblemente se deba al manejo de audio que implementa cada uno y otras características que influyen en el desempeño del programa.

## 1.5 Bibliografía

Para el desarrollo de esta actividad se ha hecho uso de los apuntes de la asignatura y concretamente se han revisado las fuentes listadas a continuación.

[1] S. Haykin. Adaptive Filter Theory (3rd edition). Prentice Hall, 1995.

[2] J. Kovačević, V.K. Goyal, and M. Vetterli. Fourier and Wavelet Signal Processing. <http://www.fourierandwavelets.org/>, 2013.