

# SISTEMA BASADO EN TECNOLOGÍA 5G DE MONITORIZACIÓN PSICOACÚSTICA DEL PAISAJE SONORO EN SMART CITIES CON OFFLOADING COMPUTACIONAL DINÁMICO EN EL EDGE

Jaume Segura-Garcia<sup>1</sup>, Santiago Felici-Castell<sup>1</sup>, Jose M. Alcaraz Calero<sup>2</sup>, Qi Wang<sup>2</sup>, Jesús López-Ballester<sup>1</sup>, Rafael Fayos-Jordán<sup>1</sup>, Juan J. Pérez-Solano<sup>1</sup>, Miguel Arana-Burgui<sup>3</sup>

<sup>1</sup> Dpt Informàtica, ETSE – Universitat de València

{e-mail: [jsegura@uv.es](mailto:jsegura@uv.es), [felici@uv.es](mailto:felici@uv.es), [jeloba@uv.es](mailto:jeloba@uv.es), [rafajor@uv.es](mailto:rafajor@uv.es), [jiperez@uv.es](mailto:jiperez@uv.es)}

<sup>2</sup> School of Computing, Engineering and Physical Sciences – University of the West of Scotland

<sup>3</sup> INAMAT – Universidad Pública de Navarra, Spain

## Resumen

En las tecnologías de próxima generación, el control de las molestias por ruido ambiental en una Smart City debería ser lo más eficiente posible. Los sistemas IoT sobre tecnologías 5G ofrecen una gran oportunidad para hacer offloading del cómputo en los nodos sensores, ya que proporciona una serie de nuevos conceptos para el cómputo dinámico que las tecnologías anteriores no ofrecían.

En este artículo, se ha implementado un sistema IoT completo basado en tecnología 5G para la monitorización psicoacústica utilizando diferentes opciones para descargar el cómputo a diferentes partes del sistema. Esta descarga se ha realizado mediante el desarrollo de diferentes splittings funcionales de los algoritmos de cálculo de las métricas psicoacústicas. Por último, se muestra una comparación del rendimiento entre los diferentes splittings funcionales y su aplicación con un análisis detallado.

**Palabras clave:** 5G LTE-M1, psicoacústica, monitorización del soundscape, splitting funcional, Smart City.

## Abstract

In the next generation technologies, the environmental noise nuisance monitoring in a Smart City should be as much efficient as possible. 5G IoT systems offer a big opportunity to offload the computation from the sensor nodes, since it provides a series of new concepts for dynamic computing that the previous technologies did not offer.

In this paper, a complete 5G IoT system for psycho-acoustic monitoring has been implemented using different options for offloading computation to different parts of the system. This offloading has been done by developing different functional splittings in the psycho-acoustic metrics algorithms. Finally, a performance comparison among different functional splittings and their implementation are shown with a detailed discussion.

**Keywords:** 5G LTE-M1, psycho-acoustic, soundscape monitoring, functional splitting, Smart City.

**PACS n°.** 43.60.-c, 43.58.-e, 43.66.-x

## 1 Introducción

Actualmente las ciudades medianas y grandes vienen sufriendo altos niveles de ruido, lo que a su vez provoca una enorme molestia subjetiva para los habitantes. Este ruido también tiene un impacto significativo en la salud humana. Es por ello que el Reglamento Europeo "Environmental Noise Directive" (END) [1] se ha centrado en abordar este problema. La directiva obliga a proporcionar a Ciudades Inteligentes (Smart Cities), una monitorización en tiempo real de las métricas clave del paisaje sonoro si la ciudad tiene más de 250000 habitantes.

Por lo tanto, END es la primera iniciativa a nivel de la Unión Europea en esta dirección y ha abordado una cuestión de salud muy importante, pero, según muchos investigadores, tiene un margen de mejora significativo. Para ser concretos, las métricas utilizadas en la definición de END sólo consideran el nivel equivalente del sonido (métrica del sonido Leq), pero no consideran los componentes espectrales de estos sonidos.

Este defecto tiene varias implicaciones para la percepción humana. Por ejemplo, durante el día una máquina expendedora no causará un ruido significativo para la percepción humana, incluso si se trata de un sonido muy intenso, mientras que durante la noche, una simple fuga de un grifo de contacto puede causar una molestia significativa incluso con un bajo nivel sonoro.

Para abordar este problema, diversos investigadores han estado elaborando diferentes métricas sonoras que proporcionan una mejora significativa en la precisión del nivel de molestia desde la perspectiva humana. Todas estas métricas tienen un problema común y bien conocido que se relaciona principalmente con la complejidad computacional asociada al cálculo de dichas métricas. Esta complejidad computacional ha hecho impracticable la creación de sensores de monitorización en tiempo real que proporcionen una valoración precisa del paisaje sonoro de las Smart Cities. Este problema ha sido la principal motivación de este trabajo de investigación.

Las redes 5G se están desplegando actualmente en las ciudades más grandes del mundo y su despliegue continuará extendiéndose a lo largo de los próximos años. 5G es el habilitador de comunicaciones de tipo máquina masiva (mMTC), en las que millones de dispositivos IoT se conectan sin problemas en toda la ciudad para prestar servicios a los ciudadanos. 5G se despliega utilizando una arquitectura de computación móvil en el EDGE (MEC) donde es posible realizar la descarga computacional de las funciones de la red en la última milla de la red, cerca de los usuarios finales, con el fin de cumplir con los requisitos de baja latencia.

La contribución de este trabajo de investigación es múltiple. En primer lugar, se ha diseñado y prototipado un dispositivo IoT con soporte para muestreo de sonido en tiempo real que es capaz de conectar y entregar información en la red 5G. En segundo lugar, se ha propuesto un nuevo diseño de los algoritmos psicoacústicos para el cálculo de las molestias psicoacústicas humanas, basado en el modelo de Zwicker y en la división funcional de los diferentes pasos de procesamiento necesarios para permitir la descarga del procesamiento computacional en el borde de la red 5G. En tercer lugar, se ha llevado a cabo una validación empírica del rendimiento de la solución propuesta en relación a la evaluación del cálculo de todas las métricas en el propio nodo.

Este documento se desarrolla de la siguiente manera: primero la introducción enfoca el problema a abordar y su orientación. La segunda sección trata de establecer un estado del arte con la discusión de diferentes enfoques y el establecimiento de un marco para nuestro enfoque. La tercera sección se centra en la descripción de los aspectos algorítmicos de las métricas psicoacústicas que se relacionan con la molestia psicoacústica y con el algoritmo de cada métrica psicoacústica. La cuarta sección describe el

sistema IoT 5G diseñado para la monitorización del paisaje sonoro centrándose en el nodo como primer elemento y luego en toda la infraestructura 5G. La quinta sección describe el banco de pruebas, las pruebas de validación realizadas y el análisis de rendimientos, considerando también la escalabilidad a una gran Smart City. Finalmente, las conclusiones discuten los logros de nuestro sistema 5G y las pruebas de rendimiento, abordando diferentes líneas para el trabajo futuro.

## 2 Materiales y métodos

### 2.1 Métricas psicoacústicas para la evaluación del paisaje sonoro

La evaluación del paisaje sonoro de la ciudad en función de métricas psicoacústicas es un tema que ha sido abordado utilizando diferentes enfoques, pero el más conocido y utilizado es el modelo de molestia de Zwicker [2], que se centra en cuatro parámetros (es decir loudness, sharpness, roughness y fluctuation strength) para evaluar la molestia psicoacústica, como una combinación no lineal de los parámetros anteriores.

En las siguientes subsecciones describiremos estas métricas y justificaremos, debido a la complejidad computacional de estos parámetros, la necesidad de desarrollar implementaciones con offloading para asignar la potencia computacional fuera del nodo o en diferentes partes del entorno 5G.

### 2.2 Loudness

Esta métrica describe la percepción de la intensidad sonora percibida por los humanos (en Sones como una escala lineal) y representa el loudness subjetivo ( $N$ ) de un sonido. No considera ninguna distinción perceptual entre sonidos "agradables" o "molestos". El procedimiento de cálculo está estandarizado en las normas DIN45631/A1 [3] e ISO 532-1:2017 [4]. Nuestra implementación se basa en la norma DIN45631. Como se muestra en la ecuación 1, el loudness se calcula sumando todos los loudness específicos ( $N(z)$ ) o contribuciones de loudness para cada CB, donde  $z$  identifica el número de CB) multiplicados por el ancho de banda de cada Bark. ( $\Delta z$ ).

En primer lugar, en esta implementación se muestrea la señal de entrada y enventana cada segundo con una ventana de Hanning. Luego se aplica una Transformada Rápida de Fourier (FFT) para obtener los SPLs, que son filtrados con filtros de tercio de octava para obtener estos niveles, denominados  $P(z)$  siendo  $z$  cada Bark. A continuación, estas salidas se procesan según la norma DIN-45631 para calcular el loudness específico, denominado  $N(z)$  por Bark. Cabe notar que para acelerar el cálculo de  $N$ , la conversión de la señal a bandas de 1/3 de octava se hace realizando un filtrado que resulta ser una multiplicación de matrices debido al dominio frecuencial.

$$N = \sum_{z=0}^{24 \text{ Bark}} N(z) \cdot \Delta z$$

Una vez que conocemos  $N$  y el loudness específico, llamado  $N(z)$  por Bark, siguiendo la expresión dada por la ecuación 1, nos permite el cálculo de  $S$  con un costo computacional casi nulo, ya que basta con aplicar la fórmula al loudness específico.

### 2.3 Sharpness

El sharpness ( $S$ ) es una valoración de la percepción sensorial humana de lo desagradable de los sonidos causada por componentes de alta frecuencia y que puede considerarse como un centroide del espectro.

Se mide en acum en una escala lineal y está estandarizada por la norma DIN 45692 [5]. Nuestro modelo utiliza el método definido por Zwicker & Fastl que es en lo que se basa el estándar. Este modelo toma como entrada la sonoridad específica (N(z)) calculada en cualquier método de sonoridad Zwicker compatible (como DIN45631). La ecuación principal se da en la ecuación 2, el denominador principal es la sonoridad total N y la suma del numerador el loudness específico (N(z)) multiplicada por una función de ponderación g(z). Esta función de ponderación aumenta de valor a medida que aumenta la tasa de banda crítica. En nuestro modelo, esta función se aproxima a  $e^{0.171 \cdot z}$ , donde z son Barks. Y el C(1KHz, 60dB) es una constante de calibración para obtener 1 acum con un tono de 1KHz a 60dB.

$$S = C(1\text{KHz}, 60\text{dB}) \frac{\sum_{z=0}^{24 \text{ Bark}} N(z) \cdot g(z) \cdot z \cdot \Delta z}{N}$$

## 2.4 Roughness

El roughness (R) describe la percepción de la fluctuación del sonido incluso cuando N o Leq,T permanecen inalterados. Analiza los efectos con diferentes grados de modulación de frecuencia (alrededor de 70Hz) en cada CB. La unidad básica para R es el Asper. En este caso, este parámetro se describe mejor en ERB. Así, para cada ERB i, g(z<sub>i</sub>) es una función de ponderación arbitraria, m<sub>i</sub> es la profundidad de modulación de cada ERB y k es la correlación cruzada entre las envolventes del ERB con los índices i y i - 2.

El modelo utilizado para la R está definido por V. Jourdes [6] basado en el modelo optimizado de P. Daniel y R. Weber. La característica principal de este modelo es el uso de la escala ERB (en lugar de Barks) por medio de los filtros Gammatone [7].

En esta implementación, la señal de entrada se muestrea y se enventana cada segundo con una ventana de Blackmann, que luego es filtrada por el filtro pasa-banda del oído medio externo. Luego se procesa con bancos de filtros Gammatone para obtener 33 señales de excitación, llamadas e<sub>i</sub>(t), siendo i cada una de ellas ERB. Luego, para cada banda i, calculamos cada valor continuo (DC) llamado h<sub>0i</sub>, así como la envolvente de modulación de cada e<sub>i</sub>(t) llamada hBP<sub>i</sub>(t), obtenida utilizando una función de ponderación de Respuesta de Impulso Infinito (IIR). A partir de estos valores para cada banda, calculamos m que es la profundidad de modulación, definida como el valor de la Raíz Cuadrática Media (RMS) de hBP<sub>i</sub>(t)/h<sub>0i</sub>, que se calibra con Calib(i). Finalmente, realizamos una correlación desplazada entre sus profundidades de modulación para obtener una correlación cruzada k<sub>i</sub>. Finalmente, usando la expresión para R dada en la ecuación 3, calculamos R. C(1KHz, 60dB) es una constante de calibración para obtener 1 asper con un tono de 1 kHz, a 60 dB, 100% modulado a 70 Hz.

$$R = C(1\text{KHz}, 60\text{dB}) \sum_{i=0.5}^{33 \text{ ERB}} (g(z_i) \cdot m_i \cdot k_{i-2})^2$$

## 2.5 Fluctuation Strength

La Fluctuation Strength (F) describe cómo fluctúan los sonidos fuertes o débiles. Depende de la frecuencia y profundidad de las fluctuaciones del loudness, alrededor de 4 Hz en cada ERB. Se mide en Vacils.

Con el fin de acelerar el cálculo de estos parámetros (R y F), los pasos iniciales de este algoritmo, desde la señal de entrada hasta el banco de filtros Gammatone, son iguales y se calculan una vez. Luego, aplicamos la función de ponderación de acuerdo a cada modelo como se explicó anteriormente, y calculamos las profundidades de modulación correspondientes m y los factores de correlación cruzada

k para cada parámetro psicoacústico R y F. En el último paso usamos la ecuación 4 para calcular la Fluctuation Strength.

Además, otra mejora para el cálculo de R y F puede hacerse con una correlación desplazada. Esta función desplaza las señales para ver el desplazamiento producido por el banco de filtros Gammatone. Esto se hace haciendo desplazamientos de la señal de 10 muestras y realizando la correlación para averiguar el valor máximo. Por si acaso, el valor encontrado es 1 o casi 1 (por ejemplo 0.999) dejamos de buscar otros desplazamientos, ya que el máximo ya ha sido alcanzado. C(1KHz, 60dB) es una constante de calibración para obtener 1 vacil con un tono de 1 kHz, a 60 dB, 100% modulado a 4 Hz.

$$FS = C(1\text{KHz}, 60\text{dB}) \sum_{i=0.5}^{33\text{ ERB}} (g(z_i) \cdot m_i \cdot k_{i-2} \cdot k_i)^2$$

### 3 Arquitectura IoT 5G y diseño del sistema

#### 3.1 Infraestructura 5G para la medida del paisaje sonoro en la Smart City

La infraestructura 5G IoT diseñada para la descripción del paisaje sonoro en el contexto de una Smart City, considera los siguientes elementos: el nodo como un dispositivo de monitorización de sonido 5G IoT que tiene los sensores conectados y recoge información, el RAN como interfaz de radio, el EDGE donde se puede aplicar algún tipo de offloading del dispositivo para permitir el ahorro de energía y el CORE donde se recoge y procesa la información. Este esquema es la base de la arquitectura propuesta por el proyecto SELFNET.

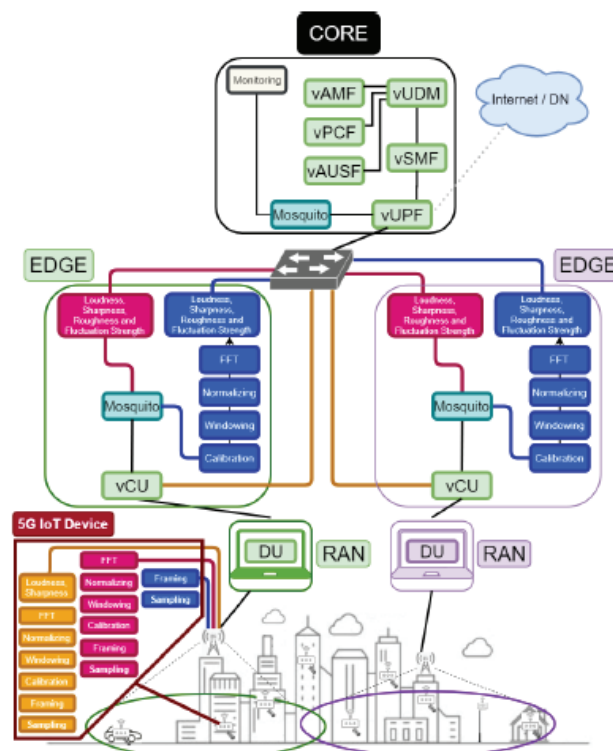


Figura 1 – Diagrama conceptual de la arquitectura de IoT para el sistema de cálculo de las métricas psicoacústicas 5G.

La figura 1 muestra un diagrama conceptual de estos elementos con sus componentes y también considera las diferentes divisiones funcionales para calcular las diferentes métricas psicoacústica consideradas en el modelo de Zwicker para la descripción de la molestia psicoacústica. La siguiente subsección se centrará en la descripción del dispositivo de monitorización de sonido de IoT 5G, que permitirá la descripción de las métricas psicoacústicas, con los otros elementos de la infraestructura de IoT 5G ya descritos en [8].

### 3.2 Dispositivo de monitorización de sonido 5G IoT

La figura 2 muestra el diseño propuesto para el dispositivo IoT 5G. En la parte inferior de la figura se puede ver un dispositivo de hardware IoT basado en el Espressif ESP32 System-on-Chip (SoC) de ultra bajo consumo<sup>1</sup> dotado de una arquitectura de CPU Xtensa. La arquitectura está montada en una placa junto con algunos periféricos. Nuestros prototipos están basados en las placas Fipy de Pycom<sup>2</sup>. La figura no presenta una descripción completa de todas las capacidades disponibles en dicha SoC, placa y periféricos conectados a la placa.

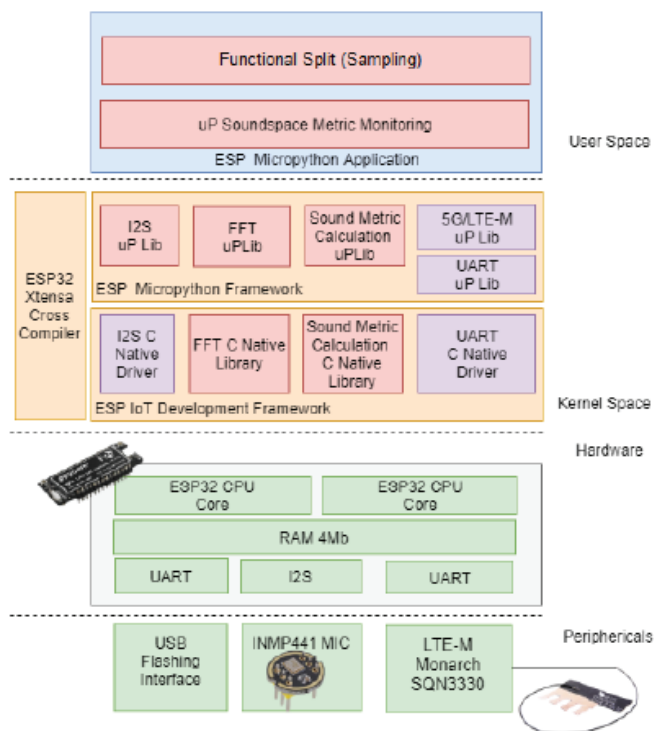


Figura 2 – Diseño de la arquitectura del Dispositivo de Monitorización de Sonido 5G IoT.

Sólo muestra los componentes clave que se utilizan en nuestro diseño. En términos de periféricos, el dispositivo IoT está conectado a un módem SQN3330 que proporciona soporte para los dos modos de conectividad diferentes que proporcionan las infraestructuras 5G para dispositivos IoT: Conectividad LTE-M y NB-IoT. Este módem se conecta al ESP32 SoC mediante una interfaz serie de receptor-transmisor asíncrono universal (UART) de alta velocidad. El dispositivo IoT también está conectado a

<sup>1</sup> ESP32 datasheet is available at

[https://www.espressif.com/sites/default/files/documentation/esp32\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf). (Visitado el 02/09/2019)

<sup>2</sup> El datasheet de Fipy de Pycom está disponible en <https://docs.pycom.io/datasheets/development/fipy/> (Visitado el 02/09/2019)



un micrófono INMP441 que permite el muestreo de sonido hasta 48kHz con una resolución de hasta 14 bits. El micrófono se conecta mediante una interfaz de sonido Inter-IC (I2S) que permite el uso de Direct Memory Access (DMA) para almacenar el sonido muestreado en tiempo real sin intervención de la CPU. Estos dos periféricos permiten escuchar un sonido de buena calidad en un rango de alrededor de 20 metros de diámetro con respecto al micrófono dentro de la ciudad en tiempo real, y también conectarse a la red para realizar el procesamiento de offloading de la señal de sonido y/o entregar las métricas al sistema de monitorización.

En la parte superior del hardware (véase la parte central de la figura 2), se genera una imagen de sistema operativo incrustada para el conjunto de instrucciones de ensamblador de la CPU Xtensa utilizando el compilador cruzado ESP32 de Xtensa y se introduce en el dispositivo IoT utilizando una interfaz USB. El sistema operativo contiene las capacidades básicas para soportar multi-threading, para hacer uso de las capacidades de hardware por medio de controladores como los controladores UART e I2S y para ejecutar procesados.

El puerto I2S proporciona soporte para llevar a cabo las funciones de muestreo que pueden desplegarse en el dispositivo IoT. El resto de las funciones no están directamente soportadas en el espacio del núcleo y por lo tanto ha requerido la generación de una nueva versión del sistema operativo con soporte para todas las funciones restantes. Se agrupan en dos bibliotecas diferentes, una librería para la transformada rápida de Fourier (FFT) que proporciona y la librería de cálculo de métricas de psicoacústicas. Cabe destacar que la Figura 2 tiene cajas con dos colores diferentes, que se utilizará para diferenciar las funciones disponibles en los dispositivos actuales IoT y las funciones completamente desarrolladas desde cero para permitir que este nuevo caso de uso sea factible.

El sistema operativo se ha ampliado con un completo marco de desarrollo dinámico que expone el lenguaje microPython (uP) como una forma de permitir a los desarrolladores configurar las funciones desplegadas en el dispositivo IoT utilizando un lenguaje de alto nivel. Este marco de trabajo de uP se compila en el espacio del núcleo y expone una máquina virtual python para permitir a los programadores desarrollar aplicaciones en el espacio de usuario. Esto proporciona un alto nivel de dinamismo y flexibilidad en la activación y configuración de las funciones soportadas en el espacio del núcleo. Así, se han diseñado varios wrappers para permitir a las aplicaciones uP hacer uso de las funciones soportadas en el kernel. En primer lugar, se ha implementado una biblioteca uP I2S para apoyar el muestreo de sonido y se ha adaptado a ESP32, basada en el trabajo realizado por Mike Teachman<sup>3</sup>. En segundo lugar, se ha diseñado una librería uP FFT, optimizada para su velocidad, y adaptada a ESP32 basada en el trabajo original realizado por Robin Scheibler<sup>4</sup>. En tercer lugar, se ha diseñado una librería de cálculo de métricas de sonido uP, optimizada para funcionar en un dispositivo de baja restricción. Las librerías para tratar con la conectividad a la red 5G ya han sido proporcionadas por los fabricantes de chips y ya han sido portadas a ESP como librería uP.

Como resultado, la flexibilidad y las nuevas funcionalidades proporcionadas en uP permiten desplegar en el dispositivo IoT diferentes modos de realizar la división funcional del proceso de cálculo de las métricas psicoacústicas.

## 4 Validación empírica

En este apartado se han realizado diferentes experimentos para validar el sistema diseñado.

---

<sup>3</sup> <https://github.com/miketeachman/micropython-esp32-i2s-examples> (Visitado el 17/09/2019)

<sup>4</sup> <https://github.com/fakufaku/esp32-fft> (Visited on 17/09/2019)

#### 4.1 Descripción del testbed

El despliegue del sistema se ha realizado de acuerdo con la especificación anterior. Los micrófonos MEMS se han conectado a los nodos FiPy (de Pycom). El firmware implementado ha sido cargado en cada uno de los nodos FiPy. Estos nodos se han conectado a la infraestructura de la red 5G, utilizando LTE-M1.

Dentro de la infraestructura 5G, se ha incluido en el EDGE de la infraestructura 5G una Máquina Virtual (RAM 4GB/4 núcleos) que contiene los analizadores psicoacústicos (para la recepción de muestreo de sonido/enventanado) y se ha instalado en el Cloud otra Máquina Virtual que contiene una plataforma Fiware para la recogida y procesamiento de información IoT. La Figura 3 muestra la infraestructura 5G con los nodo FiPy conectados a micrófonos.SMD.



Figura 3 – Infraestructura 5G con nodos para el cálculo de métricas psicoacústicas

#### 4.2 Análisis del rendimiento del cálculo de la métrica psicoacústica

Para estudiar el rendimiento de nuestra implementación en una plataforma 5G, se ha comparado con una implementación completa de los diferentes parámetros en diferentes plataformas y diferentes lenguajes de programación [9]. La tabla 1 muestra una comparativa de rendimiento en términos de tiempo computacional por cada segundo de grabación de audio entre Matlab y C++/Python, ambos ejecutándose en un ordenador Intel-i7, y la implementación de C++/Python ejecutándose en el RPi3, para los diferentes parámetros psicoacústicos. Para obtener buenos resultados, se han utilizado 100 muestras aleatorias de sonidos diarios de un segundo de duración para obtener valores dentro de un intervalo de confianza del 95%. Estos sonidos fueron tomados de la base de datos Urban-Sound8K [10], que consiste en 8000 audios etiquetados siguiendo una taxonomía similar a la descrita en ISO 12913-2 [11] para sonidos urbanos. Además, como el compilador de C++ tiene la posibilidad de elegir el nivel de optimización que ejecuta, se ha decidido realizar los cálculos con la optimización completada (habilitada por la opción -O3). Durante las pruebas de C++ con la optimización completa habilitada, el uso de la CPU de la RPi3B y RPi3B+ es del 77%, con un núcleo al 100% y los otros tres al 70%. Algo similar sucede con el PC que es un 87% de utilización, con un núcleo al 100% y los otros al 83%. Obsérvese que el código no ha sido programado utilizando hilos. Este es uno de los resultados de usar la biblioteca de Armadillo. En cuanto a las temperaturas, la CPU alcanza los 82.35 grados centígrados en ambas plataformas Rpi (RPi3B y RPi3B+) que están manteniendo constantemente el ordenador a 80 grados.



Los nodos están dentro de un chasis impreso en 3D y se ha utilizado ventilación pasiva para evitar cualquier posible ruido producido por el funcionamiento de los ventiladores. Hay que destacar que en la Tabla 1, las dos últimas columnas muestran el tiempo de cálculo de forma independiente, mientras que en la columna R & F procedemos a combinar ambas para agilizar el cálculo.

Tabla 1 – Comparación del tiempo medio (y desviación Estándar) en segundos entre diferentes dispositivos y lenguajes de programación, con optimización C++.

|            | N                 | S                  | R&F               | R                 | F                 | Total             |
|------------|-------------------|--------------------|-------------------|-------------------|-------------------|-------------------|
| Matlab     | 0.058<br>(9.4e-5) | 0.000<br>(4.4e-8)  | 0.638<br>(2.0e-3) | 0.288<br>(4.8e-4) | 0.404<br>(4.8e-4) | 0.699<br>(2.2e-3) |
| C++/Python | 0.003<br>(2.3e-5) | 0.000<br>(8.0e-12) | 0.235<br>(6.3e-5) | 0.128<br>(3.2e-5) | 0.235<br>(4.2e-5) | 0.238<br>(5.8e-5) |
| Rpi3B      | 0.018<br>(2.5e-5) | 0.000<br>(8.0e-12) | 1.462<br>(5.3e-3) | 0.849<br>(1.8e-2) | 0.742<br>(1.1e-3) | 1.479<br>(6.2e-3) |
| Rpi3B+     | 0.017<br>(2.7e-5) | 0.000<br>(8.0e-12) | 1.389<br>(5.6e-3) | 0.794<br>(1.8e-2) | 0.694<br>(1.2e-3) | 1.406<br>(7.5e-3) |

De este modo, se consigue lograr el cálculo en tiempo real tanto en Matlab como en C++ utilizando un ordenador con procesador Intel-i7. Además, al ejecutarlo con las placas de familia Raspberry Pi superamos este límite de tiempo real. El mejor tiempo de cálculo del rendimiento se muestra mediante la implementación de C++/Python que se ejecuta en el ordenador utilizando la opción de optimización.

### 4.3 Análisis del rendimiento temporal del off-loading funcional 5G en muestreo

Por otra parte, implementando el analizador de los parámetros psicoacústicos en C++/Python realizando un muestreo temporal de audio con un micrófono MEMS INMP441 sobre una placa FiPy de Pycom (con procesador SoC ESP32 de Espressif), se ha realizado un experimento de captura de audio y enviándolo al sistema 5G para procesarlo en el EDGE (más cercano a la RAN).

La Figura 4 muestra el tiempo de registro y inventariado. Como características principales podemos observar que el tiempo de ambos procesos nos permite obtener una traza de procesos que nos llevan a valorarlo como tiempo real al grabar un segundo de audio y inventararlo.

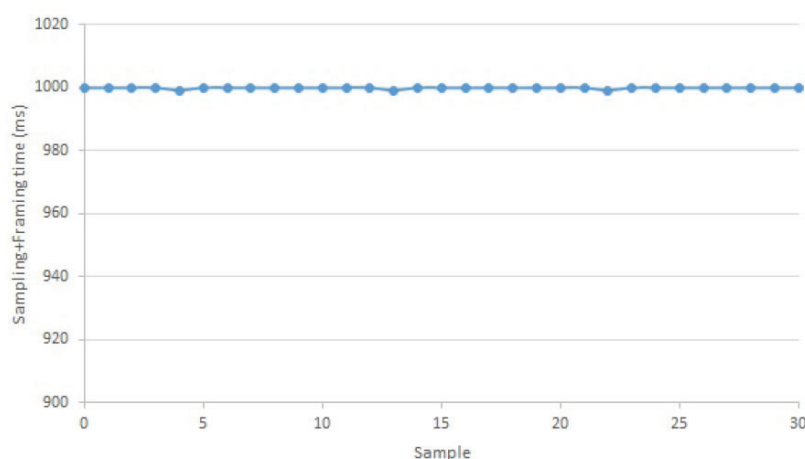


Figura 4 – Representación gráfica del tiempo de registro y inventariado.

La Figura 5 muestra el retraso producido en el tiempo de registro y inventariado. Como características principales podemos observar que el tiempo de ambos procesos nos permite obtener una traza de procesos que nos llevan a valorarlo como tiempo real al grabar un segundo de audio y inventararlo.

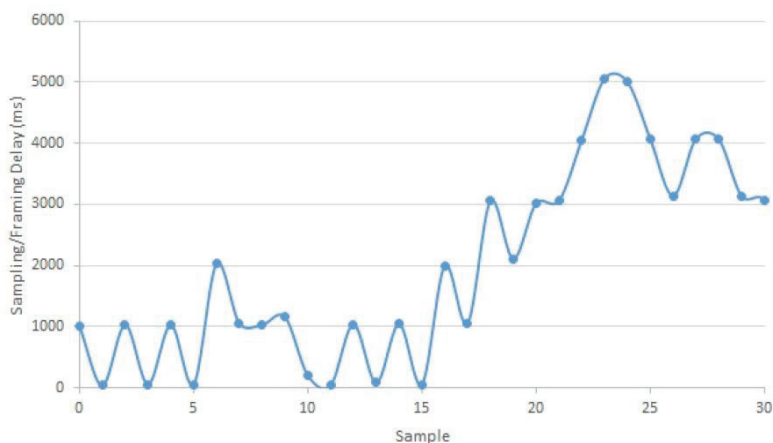


Figura 5 – Representación gráfica del retraso producido por el tiempo de registro y inventariado.

## 5 Conclusiones

En este trabajo se ha diseñado, prototipado y desplegado un piloto de una red de sensores IoT basada en un conjunto de nodos distribuidos capaces de monitorizar la molestia psicoacústica en tiempo real. Este modelo es uno de los más utilizados y aceptados por la comunidad investigadora. Sin embargo, como se muestra en el documento, requiere un alto coste computacional debido a la complejidad del análisis y del procesamiento de la señal requerida.

Debido a esta complejidad, si queremos realizar esta monitorización en una Red de Sensores Acústicos Inalámbricos (WASN) usando plataformas de bajo coste del tipo de la familia Raspberry Pi (Rpi3B o 3B+). Por este motivo, hemos mostrado una implementación eficiente y optimizada del código de este modelo que permite la estimación de una molestia subjetiva, recientemente recogida como indicador psicoacústico por la ISO 12913.

Por otra parte, se ha realizado una implementación del analizador en el EDGE de un sistema 5G y se ha implementado el offloading computacional mediante el muestreo en una placa FiPy de Pycom (basada en ESP32) que muestrea y inventana audio (en períodos de 1 s) y envía usando NB-IoT/LTE-M1 al EDGE para calcular estos parámetros psicoacústicos. Observamos que llegamos a obtener tiempo real en el cálculo de los parámetros en el EDGE, reduciendo este tiempo de cálculo en 0.3s de promedio.

Este trabajo es una primera etapa de desarrollo. Para el trabajo futuro, hemos considerado diferentes temas: extender la división funcional, uso de redes densas con 5G (comunicaciones NB-IoT/LTE-M1 o LoRa para entornos exteriores), etc.

## Agradecimientos

Esta investigación ha sido parcialmente financiada por el Ministerio de Innovación y Economía (ref.: BIA2016-76957-C3-1-R y BIA2016-76957-C3-3-R, cofinanciado con fondos FEDER) y por la Comisión Europea con la Grant H2020-ICT-2016-2/761913 SLICENET (End-to-End Cognitive

Network Slicing and Slice Management Framework in Virtualised Multi-Domain, Multi-Tenant 5G Networks). También por la Generalitat Valenciana, con las becas BEST/2020/117 y AEST/2020/048.

## Referencias

- [1] Directive of the European Parliament. 2002. Directive 2002/49/EC of the European Parliament and of the Council of 25 June 2002 relating to the assessment and management of environmental noise. , 12–25 pages.
- [2] H. Fastl and E. Zwicker. 2007. *Psychoacoustics: Facts and Models*. Springer.
- [3] DIN45631/A1:2010. 2010. Calculation of loudness level and loudness from the sound spectrum - Zwicker method - Amendment 1: Calculation of the loudness of time-variant sound. Standard. Deutsches Institut Fur Normung E.V. (German National Standard), Beuth Verlag GmbH, D.
- [4] ISO532-1:2017(E). 2017. Acoustics – Methods for calculating loudness – Part 1: Zwicker method. Standard. International Organization for Standardization, Geneva, CH.
- [5] DIN45692/A1:2009. 2009. Measurement technique for the simulation of the auditory sensation of sharpness . Standard. Deutsches Institut Fur Normung E.V. (German National Standard), Beuth Verlag GmbH, D.
- [6] V.J.P Jourdes. 2004. Estimation of Perceived Roughness. Student Thesis: Master. <https://research.tue.nl/en/studentTheses/estimation-of-perceived-roughness>
- [7] Ronald W. Schafer Oppenheim, Alan V. and John R. Buck. 1999. *Discrete-Time Signal Processing*. Upper Saddle River (NJ): Prentice Hall International
- [8] Enrique Chirivella-Perez, Ricardo Marco-Alaez, Alba Hita, Ana Serrano, Jose M. Alcaraz Calero, Qi Wang, Pedro M. Neves, Giacomo Bernini, Konstantinos Koutsopoulos, Manuel Gil Pérez, Gregorio Martínez Pérez, Maria João Barros, and Anastasius Gavras. 2019. SELFNET 5G mobile edge computing infrastructure: Design and prototyping. *Software: Practice and Experience* 0, 0 (2019), 1–16. <https://doi.org/10.1002/spe.2681>  
arXiv:<https://onlinelibrary.wiley.com/doi/pdf/10.1002/spe.2681>
- [9] A. Pastor-Aparicio, J. Segura-Garcia, J. Lopez-Ballester, S. Felici-Castell, M. García-Pineda, and J. J. Pérez-Solano. 2020. Psychoacoustic Annoyance Implementation With Wireless Acoustic Sensor Networks for Monitoring in Smart Cities. *IEEE Internet of Things Journal* 7, 1 (Jan 2020), 128–136. <https://doi.org/10.1109/JIOT.2019.2946971>
- [10] Justin Salamon, Christopher Jacoby, and Juan Pablo Bello. 2014. A Dataset and Taxonomy for Urban Sound Research. In *Proceedings of the 22Nd ACM International Conference on Multimedia (MM '14)*. ACM, New York, NY, USA, 1041–1044. <https://doi.org/10.1145/2647868.2655045>
- [11] ISO 12913-2:2018. 2018. Acoustics — Soundscape — Part 2: Data collection and reporting requirements.