

# Desarrollo de un Sistema de Localización y Seguimiento en Tiempo Real de una Fuente Sonora

Martín Moya, Rodrigo Marcuzzi y Leandro Soria  
Supervisores: Dr. Lucas Terissi y Dr. Juan Carlos Gomez

Laboratorio de Sistemas Dinámicos y Procesamiento de la Información  
Cátedra de Sistemas y Señales I  
Facultad de Ciencias Exactas, Ingeniería y Agrimensura  
Universidad Nacional de Rosario  
CIFASIS-CONICET  
moyamartin1@gmail.com  
rodrigomarcuzzi@gmail.com  
leandrosoria@hotmail.com  
lterissi@fceia.unr.edu.ar  
jcgomez@fceia.unr.edu.ar

**Resumen** En este trabajo se describe el desarrollo e implementación de un sistema de localización y seguimiento en tiempo real de una fuente sonora implementado sobre un sistema embebido. Para ello, se diseñó una placa de sonido y se programó un algoritmo basado en técnicas de procesamiento digital de señales para realizar la estima del ángulo de la fuente sonora sobre una Raspberry Pi. Por último, se realizó un control sobre un servomotor para realizar el seguimiento en tiempo real de la fuente sonora, en base a la estima del ángulo obtenida anteriormente.

**Palabras Clave:** Procesamiento Digital de Señales, Localización de Fuente Sonora, Raspberry Pi, Sistemas Embebidos

## 1. Introducción

La localización de fuentes sonoras es uno de los problemas fundamentales en distintas áreas, incluyendo sonares, teleconferencia o videoconferencia, aplicaciones de localización de teléfonos celulares, sistemas de navegación y vigilancia [1], entre otras.

En la bibliografía disponible se pueden encontrar varios métodos para lograr el objetivo, entre ellos, el uso de SNN (*Spiking Neural Network*) [2], PNN (*Pulsed Neural Networks*) [3], MCAL (*Multichannel Acoustic Localization*) [4], HRIR (*Head-related Impulse Response*) [5] y GCC (*Generalized Cross-correlation*) [6].

En este trabajo se implementó una de las técnicas más clásicas de SSL (*Sound Source Localization*) sobre un sistema embebido, con la finalidad de realizar un

seguimiento en tiempo real de una fuente sonora, cuyo fundamento recae en la TDE (*Time Delay Estimation*), también conocido como TDOA (*Time Difference of Arrival*) en un arreglo de micrófonos haciendo uso de la GCC-PHAT (*Generalized Cross Correlation with Phase Transform*) como se implementó en [7,8] .

El sistema está compuesto por un arreglo de micrófonos, un servo motor y un sistema de cómputo que, a partir de la información obtenida de los micrófonos, puede calcular el ángulo de incidencia de la fuente sonora respecto al plano que conforman los mismos. Con este resultado se controla el servomotor para que los micrófonos se orienten en dirección a la fuente sonora.

La finalidad de este sistema es obtener un punto de partida para el desarrollo de un sistema de localización de fuentes sonoras en tres dimensiones y un prototipo funcional para la investigación académica en el área del procesamiento digital de señales.

El resto de este trabajo está organizado de la siguiente forma. En la sección 2 se realiza una introducción teórica planteando el problema de detección del ángulo de una fuente sonora. En la sección 3 se muestra el desarrollo de la solución propuesta, que abarca el diseño de una placa de sonido y de una estrategia para controlar el servomotor. Finalmente, en la sección 4 se presentan las conclusiones del trabajo.

## 2. Planteo del problema

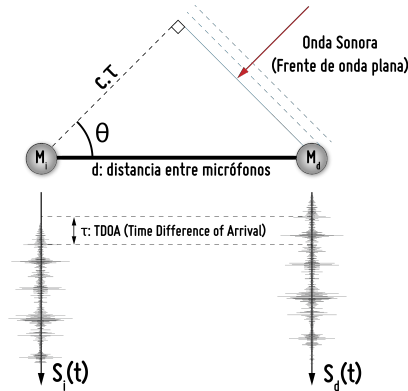
En la Figura 1 se representa un arreglo de dos micrófonos. Éstos se encuentran separados una distancia  $d$ . Sobre ellos incide un frente de onda (el cual se supondrá plano como hipótesis simplificatoria) a un determinado ángulo  $\theta$  que deberá ser calculado. Suponiendo que el frente de onda se desplaza a una velocidad  $c = 343,2 \text{ m/s}$ , se puede verificar que:

$$\cos(\theta) = \frac{c \cdot \tau}{d} \quad (1)$$

donde  $\tau$  representa el retardo de arribo (TDOA) de la onda sonora en el micrófono izquierdo respecto al micrófono derecho. De este modo, despejando  $\theta$  de la Ecuación (1), se halla una expresión para determinar el ángulo de arribo a partir del TDOA:

$$\theta = \cos^{-1} \left( \frac{c \cdot \tau}{d} \right) \quad (2)$$

La dificultad del obtener el ángulo reside en tener una correcta estimación del tiempo de retardo entre los micrófonos. La forma más estándar de estimar este retardo se basa en un análisis de correlación cruzada de señales, utilizada para



**Figura 1.** Descripción del sistema utilizado para determinar el ángulo de incidencia de una onda sonora.

medir el grado de similitud entre dos señales. Se define la correlación discreta entre dos señales discretas  $x(n)$  e  $y(n)$  de la siguiente forma:

$$r_{xy}(\ell) = \sum_{n=-\infty}^{\infty} x(n) \cdot y(n - \ell); \quad \ell = 0, \pm 1, \pm 2, \dots \quad (3)$$

El índice  $\ell$  es un parámetro de desplazamiento temporal, que determina el tiempo relativo entre ambas señales. El cálculo de esta secuencia resulta impreciso en ambientes sumidos en ruido, además de que requiere una carga computacional relativamente elevada. Como alternativa, se decide aprovechar una propiedad que vincula a una secuencia de correlación cruzada entre dos señales con sus respectivas transformadas de Fourier en Tiempo Discreto (DFT):

$$\mathcal{F}\{r_{xy}(\ell)\} = X(f)\bar{Y}(f), \quad (4)$$

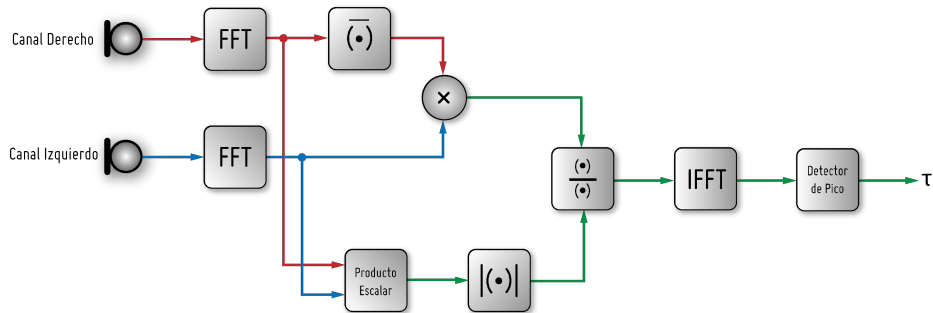
donde  $\mathcal{F}\{\cdot\}$  representa la transformada de Fourier,  $X(f)$  es la transformada de Fourier de la señal  $x(t)$  e  $\bar{Y}(f)$  es el conjugado complejo de la transformada de Fourier de la señal  $y(t)$ .

En [7] se utilizó una técnica basada en correlación de señales en el dominio frecuencial llamada GCC-PHAT cuyos resultados dentro de ambientes ruidosos fueron satisfactorios. Ésta se define de la siguiente forma:

$$G_{PH}(f) = \frac{X_L(f) \cdot \bar{X}_R(f)}{|X_L(f)| \cdot |X_R(f)|}, \quad (5)$$

donde  $X_L(f)$  representa la FFT (*Fast Fourier Transform*) de las muestras del canal izquierdo y  $\bar{X}_R(f)$  es el conjugado de la FFT de las muestras del canal derecho. La elección de este algoritmo se basa en la robustez del mismo ante ruido en la señal de entrada y su bajo costo computacional en comparación a

implementar la correlación cruzada en el dominio temporal (puede demostrarse que el cálculo de la FFT de una señal posee complejidad  $\mathcal{O}(n \log n)$ ). La técnica GCC-PHAT consiste en blanquear (*whiten*) las señales de entrada (es decir, extender su espectro frecuencial y que se asemeje más a ruido blanco). Posteriormente, se realiza una ponderación sobre las mismas. El diagrama de bloques de un procesador GCC se puede observar en la Figura 2.

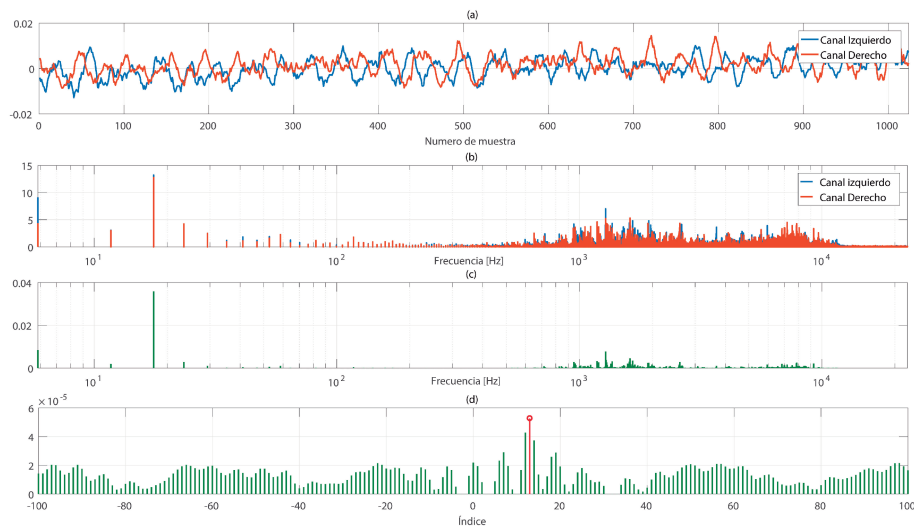


**Figura 2.** Diagrama de bloques de un procesador GCC.

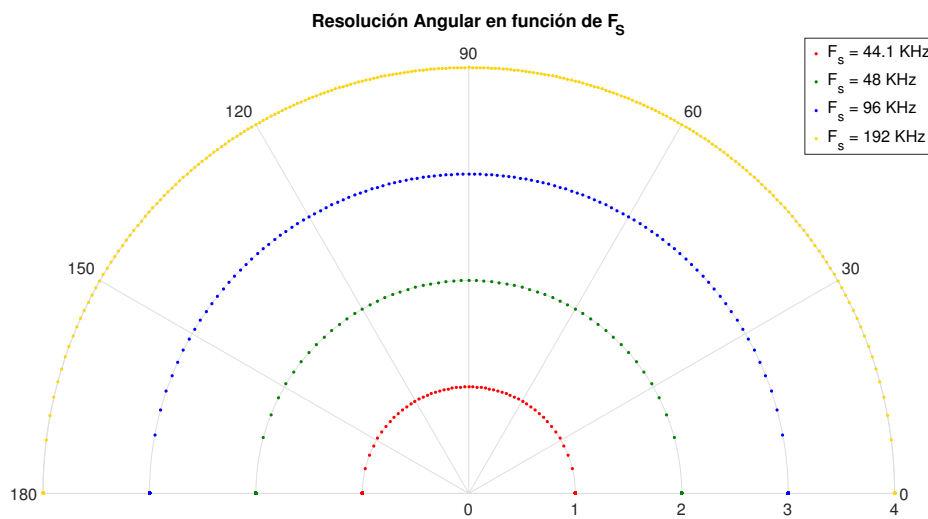
Una vez obtenido el resultado de la Ecuación (5), se realiza su IFFT (*Inverse Fast Fourier Transform*) para obtener la secuencia de correlación cruzada, sobre la cual se detecta el pico máximo del vector de muestras resultante ( $\ell_{max}$ ). Este pico corresponde al TDOA entre las dos señales provenientes de los micrófonos y, finalmente, se obtiene la estima del ángulo aplicando una variante de la ecuación (2), en donde se tiene en cuenta la frecuencia de muestreo de las señales.

$$\theta = \cos^{-1} \left( \frac{c \cdot \ell_{max}}{F_s \cdot d} \right) \quad (6)$$

En la Figura 3 se muestra en forma gráfica cómo se procesan las señales hasta obtener el índice correspondiente a su retardo temporal. En (a) se muestra las señales captadas por los micrófonos, en (b) el espectro de las señales muestreadas, en (c) el espectro frecuencial de la correlación pesada entre las señales y, finalmente, en (d) la antitransformada de (c). Debe tenerse en cuenta que como se trabaja en un dominio discreto (es decir, las señales no son continuas si no que son muestreadas a una frecuencia determinada), la estima del ángulo de arriba también será discreta. Debe notarse que dado que aparece la inversa del coseno en (6), la resolución angular del sistema no será uniforme. La incidencia de la frecuencia de muestreo sobre el resultado se puede observar en la Figura 4, donde se puede observar que a medida que aumenta la frecuencia de muestreo la cantidad de puntos se incrementa, lo que se traduce en una mejora de la resolución angular. En este caso, se utilizó una frecuencia de muestreo de 48KHz donde la resolución angular resulta aceptable para el sistema.



**Figura 3.** Obtención del TDOA entre dos señales. De arriba a abajo: (a): Señales captadas por los micrófonos (dominio temporal). (b): Espectro de las señales muestreadas. (c): Espectro frecuencial de la correlación pesada entre las señales. (d): Antitransformada de (c). En ella el pico máximo (marcado en rojo) se corresponde con el TDOA.

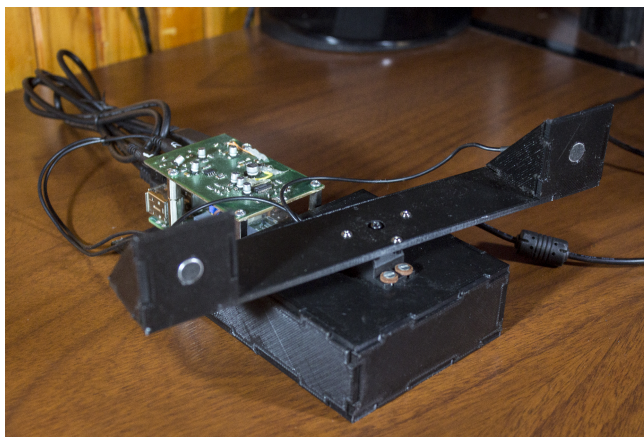


**Figura 4.** Resolución angular en función de la frecuencia de muestreo.

### 3. Solución Propuesta

Una vez introducido el problema de forma teórica, se propuso realizar un prototipo para verificar el funcionamiento del algoritmo. Se diseñó una platafor-

ma conformada por un brazo móvil donde se disponen dos micrófonos de tipo *electret*. La rotación de este brazo se logró con ayuda de un servomotor. Además, se diseñó una base donde montar el hardware utilizado. Todas las piezas fueron impresas en 3D y ensambladas mediante ranuras y tornillos. En la Figura 5 se muestra una foto del conjunto final.



**Figura 5.** Foto del sistema completo.

Con respecto al hardware, se utilizó una Raspberry Pi modelo 3B+ como unidad de cómputo, y una placa de sonido estéreo realizada de forma casera. La Raspberry Pi es la encargada de correr los algoritmos de procesamiento de señales y controlar finalmente al servo.

### 3.1. Placa de Sonido

La primera etapa de la solución consiste en obtener la información proveniente de la fuente sonora. Para ello se diseñó una placa de sonido, cuya PCB se puede observar en la Figura 6, basada en el circuito integrado PCM2904 del fabricante *Texas Instruments* cuyas características, obtenidas a partir de las hojas de datos [9], son:

- CODEC de audio a interfaz USB-HID.
- ADC y DAC estéreo tipo Delta-Sigma de 16 bits, operando a 48KHz.
- Generador de clock *On-chip* con un cristal de 12MHz.
- Drivers compatibles con Linux y Windows.

El CODEC se encarga de transformar la señal proveniente de los micrófonos en un secuencia de bits mediante PCM (*Pulse Code Modulation*) y empaquetar las señales muestreadas en una trama apta para la comunicación a través de USB. Previo al procesamiento digital, se realiza una adaptación eléctrica de las señales captadas por los micrófonos.

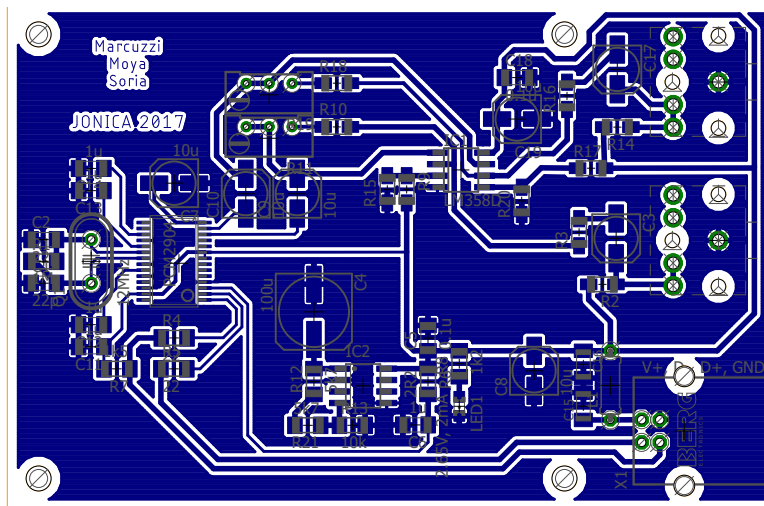


Figura 6. Layout del circuito adquisidor

### 3.2. Control del Servo

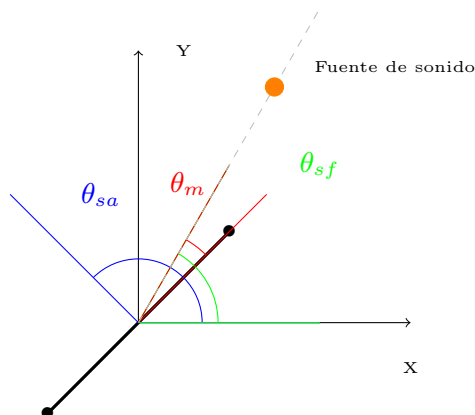
Una vez que se estimó el ángulo donde se encuentra la fuente sonora, es necesario alinear el brazo de modo que éste quede perpendicular a la fuente. Para ello, es necesario conocer la posición donde se encuentra el brazo en el instante anterior. En la figura 7 se muestra un esquema representando la situación planteada.  $\theta_{sa}$  representa el ángulo del servo en el instante previo a la detección,  $\theta_m$  representa el ángulo estimado por el algoritmo, y  $\theta_{sf}$  representa el ángulo al que debe posicionarse el servo para que el brazo quede perpendicular a la fuente. Debe tenerse en cuenta que  $\theta_{sa}$  y  $\theta_{sf}$  están referido a los ejes de coordenadas, mientras que  $\theta_m$  esta referido a la posición del brazo.

Puede concluirse que el ángulo al que se debe orientar el servo resulta ser:

$$\theta_{sf} = \theta_m + \theta_{sa} - 90^\circ \quad (7)$$

Obtenido el ángulo de giro del mismo, la unidad de cómputo se encarga de modificar el ciclo de trabajo de la señal PWM del servo de modo que los micrófonos se alineen con la dirección a la fuente sonora. Debido a que la Raspberry Pi corre una distribución de Linux y este sistema operativo no es determinístico, la generación de la señal PWM presentaba fluctuaciones en su ciclo de trabajo, lo que se traduce en la presencia de vibraciones en el servo. Para solucionar este inconveniente fue necesario utilizar una librería que trabaja a nivel de Kernel de los drivers para el control de la GPIO<sup>1</sup> de la Raspberry Pi, obteniendo una señal PWM estable.

<sup>1</sup> GPIO: General Purpose Input/Output.



**Figura 7.** Estrategia de control del servo.  $\theta_{sa}$  (Azul): ángulo actual del servo.  $\theta_m$  (Rojo): ángulo estimado por los micrófonos.  $\theta_{sf}$  (Verde): ángulo final del servo.

## 4. Conclusiones

En este trabajo se expuso la construcción de un sistema de localización de una fuente sonora en tiempo real mediante procesamiento digital de señales. Se diseñó hardware capaz de digitalizar señales tomadas por micrófonos. Posteriormente, se desarrolló software capaz de acceder a este hardware y aplicar en forma algorítmica las soluciones teóricas planteadas en la Sección 2. Además de verificar el correcto funcionamiento del algoritmo propuesto, se logró obtener un sistema capaz de funcionar en tiempo real, obteniendo resultados satisfactorios.

### 4.1. Futuras Mejoras a Realizar

Se propone como mejora a futuro implementar una placa de sonido basada en un CODEC que soporte más canales y tenga mayor frecuencia de muestreo. De este modo se podrá tener una mayor resolución espacial, y en el caso que se agregue un actuador adicional, realizar una detección y seguimiento tridimensional de fuentes sonoras.

Por otro lado, se buscará desarrollar un sistema embebido con un firmware basado en un RTOS, de modo de ejecutar únicamente las tareas propias del algoritmo en forma determinística, asegurando de esta forma comportamiento más veloz del dispositivo.

Por último, con el objetivo de buscar mayor fluidez en el movimiento del servo, se puede implementar una estrategia de control sobre el mismo utilizando un encoder rotativo u otro sensor (del tipo inercial como un acelerómetro o giróscopo, por ejemplo) para sensar la posición angular del servo.



## Referencias

1. Seunghun Jin, Dongkun Kim, Hyung Soon Kim, Chang Hoon Lee, Jong Suk Choi, and Jae Wook Jeon. Real-time Sound Source Localization System based on FPGA.
2. Mohammad Mahdi Faraji, Saeed Bagheri Shouraki, and Ensieh Iranmehr. 2015. Spiking Neural Network for Sound Localization Using Microphone Array. Presented at 23rd Iranian Conference on Electrical Engineering.
3. Kaname Iwasa, Mauricio Kugler, Susmu Kuroyanagi and Akira Iwata, "A Sound Localization and Recognition System using Pulsed Neural Networks on FPGA," in Proceeding of International Joint Conference on Neural Networks, Orlando, Florida, USA, Agosto 12-17, 2007
4. Ivan Aleksic, Zeljko Hocenski, Petar Horvat. 2010, May. Acoustic Localization based on FPGA. Presented at MPIRO. (2010)
5. Giwoo Park, Pinaki S. Chanda, and Tae Ik Kang. Implementation of real-time 3-D positioning sound synthesis algorithm for a handheld device. Presented at ICACT 2006. pp 1493-1496.
6. Duy Nguyen, Parham Aarabi, and Ali Sheikholeslami. "Real-Time Sound Localization Using Field-Programmable Gate Arrays", in Proceeding of IEEE International Conference on Acoustics, Speech & Signal Processing, Hong Kong.
7. Jin Fu Liou, Junal Joshi and Gaurang Vador: Real-Time Sound Source Localization. Project Report for EE 586
8. Yushi Zhang and Waleed H. Abdulla: A comparative study of Time-Delay Estimation Techniques Using Microphone Arrays. School Engineering Report No. 619. (2005)
9. Texas Instruments, PCM2904/6 Stereo Audio Codec With USB Interface, Single-Ended Analog Input/Output and S/PDIF (2002)