Scientific Research Publishing

Journal of Computer and Communications, 2017, 5, 60-83
http://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

# BigEar: Ubiquitous Wireless Low-Budget Speech Capturing Interface

**Stefano Gorla, Sara Comai, Andrea Masciadri, Fabio Salice**

Department of Electronics, Information and Bioengineering of the Politecnico di Milano, Como, Italy
Email: stefano.gorla@mail.polimi.it, sara.comai@polimi.it, andrea.masciadri@polimi.it, fabio.salice@polimi.it

**How to cite this paper:** Gorla, S., Comai, S., Masciadri, A. and Salice, F. (2017) BigEar: Ubiquitous Wireless Low-Budget Speech Capturing Interface. *Journal of Computer and Communications*, **5**, 60-83.
https://doi.org/10.4236/jcc.2017.54005

**Received:** January 26, 2017
**Accepted:** March 11, 2017
**Published:** March 14, 2017





## Abstract

This article presents BigEar, a wireless low-cost speech capturing interface that aims to realize unobtrusive and transparent context-aware vocal interaction for home automation. The speech recognition process implemented in BigEar system considers noise sources including possible holes in the reconstructed audio stream and tries to overcome them by means of inexactness toleration mechanisms to improve intelligibility of the reconstructed signal. Key contribution of this work is the use of extremely low cost devices to realize a modular flexible and real-time wireless sensor network. On-field implementation and experiments show that the proposed solution can perform real-time speech reconstruction, while listening tests confirm the intelligibility of the reconstructed signal.

## Keywords

Wireless Sensor Networks, Speech Capture, Degraded Speech Recognition, Ubiquitous Systems, Low Cost Architectures

## 1. Introduction

The ageing of world's population will raise the demand and challenges of elderly care in coming years. Based on a study of the US census, the number of people aged over 65 will increase by 101 percent between 2000 and 2030, at a rate of 2.3 percent each year; during that same period, the number of family members who can provide support for them will increase by only 25 percent, at a rate of 0.8 percent each year. Several approaches have been devised to deal with the needs of older people proactively.

Assistive domotics represents a relatively recent effort in this direction addressing the needs of people with disability, older persons, and people with little or no technical affinity, and offers new levels of safety, security and comfort, and thereby the chance to prolong their safe staying at home.

The BRIDGe[1] (Behaviour dRift compensation for autonomous InDependent livinG) project [1], carried out at Politecnico di Milano-Polo di Como, aims to build strong connections between a person living independently at home and his/her social environment (family, caregivers, social services) by implementing a system that provides focused interventions according to the user's needs.

BRIDGe addresses the needs of people with mild cognitive or physical impairments and, more generally, fragile people whose weakness threatens their autonomy, health or other important aspects of their life. Fragile people need mutual reassurance: they typically want to be independent and autonomous, but they also know that often somebody else must be present to help them.

BRIDGe's core is a wireless sensor-actuator network that supports house control and user behavior detection through a rich and flexible communication system between the person and his/her social environment, aiming at reassuring both the family and the user. BRIDGe is based on a modular architecture that can be easily configured to satisfy the single user needs, including: house control e.g., lighting and shutter control; home appliance monitoring for user activity recognition and energy consumption measurements purposes; presence detection, *i.e.* identifying the presence of people in specific areas of the house; indoor localization along with status (moving, sitting, falling, and so on); event- and status-based information transmission to inform caregivers promptly about specific events, such as when a fall is detected.

Target of this work is to model, realize and implement a distributed audio acquisition system called BigEar (uBiquitous wIreless low-budGet spEech cApturing inteRface) to support vocal interaction between the user and the assistive environment, for example, to vocally control some parts of a dwelling like lights, doors, etc. BigEar has been built according to the following Wireless Sensor Network [2] requirements:

The adopted technology (hardware and software) has to consider the economical possibilities of people.

The absence of power and/or signal cables is a strong requirement in order to lower costs for house adaptation. Moreover, wireless systems can ensure a higher degree of flexibility and configurability than wired systems.

The key for pervasiveness is distributed computing [3], an interaction model in which the devices concur in building a result whose information content is more than the sum of single contributions. Moreover, sensors are completely independent and an eventual failure will not completely compromise the result of the sensor network collaboration.

The system should be implemented using a modular approach in order to be scalable and quickly configurable to match the environment characteristics and the user's needs.

Speech recognition should be immediate, so speed of processing is a crucial requirement in order to give the user an immediate feedback; assistive domotic interaction has to be as fast as possible.

[1]http://atg.deib.polimi.it/projects/atg.deib.polimi.it/projects.

The proposed system consists of different modules that take into account the acquisition environment with its acoustic characteristics and the behavior of the sensor-network model. Such modules have been simulated to investigate the archiecture capabilities of the system. Then, a Reconstruction Algorithm reconstructing the audio signal starting from the audio packets received by the microphones has been developed.

This work is organized as follows: after a brief analysis of existent solutions in literature (Section 2), in Section 3 the architecture of BigEar system is described, focusing on the characterization of the context of use, on the features of the prototyping board, and on the communication protocol among the components of the system. Section 4 describes the speech acquisition model that has been simulated to define the working parameters before the realization of the system and the real-world implementation. Section 5 explains operating principles of the BigEar Reconstruction Algorithm, focusing on crucial aspects such as energy compensation, time-delay analysis and streams superposition. The audio data captured by means of the real-world prototype have been compared with the ones generated by the simulated model, and results of this comparison are discussed in Section 6. Speed of processing and the quality of the reconstructed speech signal are evaluated in Section 7. Finally, in Section 8 final remarks conclude the paper and look out over the future works.

## 2. Related Work

Different solutions have been proposed in the literature exploiting audio in smart homes, briefly described in the following subsections.

### 2.1. Sweet-Home Project

The Sweet-Home project [4] aims at designing a smart home system based on audio technology focusing on three main goals: to provide assistance via natural *man-machine interaction* (voice and tactile command), to ease social e-inclusion, and to provide security reassurance by detecting situations of distress. The targeted smart environments are multi-room homes with one or more microphones per room set near the ceiling.

To show the results of the project, a *smart home* was set up. It is a thirty square meters suite flat including a bathroom, a kitchen, a bedroom and a study; in order to acquire audio signals, seven microphones were set in the ceiling. All the microphones were connected to a dedicated PC embedding an 8-channel input audio card.

The authors based the architecture on a multichannel audio card; in general dedicated hardware increases costs and reduces flexibility. Moreover, a wired approach is hard to implement since it requires to fix wires onto walls or into existing electrical pipes. BigEar project is based on low-cost hardware and wireless connections in order to preserve flexibiliy, ease of installation and reduce costs.

### 2.2. Wireless Sensor Networks for Voice Capture in Ubiquitous Home Environments

The authors in Palafox and García-Macias [5] present a voice capture application using a wireless sensor network (WSN). The WSN nodes (each node is a MicaZ *mote* with a high sensitivity microphone) are grouped in clusters with a special node (cluster-head) coordinating the cluster activities in order to avoid to capture and transmit the same voice command by two or more nodes; the authors consider the command duplication unacceptable. Each cluster-head collects audio data from their cluster nodes and relays it to a base station. Each node continuously senses audio signals at 2 kHz sampling frequency; if the sensed signal intensity exceeds a predetermined threshold, the node sends a notification to the cluster-head. The cluster-head selects a node to capture the command; the selected node enters into 8 kHz sampling frequency, captures three seconds of audio (with 8 bit resolution) and transfers the audio data to the cluster-head node which, in turn, relays it to the base station where a computer processes speech recognition tasks. The authors implement two capturing techniques: capture-and-send without coordination (consisting of human voice detection, three seconds of audio recording and transmission of the data packet to the cluster-head) and coordinate (consisting of human voice detection, node selection by the cluster-head, three seconds of audio recording and transmission of the data packet to the cluster-head). The main limit of this solution, beside having a quite high cost, is the sampling of three seconds instead of having a continuous voice detection and reconstruction. Our solution improves such limitation.

### 2.3. Exploiting WSN for Audio Surveillance Applications: The VoWSN Approach

The work in Alesii *et al.* [6] focuses on the analysis of fundamental issues about the transmission of the voice using a wireless sensor network (WSN). The paper is based on the MicaZ wireless sensor nodes. The prototype of the system has been developed through the use of the CrossBow MicaZ-TinyOS_v1.5. and a PC has been used for listening and the analysis of recorded data. The work focuses on audio surveillance systems (*i.e.* continuous or event-driven audio monitoring tailored to voice signals that have to be archived and/or postprocessed) and on multi-point sampling to make proper signals (or noises) cancellations. BigEar nodes are based on Wixel prototyping boards, that allow to reduce significantly costs with respect of the solution implemented from the authors, that is based on MicaZ *motes* [7] [8].

### 2.4. Differentiating Emergency Voice Traffic in Indoor Wireless Activity Monitoring Network

Demir *et al.* [9] propose an indoor wireless activity monitoring network (WAM-N) to transmit data in real time to a monitoring application. The architecture is based on a personal device where data from an accelerometer are transmitted to the sink for the detection of current physical activities (e.g. lying and sitting); an

acoustic sensor was located close to the bed of the older person, to transmit short voice commands (e.g. need help, open the door etc.) for emergency attendance. The paper aims at experimenting a network which treats the voice data as emergency traffic and tries to achieve a certain Quality of Service. The system is only simulated considering the voice and activity data into two different Quality of Service classes: class 1 for voice and class 2 for activity data. The voice data segment (55 bytes) is periodically sent in every 3.57 ms and each voice command is represented by 800 data segment while 55 byte activity data segment is periodically sent in every 2.7 s; they assume each captured voice command is digitized with a sampling rate of $f_s = 8$ KHz and bit depth of 8-bit. The authors consider only a single voice source for detecting the user command; compared to this approach we use a set of microphones, which are contemporaneously active. In this way, BigEar solution implements a wireless unobtrusive network that takes advantage of the simultaneous multi-sensor acquisition to reconstruct the speech signal regardless of the position of the source with respect to the sensors.

### 2.5. The Research and Design on Time Division Duplex (TDD) Voice WSN

Rong-lin *et al.* [10] present an architecture (including hardware architecture of voice node, routing node and gateway node) for Voice Wireless Sensor Networks (VoWSN). The main goal of this paper is the quality of the transmitted voice; such an objective justifies both costs and the energy consumption of the presented hardware solution. The voice node includes a voice circuit (with signal amplifier, filtering, acquisition, quantization, encoding and decoding, A/D and D/A conversion), a digital signal processing circuit (with real-time voice digital signal processing including ADPCM encoding to reduce data rate), and a ZigBee module; the routing node includes only the ZigBee wireless communication module, while the gateway nodes includes the ZigBee communication module, the CDMA module, and an ARM processing circuit. The authors implement the time division duplex (TDD) method to achieve duplex voice communications; in particular, they use the same frequency but different time slots for data sending and receiving.

### 2.6. Blind Alignment of Asynchronously Recorded Signals for Distributed Microphone Array

In this work, Ono *et al.* [11] present an architecture of independent recording devices that is used as a distributed microphone array. The main goal is to introduce a novel method for the alignment of recorded signals to estimate the localization of microphones and sources. The authors implemented only a *simulative experiment* with 9 microphones and 8 sources randomly positioned. As source signals, real-recorded hand claps were used and each source was not overlapped each other. The sampling frequency used by the authors was 44,100 Hz and the signal length was 5.0 s. High sampling frequency force the use of devices with high computing capabilities that reflects on costs of the overall architecture.

Moreover, it requires high bandwidth in order to transmit data between nodes. BigEar solution focuses on vocal signals and minimizes bandwidth requirements. As it will be discussed in following sections, our approach ensures a proper alignment of audio streams generated by different sensors.

### 2.7. BigEar Approach

The approach proposed in this work tries to improve the current state of the art by providing a faster and flexible access to the transmission channel that allows a more widespread acquisition, based on a low cost solution. Among non functional requirements we have considered that house adaptation should be avoided to ensure high degree of modularity and configurability. Moreover, closed systems and dedicated hardware have been considered as second-best choices not only for their licensing costs, but mainly to keep high levels of flexibility.

## 3. BigEar Architecture

Figure 1 illustrates the architecture of the BigEar system. It is composed of a network of *audio sensors* that distributively capture audio in a room. The speech is sent to a *main receiver* (BigEar Receiver), acting as an interface that converts speech packets received via radio channel into serial data to be sent to the *Base Station*. The Base Station contains the application logic to handle speech packets. Since the audio sensors perform a *space-time sampling* of the audio inside a room, the application logic tries to reconstruct a good-quality speech stream starting from packets that arrive to the base station with different timestamps and with different physical characteristics. Indeed, each sensor samples the audio signal that reaches the microphone after undergoing variations due to the physical model of the environment: different delays and amplitudes that depend on the position of the person with respect to the sensors, and reflections diffusions due to the geometry of the room and materials of the walls and furniture.

Granularity of space-time sampling is influenced by:

- Number of audio sensors w.r.t the dimensions of the room: the bigger is the

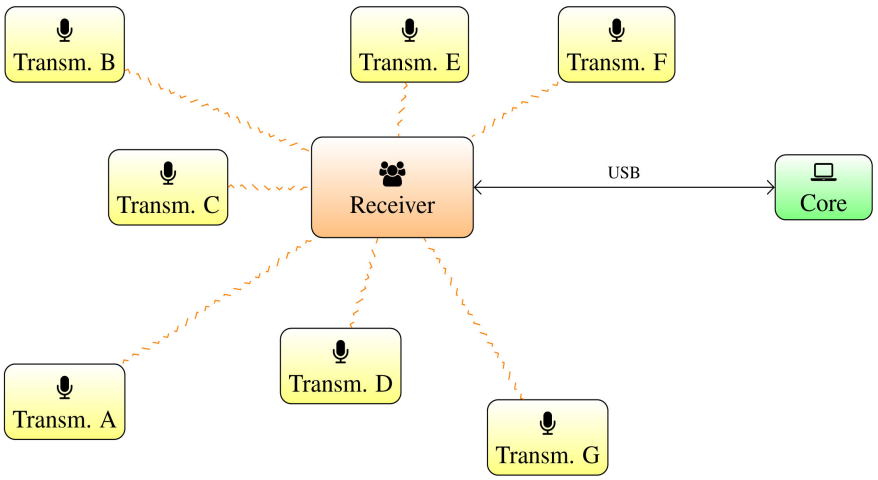


Figure 1. Overview of the BigEar architecture.

number of sensors spread in the room, the finer is the granularity of space sampling.

• Audio sensor internal characteristics and constraints: each sensor needs time in order to sample data (depending on ADC type), store them into buffers and send them to the main receiver.

• Network communication protocol characteristics and constraints: the number of packets sent to the main receiver is affected by the number of collisions that may happen on the channel and also by the protocols themselves (handshaking, request-response timings, timeslot allocations).

### 3.1. BigEar Audio Sensors

The leaf nodes of the architecture are represented by the audio sensors, that are built using Wixel Programmable USB Wireless Modules, general-purpose boards featuring a 2.4 GHz radio and USB port. The Wixel is designed around the CC2511F32 System-on-Chip (SoC) from Texas Instruments, which has an integrated radio transceiver, 32 KB of flash memory, 4 KB of RAM, and a full-speed USB interface.

Wixel's ADC is connected to a simple signal acquisition and conditioning stage that captures audio signals. ADC resolution represents an important design choice: the higher is the resolution, the lower is the quantization error. On the one hand, low resolutions allow high sampling frequency and so a higher temporal resolution with the drawback of a lower number of quantization levels; on the other hand, high resolutions reduce quantization error granularity. This is an important key factor in signal post-processing.

### 3.2. BigEar Receiver

The only task of the BigEar Receiver is to act as a Radio-to-USB interface (and vice versa) between BigEar Audio Sensors and the Base Station. It mainly receives radio packets from the sensors, transforms them into hexadecimal nibbles and sends them to the Base Station via the USB port. When the Base Station needs to send commands to the sensors (or to reply to protocol messages), BigEar Receiver receives hexadecimal nibbles through the USB port, converts them into bytes and sends them using the built-in radio module. Like BigEar Audio Sensors, also the BigEar Receiver is based on a Wixel Programmable Module.

CC2511F32 SoC has been programmed in order to handle up to 256 different radio channels [12]. All the sensors share the same channel used by the BigEar Receiver; if the network architecture requires channels separation (e.g., to reduce the number of collisions), a second BigEar Receiver connected to another USB port of the Base Station is needed.

### 3.3. Base Station

The Base Station is the device that collects data from the sensors and arranges packets in order to produce a clear and intelligible speech signal. In order to receive packets from audio sensors, it needs to be connected via USB port to the BigEar Receiver, which acts as a bidirectional Radio-to-USB dongle between the

Base Station and the wireless audio sensors.

The Base Station receives radio packets containing each one a set of bufferized audio samples tagged with a timestamp and the sensor ID; for each sensor, audio samples are arranged according to their timestamp. In this way, for each sensor a coherent but incomplete stream is obtained: indeed, audio samples are in the right time position with respect to the sensor timestamp, but there is no guarantee that the transmission time is less than, or at most equal to, the sampling time.

Once the samples have been sorted by their timestamps, the application performs a time delaying-or-advance of the audio streams coming from the sensors in order to remove the delays caused by the different distances between the mouth of the user and the sensors. Therefore, in-phase audio contributions are obtained; they can be summed each other in order to produce a seamless stream.

During the alignment process the different energy contribution of the sensors are considered: the closer is the sensor to the user, the bigger will be the signal amplitude and vice versa.

Figure 2 summarizes the Speech Reconstruction Logic carried out by the Base Station: in the left plot in Figure 2(a), audio packets can be seen as received from the Base Station. Then, the Base Station exploits timestamp information carried by each audio packet to arrange audio samples onto the sensor's timeline (right plot in Figure 2(a)). Figure 2(b) illustrates the operating principles of cross-correlation analysis that allows the Base Station to obtain the in-phase contributions that will be superposed to generate a unique, coherent and intelligible speech signal.

### 3.4. Network Protocols

Network protocols have a big impact on the efficiency of the whole system: collisions, granularity of the network of sensors and presence of protocol messages can affect the number of audio packets transmitted and received successfully.

In general, when the granularity of the network increases, also the likelihood of collisions grows. At the same time, the number of service messages to implement synchronization mechanisms have to be increased in order to reduce the number of collisions. This can be done at the expense of the channel availability and software complexity.

The simplest protocol that can be adopted in this scenario is ALOHA [13]. Our application has been tested using the pure ALOHA protocol (without acknowledge) in order to exploit and examine system capabilities with the simplest communication protocol.

## 4. Speech Acquisition Model and Implementation

Figure 3 shows a functional view of the BigEar application within its acquisition environment. It is composed of four interconnected modules: the Audio Model block performs the acoustic simulation of the acquisition environment, the Sensor Network Model (which is in turn composed of two inner blocks) simulates the behavior of the transmitters-receiver network; finally, the Speech Reconstruction

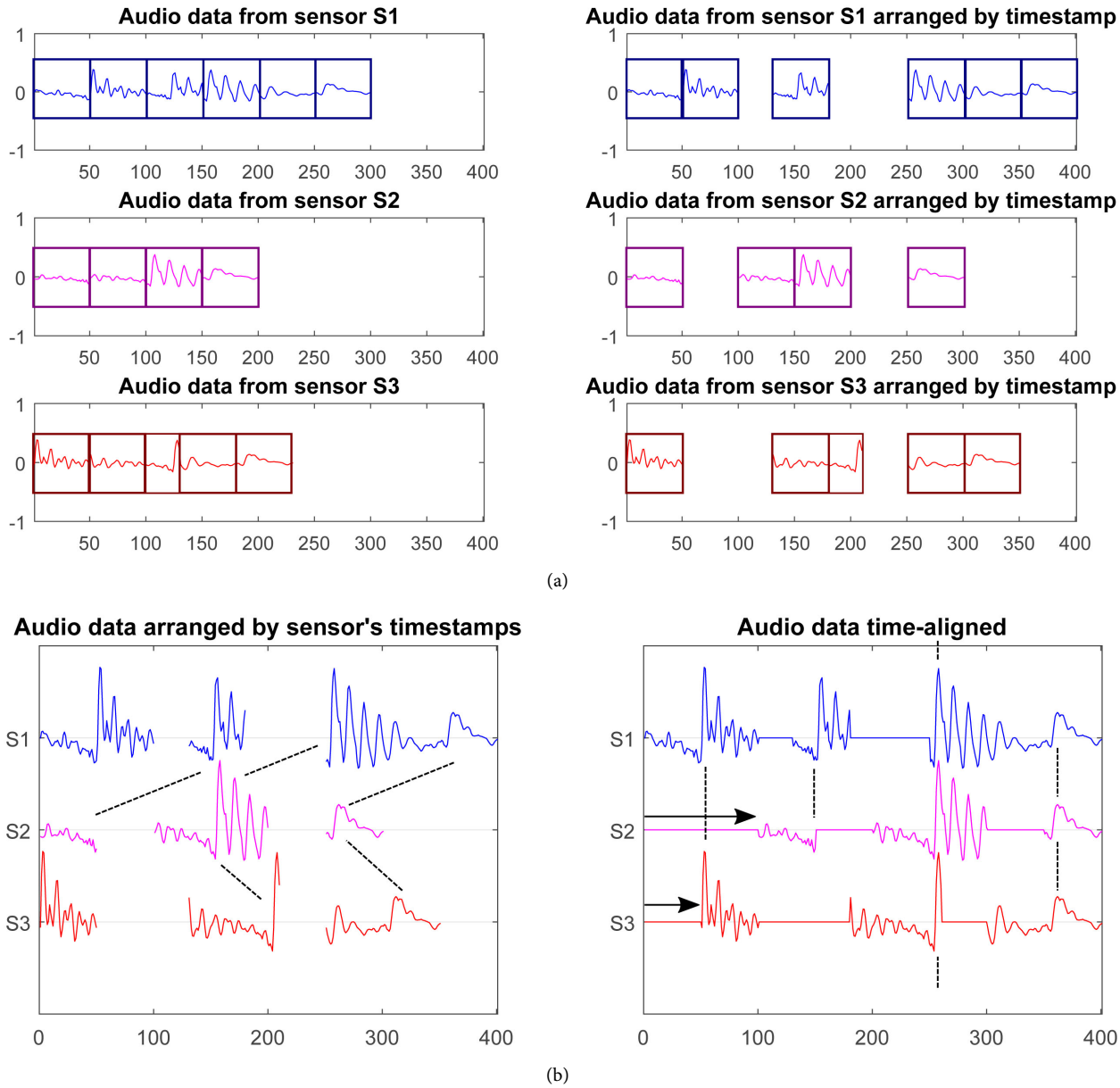


**Figure 2.** Base station reconstruction logic. (a) Audio packets arranged w.r.t. their timestamp; (b) Audio packets are aligned exploiting cross-correlation between signals.

block performs the reconstruction of the speech signal.

While the first three blocks concerning the behavior of the sensor-receiver network have been simulated, the Speech Reconstruction block has been implemented and its prototype will be described and discussed in Section 5.

### 4.1. Audio Model

The Audio Model block models the acoustic environment and sets: the room dimensions (and optionally other parameters such as reflection and diffraction coefficients of walls), the location of the sensors and of the audio source. Once an input audio file is provided, the block produces an audio stream for each sensor. Each stream differs for its amplitude, delay and diffusion, depending on

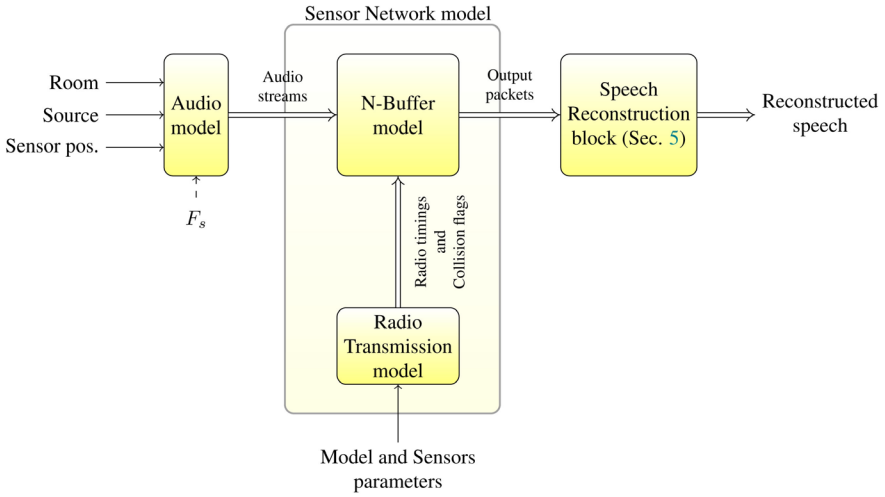


**Figure 3.** Architecture model.

the acoustics characteristics of the room and on the position of the sensor with respect to the source.

In order to create an audio model of the typical use case, we made some assumptions about the typical environment where the system may work. The room is modeled as a parallelepiped represented by a three-dimensional space bounded by six surfaces (four walls, ceiling and floor). Each room surface has its own absorption and scattering (diffusion) coefficients. Sound scattering due to furniture and other objects in the room can be approximated by higher levels of overall room diffuseness. Audio modeling is performed by means of MCRoomSim multichannel acoustics MATLAB simulator [14].

## 4.2. Sensor Network Model

This module recreates a realistic simulation of the behavior of the network where each sensor samples audio signals, buffers them into packets of a specific size and sends them according to a network communication logic. The module is internally split in two parts: the *Radio Transmission model*, that implements the communication protocol including possible interactions between a receiver and the transmitters or between transmitters, and the *N-buffer model*, that carries out the internal buffer mechanism of each transmitter.

The Sensor Network module receives in input the audio streams generated by the audio simulator described in Section 4.1, and provides as output the packets of audio sampled and transmitted by each sensor.

**Radio Transmission model** This block implements the pure ALOHA protocol described in Section 3.4, where each transmitter sends data whenever there is a packet to send and then waits for a random delay before sending another packet. Since audio data are time-dependent, for our purposes it is worthless to re-transmit audio packets, so the transmitter will not wait for any acknowledgment from the receiver. The random delay between transmissions is obtained by the internal random number generator of each transmitter and it is chosen be-

tween 0 and a maximum value $T_{\max \text{Delay}}$.

The model also checks for collisions. Given the time instant $t_{(i,j)}$ in which the $j^{\text{th}}$ transmitter starts to transmit the $i^{\text{th}}$ packet, and called $t_{\text{busy}}$ the duration of the transmission, all the transmissions started in the interval $\left[t_{(i,j)} - t_{\text{busy}}, t_{(i,j)} + t_{\text{busy}}\right]$, where $t_{\text{busy}}$ are marked as colliding.

***N*-Buffer model** This block implements the buffering system internal to each transmitter. In a real world, data are continuously sampled and buffered by each transmitter in order to be ready to send them when needed; during the simulation, instead, the time instants in which transmission occurs are known, but we need to model the buffering structures to know which data are ready for being transmitted. This block produces in output the audio samples packed as if they were coming from real transmitters. The structure of each packet is described in Figure 4: each packet contains the ID of the transmitter, the timestamp of the first sample of the packet and a number of samples that correspond to the frame size of each buffer. Only the timestamp of the first sample is sent, since the timestamps of other samples can be inferred by adding $\tau_i = i/F_s$, where $i$ is the (0-based) index of $i^{\text{th}}$ sample in the packet and $F_s$ is the sampling frequency.

Multiple buffering allows the sensor to work simultaneously on read and write sides: a periodical interrupt routine acquires the signal and stores samples into the *write frame*, while the main loop can read from the *read frame* for the transmission.

### 4.3. Real-World Implementation

The three modules described so far compose the virtual model that has been used in order to define the working parameters before the realization of the prototypes. In the real world, the system is composed of a set of audio sensors that perform a *space-time* sampling of a room. Before sampling, the audio signal converted by each microphone capsule has to be amplified and biased in order to match to ADC characteristics. Each audio sensor samples the signal and packs data into frames in order to send them to the receiver. The multi-buffer internal structure of each transmitter allows an efficient application logic where the sampling stage is managed by means of a timer-handled interrupt routine, and the network logic is handled by the main loop of the applications. Network structure can be layered onto several radio channels in order to reduce the number of collisions. A separate BigEar Receiver is needed for each radio channel.

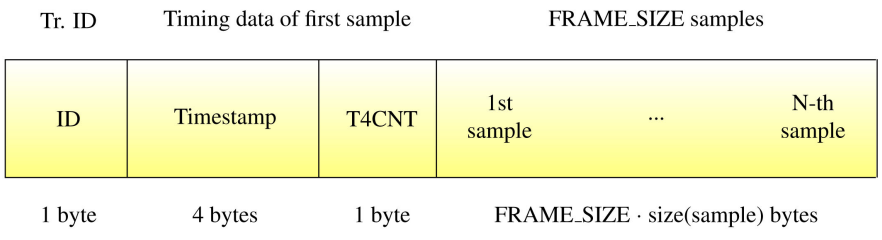


**Figure 4.** BigEar data field structure.

Once the packets arrive to the BigEar receiver, they are converted into hexadecimal nibbles and serially sent to the Base Station by means of the USB port. The Base Station, in its experimental form, is composed of a pipelined application that listens to each BigEar Receiver connected to the USB ports of the machine, receives audio packets and stores them into an ordered sequence of files that are processed by means of a MATLAB script that reconstructs the audio data using the reconstruction principles described in Section 5.

## 5. Speech Reconstruction

This section illustrates how audio packets are processed in order to produce the best speech signal in terms of intelligibility.

Starting point of the Speech Reconstruction is constituted by the audio packets received by the Base Station from each audio sensor. Due to sound propagation laws, the closer is the sensor to the source, the higher will be the power of captured audio signal; so, in order to preserve energy of the reconstructed signal, audio packets are unbiased and normalized. Then, audio samples are arranged using timestamp information contained in each packet (as already illustrated in Figure 4).

Delays introduced by the different distances between the source and the audio sensors (the closer is the sensor to the source, the lower will be the time of arrival of pressure wave to the microphone) are compensated in the streams alignment stage using cross-correlation analysis.

When the signals have been properly aligned they can be superposed with several summing or replacing methods in order to preserve the signal energy and not to introduce energy variations due to the different number of contributions that are summed at the same time.

The flowchart in Figure 5 illustrates the operation performed by the Speech Reconstruction module; they are described in the next subsections.

### 5.1. Energy Compensation

Audio packets have to be processed in terms of energy compensation to prevent distortions. In particular, the following steps are performed:

- Bias removal, in order to eliminate possible incorrect polarization of the input stage.
- Normalization of input signals, to remove the amplitude attenuation due to the different distances between the speech source and the sensors.

#### 5.1.1. Bias Removal

Incorrect polarization of the input signal can affect the result of the reconstruction block, that is based on the summation of contributions that vary randomly in time. Audio signals coming from different sensors are affected by different polarization. The summation of different DC components corresponds to the superposition to the audio signal of a square wave whose frequency and amplitude are randomly changing, introducing in this way harmonic distortion to the

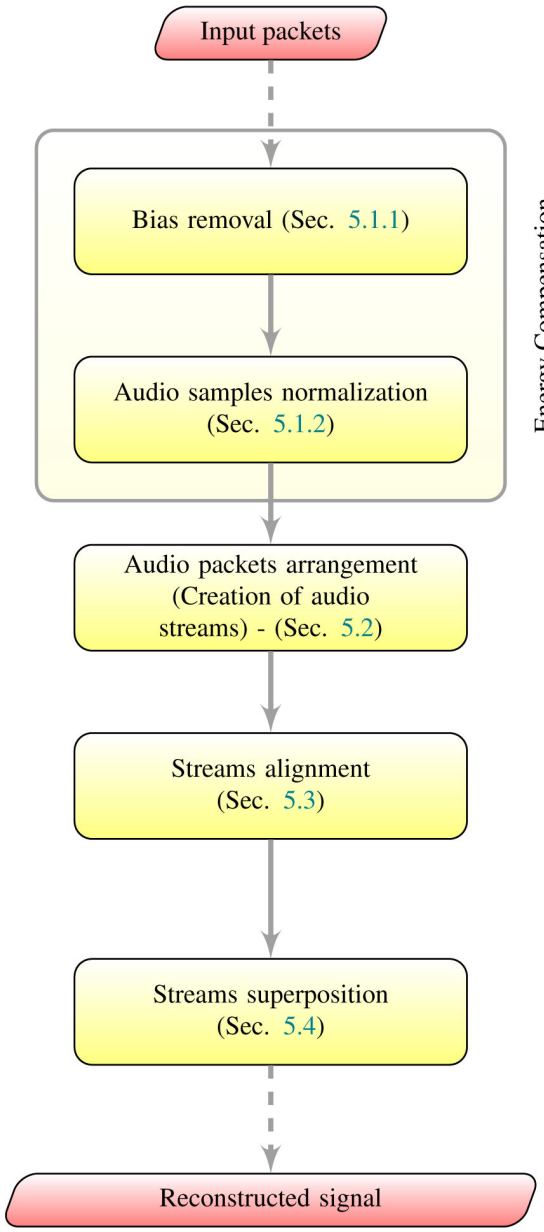


**Figure 5.** Flowchart of the signal reconstruction block.

speech signal, as illustrated in **Figure 6**.

$$a_{(i,\bar{j})} = a_{(i,\bar{j})} - \mathbb{E}\left[a_{(i,\bar{j})}\right] = a_{(i,\bar{j})} - \sum_{k=1}^{N} \frac{a_{(i,\bar{j})}}{N}, \tag{1}$$

where $N$ is the number of sensors.

### 5.1.2. Normalization

Normalization removes energy dependence on the distance between the speech source and the sensor. In this way, neglecting differences in frequency response of microphones and small variations in spectral content due to room acoustics, contributions of different sensors can be summed without compensations coefficients:

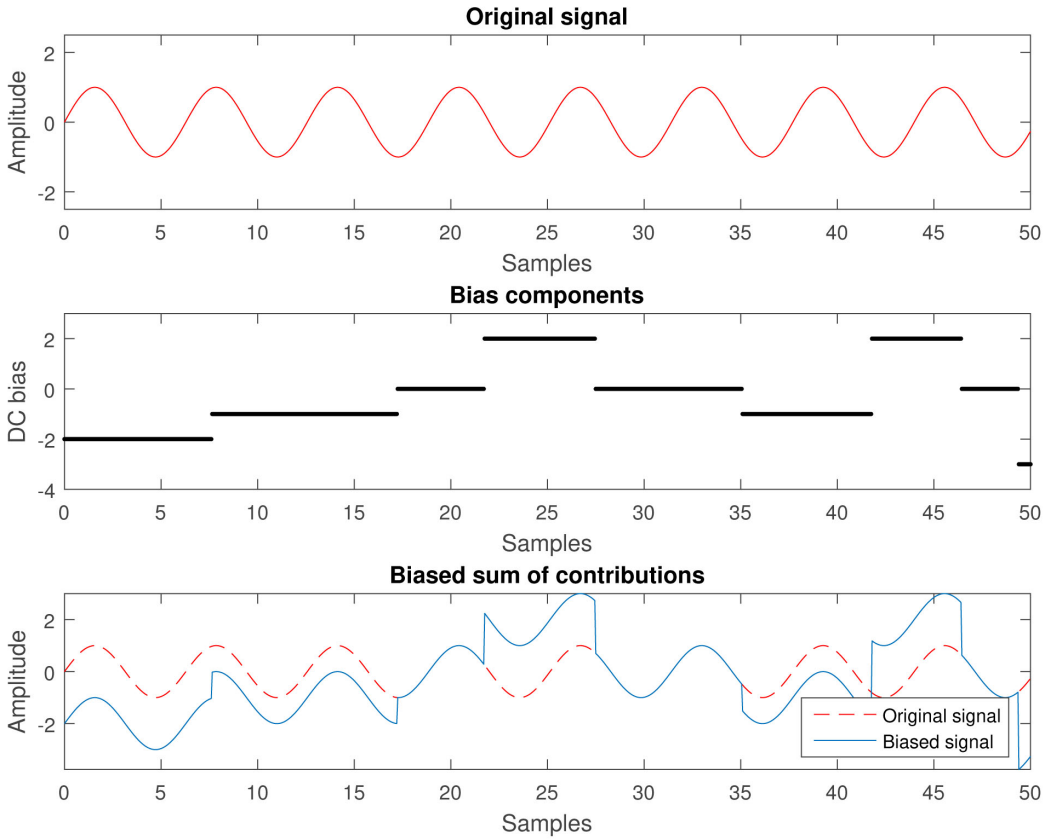


**Figure 6.** The image shows the effect of the acquisition of the same signal by means of different sensors, each one characterized by a different DC bias.

$$a_{(i,j)} = \frac{a_{(i,j)}}{\max\limits_{1 \le k \le N} a_{(k,j)}} \quad \forall j \in (1, N). \tag{2}$$

## 5.2. Audio Packets Arrangement

The second step of Speech Reconstruction is the arrangement of audio packets. Audio samples are extracted from each packet and get ready for processing using two matrices: $\mathbb{A}$ and $\mathbb{P}$, where each element $a_{(i,j)} \in \mathbb{A}$ represents the $i^{\text{th}}$ sample transmitted by the $j^{\text{th}}$ sensor and the corresponding element $p_{(i,j)} \in \mathbb{P}$ represents the position of the audio sample in the stream expressed in discrete-time units:

$$p_{(i,j)} = \left\lfloor F_s \cdot \left( t_{(i,j)} - t_{\min(j)} \right) \right\rfloor \quad \text{where} \quad t_{\min(j)} = \min_{j=\tilde{j}} \left( t_{(i,j)} \right). \tag{3}$$

Using position information, audio samples are correctly spaced on the sensor's timeline. For each sensor $j : 1 < j \le N$, a vector $y_j$ of samples is created:

$$y_j \left( p_{(i,j)} \right) = a_{(i,j)}. \tag{4}$$

The elements in $y_j$ where no audio samples are present are 0-filled.

## 5.3. Streams Alignment

The streams generated from audio data coming from sensors are aligned to

reduce the delay due to the distance of the speech source with respect to the position of the sensors. The alignment is obtained by using the cross-correlation function [15]. In order to apply it efficiently, the audio streams are processed according to their informative contribution: they are sorted by their normalized power in descending order to allow the cross-correlation algorithm to work in the best condition.

Cross-correlation is a measure of similarity of two series as a function of the lag of the one relative to the other. For two generic discrete functions $f[m]$ and $g[m]$, cross-correlation measure is defined as:

$$R_{fg}[n] = (f \star g)[n] \triangleq \sum_{m=-\infty}^{+\infty} f^*[m]g[m+n], \tag{5}$$

where $f^*$ denotes the complex conjugate of $f$. The equation essentially slides the $g$ function along the x-axis, and calculates the integral of their product at each position. When the functions match, the value of $(f \star g)$ is maximized. Thus, applying cross-correlation to the streams $y_i$ and $y_j$ generated by two different sensors $i$ and $j$ means to find the delay $n$ (expressed in number of samples) that should be applied to $y_j$ to obtain the best in-phase superposition with $y_i$.

### Envelopes Cross-Correlation

A drawback of Cross-correlation function is the inability in discriminating between the *true* signal and noise or holes.

Cross-correlation function operates on signals that, for their origin, are noisy and holey. If holes and noise are negligible, cross-correlation gives expected results; if sequence of zeros (*holes*) are much bigger than the signal itself, or if the signal is subject to particular types of noises such as impulse trains, the Cross-correlation function would produce wrong results. To overcome this problem, instead of applying Cross-correlation function directly on noisy or holey signals, it has been applied to the *positive envelopes* of the signals themselves. A positive envelope is a particular representation of a signal that evidences the *shape* of the signal. **Figure 7** illustrates the result of the alignment step of the *envelopes*. On the image, for the sake of readability, envelopes of the streams coming from different sensor have been shifted along y-axis. It can be noted that peaks and valley of the signals are globally aligned. This alignment technique offers higher robustness with highly noisy or highly depleted streams, although the effort for a better alignment could be frustrated from the lower intelligibility of the speech signal.

## 5.4. Streams Superposition

Once audio streams obtained by sensor acquisition have been made uniform by means of unbiasing and normalizations, and they have been delayed to make them coherent, they need to be superposed in order to reconstruct the recorded speech signal. Two methods have been implemented: Weighted Sum of Contribution and Holes Replacement.

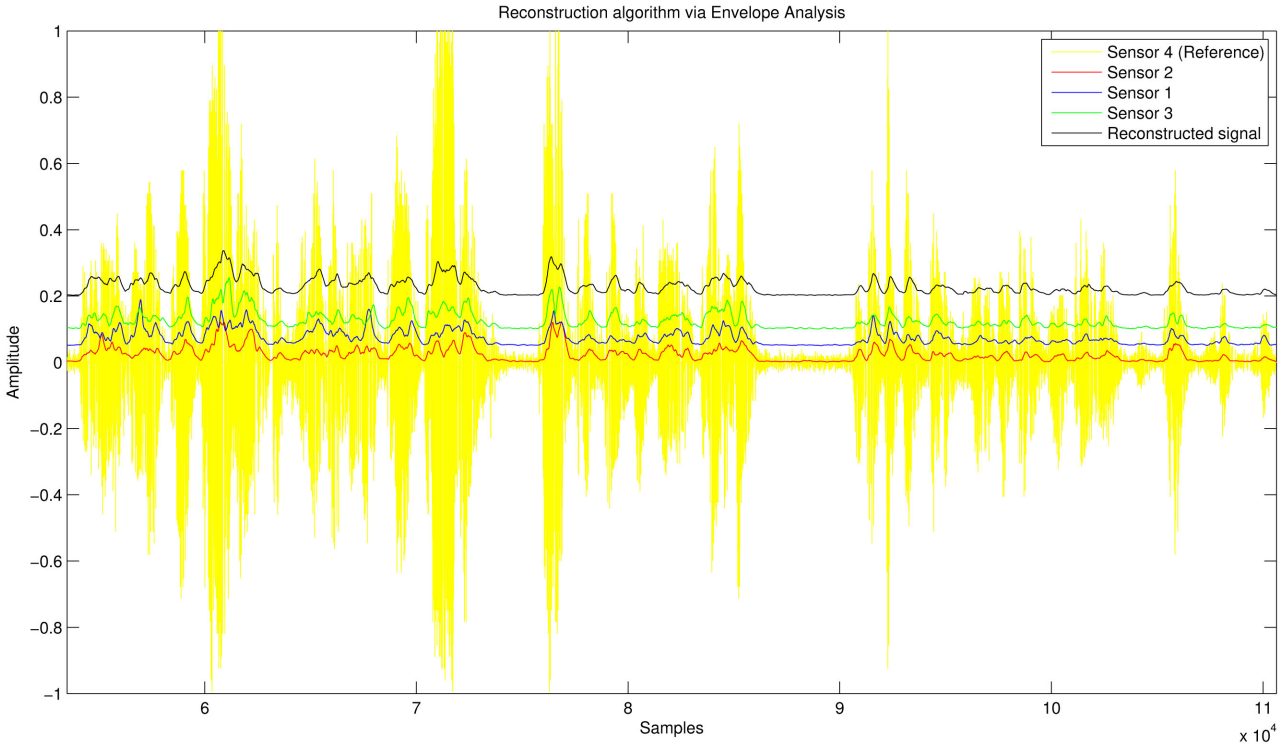


**Figure 7.** Cross-correlation analysis and alignment on signal's envelopes.

### 5.4.1. Weighted Sum of Contributions

Contribution coming from different sensors are summed and scaled to prevent amplitude artifacts. Given $y_{(i,j)}$ the $i^{\text{th}}$ sample of the audio stream coming from $j^{\text{th}}$ sensor and $w(i)$ the number of sensors that contribute to the $i^{\text{th}}$ sample, the $i^{\text{th}}$ sample of the resulting stream $y_{\text{sum}}$ is given by:

$$y_{\text{sum}}(i) = \frac{\sum_{j=1}^{N}(y_{i,j})}{w(i)}. \tag{6}$$

Weighted Sum is needed for energy preservation and for avoiding harmonic distortions due to the summation of contributions. **Figure 8** illustrates an example of distortion caused by the sum of multiple contributions without weighting.

### 5.4.2. Holes Replacement

Weighted Sum of Contribution presents some drawbacks: it does not take into account the big differences in the spectrum of signals and in the environment contributions between sensors located in different places. Each BigEar Audio Sensor is subject to an environment contribution that depends on:

- The distance between the sensors and the speech source;
- The position of the sensors in the environment.

Contributions can be very different in terms of signal spectrum and of reverberation. In general, the closer the sensors, the lower will be the overall effect of the environment-inducted artifacts since spectrum of the signals will be similarly colored and reverberation tails will be alike.

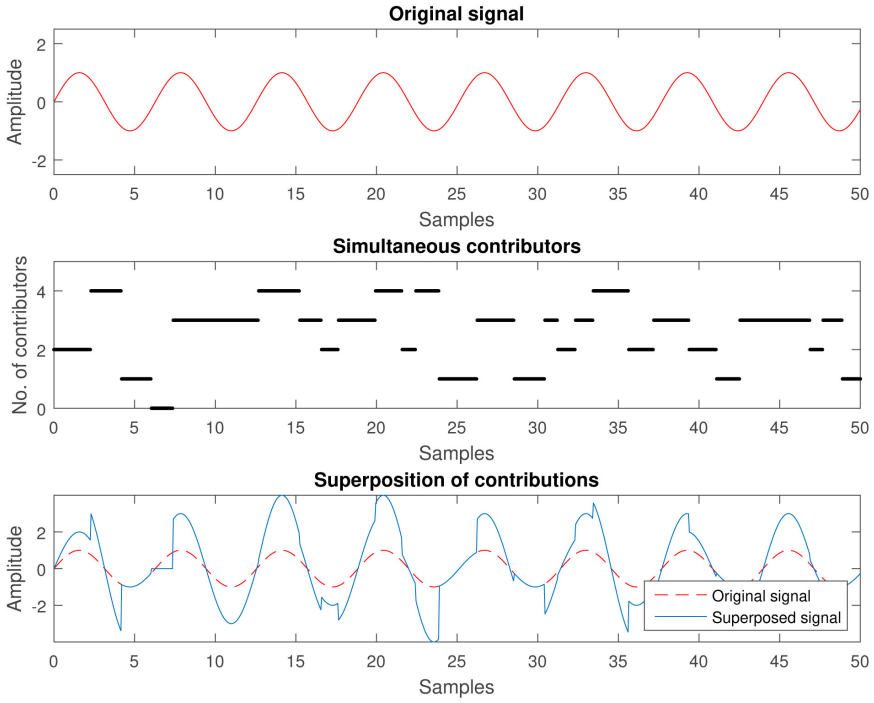


**Figure 8.** Harmonic distortion due to unweighted sum of contributions.

For this reason, an alternative superposition policy has been tested: instead of performing a weighted sum of each contribution, only the holes in the most powerful audio stream are filled with contributions coming from other sensors. This method reduces the number of summation artifacts, provided that the reference signal (the one on which the holes will be replaced with samples coming from other sensors) has the higher number of valid samples, otherwise there is the risk that *replacing artifacts* will become prominent with respect to *summing artifacts*. A comparison metric between Weighted Sum and Holes Replacement will be discussed in Section 7.2.

## 6. BigEar Simulation and Model Validation

Once the system has been implemented and the prototype realized, some metrics have been defined to compare the data captured by means of the BigEar prototype and the data obtained by means of the BigEar simulated model described in Section 4.

This Section focuses on the quality of reconstructed signal analyzing amount of overlapping data between the stream generated by each sensor, while Section 7 illustrates system capabilities in terms of speed of processing and outlines qualitative aspects of the reconstructed speech.

### 6.1. Metrics Definition

The metrics defined in this sections provide quantitative measures concerning the reconstructed speech signal. As already mentioned in Section 5, the success of speech recognition is influenced by the number and the size of holes in the

reconstructed signal. Moreover, the BigEar Reconstruction algorithm convergence is influenced by the amount of information that can be overlapped for the Cross-correlation alignment. The following metrics are therefore defined:

$$Fill\_ratio = \frac{\text{No} \cdot \text{of samples}}{N} \quad \text{where } N = \text{Length of the stream}\left(\text{in samples}\right).$$

Referring to the reconstructed signal, it represents the amount of samples with respect to the total length of the stream. The more the value is close to 1, the more the reconstructed signal is complete.

$$NoH = \text{Normalized number of } 0-ed \text{ sequences}$$

$$SoH = \text{Average size of } 0-ed \text{ sequences.}$$

Since reconstructed signal is given by the superposition of audio packets sampled by different sensors at random time instants, it can be affected by sequences of *empty samples*. In conjunction with $NoH$, $SoH$ characterizes the distribution of empty samples (holes) into the reconstructed signal. In case of constant $Fill\_ratio$, $SoH$ and $SoH$ allow to compare whether empty samples are gathered into few big blocks or are diffused into many small blocks.

$$Sf = \text{Average number of contributor per sample.}$$

$Sf$ gives a measure of the contribution of each single transmitter to the construction of the final speech signal. $Sf \in \left(0, N_{TX}\right]$ where $N_{TX}$ is the number of transmitters. The higher $Sf$, the higher the overlapping of the streams obtained by the different transmitters.

## 6.2. Simulation Setup

Simulations have been performed using the BigEar Simulator described in Section 4 in a MATLAB 2015b environment [16] using McRoomSim v. 2.14 [14] and varying some parameters in order to study system behavior under different configurations. The parameters that have been changed are: the number of sensors, their positions in the room, the radio channel configuration (how many transmitters communicating on the same radio channel), and the maximum delay between adjacent transmission of the same transmitter.

From these simulations, Statistic data and Metrics have been calculated according to Section 6.1. These data will be compared with real data obtained from On-field setup (Subsection 6.3) and discussed in Section 6.4.

## 6.3. On-Field Setup

**Near-field Tests** During Near-field tests, the consistence between the simulated model and the real world has been probed. In this setup, BigEar Audio Capture boards were placed side by side on a plane surface, and the speaker has been asked to talk at a distance of about 0.6 m far from the microphones. Then data have been captured using different configurations:

- Number of transmitters and channel configuration. Character sequences indicate the number of channel and how many transmitters are transmitting on

the same channel (e.g. AAB means two transmitters on radio channel A and one transmitter on radio channel B):

-One transmitter: A.

-Two transmitters: AA-AB.

-Three transmitters: AAA-AAB.

-Four transmitters: AAAA-AAAB-AABB.

• Maximum delay between adjacent transmissions from the same transmitter ($T_{\max\,\mathrm{Delay}}$ parameter): 1-3-7-15-31-63 ms.

**Far-field Tests** During Far-field tests, the focus has shifted on the Reconstruction Algorithm. This is a test stage close to real situation since BigEar Audio Capture boards have been fixed to poles 1.60 m high from ground level and have been placed in a medium-size room. The talker has been asked to speak from an asymmetric position to examine the signal power differences between the different streams.

Figure 9 shows the obtained plots; black asterisks mark real values obtained from the prototypes, while lines indicate the simulated ones. By varying the $T_{\max\,\mathrm{Delay}}$ parameter and the number of transmitters, the obtained curves are asymptotic. Differences are notable when $T_{\max\,\mathrm{Delay}} \in \{1,3,7\}$, *i.e.*, when the average distance between adjacent transmissions of the same transmitter are comparable with the duration of a frame of samples

$$(\mathit{samples\_per\_frame} \cdot \mathit{sampling\_period} = 20 \times \frac{1}{6040\ \mathrm{Hz}} = 3.31\,\mathrm{ms}).$$

This difference is due to the modular structure of the BigEar Simulator: the N-buffer Internal model (Section 4.2) does not communicate to its predecessor - the Radio Transmission model (Section 4.2)-any information about the buffer status. In the real world, if the buffer is empty, no transmission happens; instead, the Radio Model makes no considerations on the buffer status, with the result that virtual transmitters that have no data to transmit also contribute to the saturation of the audio channel and then to the valid packet loss.

Looking at $\mathit{Fill\_ratio}$, it can be observed that in most cases the real $\mathit{Fill\_ratio}$ is slightly higher than the simulated $\mathit{Fill\_ratio}$. The motivation is due to the fact that the model adopts $T_{\mathrm{busy}} = 1$ ms as duration of the transmission, while for the prototype the measured duration of a transmission is 0.937 ms.

In general, the increase of the number of transmitters leads to an increment of the overlap between sampled data, while the increase of the used radio channel leads to the reduction of the collisions between packets traveling on the same channel. By comparing Figure 9(a) with Figure 9(b) it can be observed that doubling the number of transmitters and working on 2 channels instead of 1, a big increment in $\mathit{Fill\_ratio}$ and in $Sf$ (support factor) are obtained, thus improving the quality of signal (in term of size of holes) and the support factor, *i.e.* the quantity of overlapped samples between the streams.

## 7. Experimental Results and Evaluation

In this section experimental results will be evaluated to test the reactivity of the

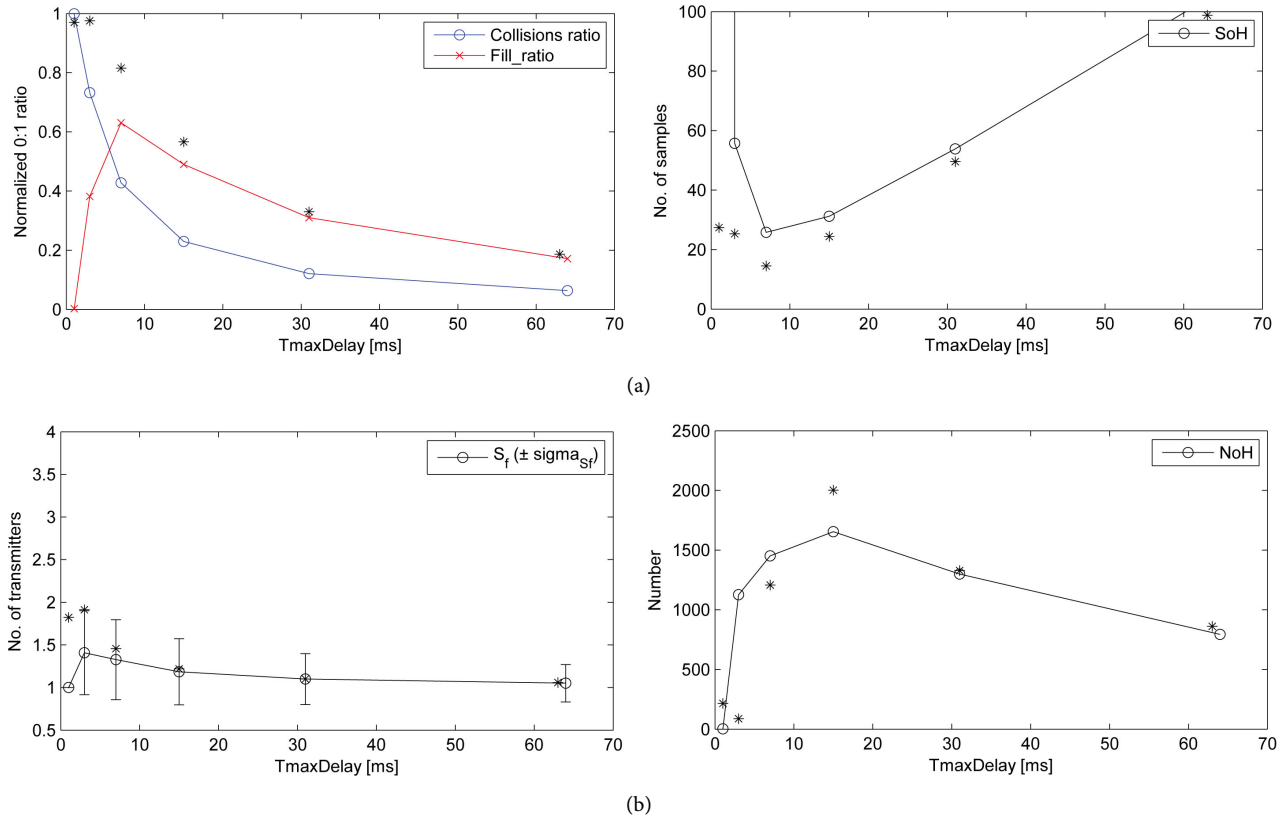


**Figure 9.** Metrics of the reconstructed signal plotted as a function of $T_{\max\ \mathrm{Delay}}$ parameter. (a) Test case: 2 transmitters on the same channel (AA); (b) Test case: 4 transmitters on the two channels (AABB).

system and the accuracy of the speech recognition process.

## 7.1. Speed of Processing Evaluation

System reactivity can be considered as the system's ability to interact with the user in real-time, *i.e.* to perform an action as soon as possible. Due to the modular architecture of the system, the reactivity can be analyzed from different points of view.

**Clock Tests** During the implementation of the hardware part of the BigEar prototype, the timer stability have been verified since one of the crucial points of the application is that every BigEar Audio Capture board samples the analog signal at the right sampling frequency $F_s$. Moreover, it is important to observe that the *N*-Buffer mechanism works perfectly in order to avoid corrupted data that could generate unattended behaviors in the reconstruction step. All the tests confirmed the clock stability.

**Speed of Speech Reconstruction Algorithm** The speed of the Speech Reconstruction Algorithm can be expressed as the ratio of the length of the considered audio segment $\Delta T_{\mathrm{rec}}$ over the duration of the elaboration process $\Delta T_{\mathrm{elab}}$. This metric, called Realtime Performance Ratio (RPR), is defined as:

$$\mathrm{RPR} = \frac{\Delta T_{\mathrm{rec}}}{\Delta T_{\mathrm{elab}}} = \frac{\left(\text{length of reconstructed signal}\right)\cdot\dfrac{1}{F_s}}{\Delta T_{\mathrm{elab}}}. \tag{7}$$

This measure depends on the number of transmitters, since with a higher number of transmitters, there is also a higher data flow. So the metric can be used as a global trade-off parameter: $\mathrm{RPR} > 1$ states that the whole system is able to buffer, send and process data faster than sampling. For each test case discussed in Section 10, RPR has been measured. For all the tests $\mathrm{RPR} \gg 1$, *i.e.* the processing speed of the BigEar Reconstruction Algorithm is faster than the sampling speed.

### 7.2. Reconstruction Quality Metric

During Far field tests, the speech signal was reconstructed using both Weighted Sum method (Section 5.4.1) and Holes Replacement method. Listening tests have denoted big differences in reconstructed speech signal depending on the superposition policy adopted. As explained in Section 5.4.2, the higher the distances between BigEar Audio Capture boards, the higher the differences in the audio signals due to different environment reflections and diffusions. These differences cause discontinuity artifacts in the reconstructed signal at the positions where different contributions are superposed in the attempt to fill the holes in the reconstructed signal (described in Section 5).

In order to examine how superposition methods could affect the presence of artifacts, Potential Artifact Ratio metric counts the number of positions where artifacts could be generated and normalizes it with respect to the length of the signal, obtaining thus a comparable metric.

$$A_{ws} = \sum_{k=1}^{N_{TX}} \frac{\mathrm{edges}_k}{N},$$

where $\mathrm{edges}_k = 2 \cdot NoH_k$ and $NoH_k = \mathrm{no} \cdot \mathrm{of\ holes\ in\ the\ stream\ producedby}$ $k^{\mathrm{th}}$ sensor.

$$A_{hr} = \sum_{k=1}^{N_{TX}} \frac{\left(\mathrm{edges}_k - \mathrm{edges}_k^{h<k}\right)}{N},$$

where $\mathrm{edges}_k^{h<k} = \mathrm{edges\ in\ the}\ k^{\mathrm{th}}$ stream covered by samples of previous streams.

Since number of potential artifacts is dependent on the chosen superposition policy, two different calculation methods are needed: $A_{ws}$ is the metric used for Weighted Sum reconstruction and $A_{hr}$ is the one used for Holes Replacement method.

Figure 10 shows that for each TmaxDelay set, Weighted Sum method (whose Artifact Ratio is denoted with $A_{ws}$) is more prone to artifacts creation than Holes Replacement method. Moreover, as expected, Potential Artifacts Ratio grows with the number of transmitters that compose the system, in particular when multiple transmitters operate on multiple channel: since there is high overlapping between audio packets, Weighted Sum has more data to superpose.

The approach of the Holes Replacement policy (Section 5.4.2) is different: it adopt as reference the more powerful signal, then it uses other streams for holes replacement. In this way, the Potential Artifacts Ratio metric gives better results, keeping low the number of points in which an artifact could be generated.

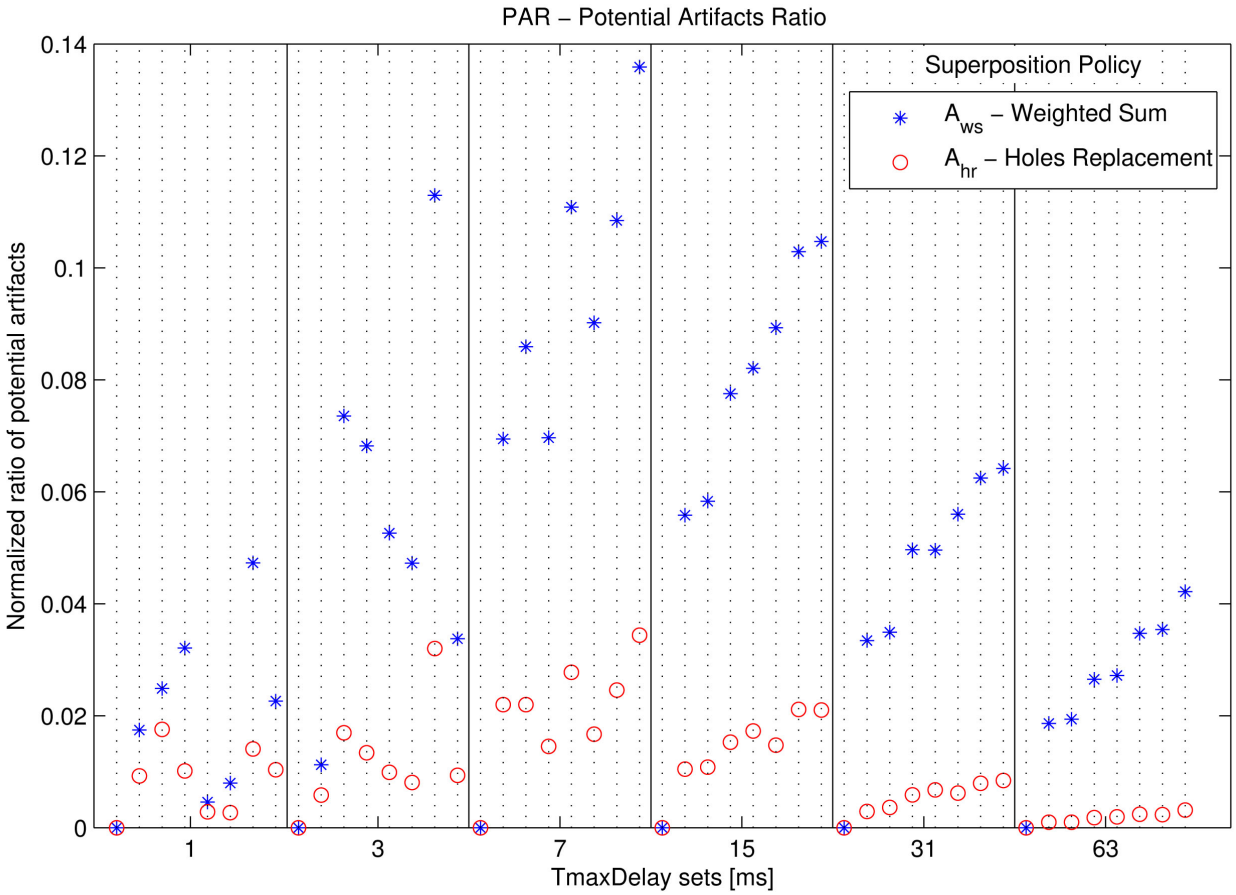


**Figure 10.** Potential Artifacts Ratio plotted for different test cases, divided by TmaxDelay sets.

## 8. Conclusions and Future Works

In this paper, we have presented an application based on a distributed wireless sensor network that performs a *space-time audio sampling* of an environment. It is based on the low cost technology, *i.e.* on Wixel Prototyping boards, whose cost is around 20 $ each one; cost for speech acquisition circuit is under 10 $ per board[2].

A virtual model of the architecture has been first implemented, including an Audio Model that performs the acoustic simulation of the acquisition environment and a Sensor Network Model simulating the behavior of the transmitters-receiver network: this BigEar Simulator can be used to perform an apriori analysis to identify the best parameters (such as number of sensors, position of sensors, number of channels, software-configurable parameters) for a specific use case, minimizing production and installation costs. A real-world system has also been implemented to examine its real behavior and capabilities. A speech reconstruction algorithm has been proposed to reconstruct the audio signal coming from different microphones; finally, since in case of speech recognition, the reconstructed stream may contain holes; an inexactness toleration mechanism has been included in the speech recognition process to improve recognition accuracy. The whole architecture is scalable and it can be easily reconfigured by adding or

removing sensors from the sensor network.

Results show that the BigEar Reconstructor algorithm can perform real-time speech reconstruction, and listening tests confirm the intelligibility of the reconstructed signal.

As future work, we plan to improve the BigEar Reconstruction Algorithm to properly feed GSR by filtering only vocal commands. Some experiments have shown that the differential information of power and delay of the signals acquired by the sensors can be used to make a coarse-grain localization of the source. Further studies will lead to a significant increase in localization accuracy to associate each keyword to a spatial information. In order to neutralize effects of superposition artifacts, filtering or far-field speech processing methods can be integrated into the BigEar Reconstructor algorithm; moreover, periodical training stages can be adopted for identifying physical and spectral characteristics of the ambient noise. Finally, the Network Interaction Model could be extended to other network protocols than pure ALOHA family in order to explore how Reconstructed Signal Metrics are influenced by different Network Interactions. In particular, different Network Protocols might help in reducing superposition artifacts; furthermore, Network Protocol could include synchronization mechanisms to prevent sensor clock drift.

## References

[1] Mangano, S., Saidinejad, H., Veronese, F., Comai, S., Matteucci, M. and Salice, F. (2015) Bridge: Mutual Reassurance for Autonomous and Independent Living. *IEEE Intelligent Systems*, **30**, 31-38. https://doi.org/10.1109/MIS.2015.58

[2] Akyildiz, I.F., Su, W., Sankarasubramaniam, Y. and Cayirci, E. (2002) Wireless Sensor Networks: A Survey. *Computer Networks*, **38**, 393-422.

[3] Coulouris, G., Dollimore, J., Kindberg, T. and Blair, G. (2011) Distributed Systems: Concepts and Design. 5th Edition, Pearson Education, London.

[4] Lecouteux, B., Vacher, M. and Portet, F. (2011) Distant Speech Recognition in a Smart Home: Comparison of Several Multisource ASRs in Realistic Conditions. *Interspeech* 2011, International Speech Communication Association, Florence, August 2011, 2273-2276. https://hal.archives-ouvertes.fr/hal-00642306

[5] Palafox, L.E. and Garcia-Macias, J.A. (2009) Wireless Sensor Networks for Voice Capture in Ubiquitous Home Environments. 4*th International Symposium on Wireless Pervasive Computing*, Melbourne, 11-13 February 2009, 1-5. https://doi.org/10.1109/iswpc.2009.4800614

[6] Alesii, R., Gargano, G., Graziosi, F., Pomante, L. and Rinaldi, C. (2009) WSN-Based Audio Surveillance Systems. In: Mastorakis, N., Mladenov, V. and Kontargyri, V.T., Eds., *Proceedings of the European Computing Conference*, Springer US, 675-681. https://doi.org/10.1007/978-0-387-84814-3_67

[7] Ciuca, D., Pomante, L. and Rinaldi, C. (2012) A Speech Indicator for the VoWSN Approach. 5*th International Symposium on Communications Control and Signal Processing*, Rome, 2-4 May 2012, 1-4. https://doi.org/10.1109/isccsp.2012.6217759

[8] Pomante, L. and Santic, M. (2016) Methodologies, Tools and Technologies for Location-Aware AAL. *IEEE* 2*nd International Forum on Research and Technologies for Society and Industry Leveraging a Better Tomorrow* (*RTSI*), Bologna, 7-9 Sep-

[2]Quotations: second quarter of 2015.

tember 2016, 1-4. https://doi.org/10.1109/RTSI.2016.7740566

[9] Demir, A.K., Turkes, O. and Baydere, S. (2014) Differentiating Emergency Voice Traffic in Indoor Wireless Activity Monitoring Network. *IEEE* 10*th International Conference on Wireless and Mobile Computing, Networking and Communications* (*WiMob*), Larnaca, 8-10 October 2014, 598-603. https://doi.org/10.1109/wimob.2014.6962231

[10] Hu, R.-L., Yin, J.-R., Gu, X.-J., Gu, X.-P. and Chen, L.-Q. (2010) The Research and Design on TDD Voice WSN. 2010 *International Conference on Multimedia Technology* (*ICMT*), Ningbo, 29-31 October 2010, 1-4. https://doi.org/10.1109/icmult.2010.5629848

[11] Ono, N., Kohno, H., Ito, N. and Sagayama, S. (2009) Blind Alignment of Asynchronously Recorded Signals for Distributed Microphone Array. *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, 18-21 October 2009, 161-164. https://doi.org/10.1109/aspaa.2009.5346505

[12] Pololu Corp. (2015) Wixel SDK Documentation. http://pololu.github.io/wixel-sdk/

[13] Abramson, N. (1970) The Aloha System: Another Alternative for Computer Communications. *Proceedings of the Fall Joint Computer Conference*, Houston, 17-19 November 1970, 281-285. https://doi.org/10.1145/1478462.1478502

[14] Wabnitz, A., Epain, N., Jin, C. and van Schaik, A. (2010) Room Acoustics Simulation for Multichannel Microphone Arrays. *Proceedings of the International Symposium on Room Acoustics*, Melbourne, 29-31 August 2010, 1-6.

[15] Rhudy, M., Bucci, B., Vipperman, J., Allanach, J. and Abraham, B. (2009) Microphone Array Analysis Methods Using Cross-Correlations. *ASME* 2009 *International Mechanical Engineering Congress and Exposition*, Lake Buena Vista, Florida, 13-19 November 2009, 281-288. https://doi.org/10.1115/imece2009-10798

[16] MATLAB, Version 8.6.0 (2015) Natick, Massachusetts: The MathWorks Inc., 2015.