

## Article

# Real-Time Implementation of a Frequency Shifter for Enhancement of Heart Sounds Perception on VLIW DSP Platform

Vincenzo Muto <sup>1</sup>, Emilio Andreozzi <sup>1,\*</sup>, Carmela Cappelli <sup>2</sup>, Jessica Centracchio <sup>1</sup>, Gennaro Di Meo <sup>1,\*</sup>, Daniele Esposito <sup>1</sup>, Paolo Bifulco <sup>1</sup> and Davide De Caro <sup>1</sup>

<sup>1</sup> Department of Electrical Engineering and Information Technologies, University of Naples Federico II, Via Claudio, 21, I-80125 Naples, Italy; vincenzo.muto@unina.it (V.M.); jessica.centracchio@unina.it (J.C.); daniele.esposito@unina.it (D.E.); paolo.bifulco@unina.it (P.B.); dadecaro@unina.it (D.D.C.)

<sup>2</sup> Department of Humanities, University of Naples Federico II, Via Porta di Massa, 1, I-80133 Naples, Italy; carcappe@unina.it

\* Correspondence: emilio.andreozzi@unina.it (E.A.); gennaro.dimeo@unina.it (G.D.M.)

**Abstract:** Auscultation of heart sounds is important to perform cardiovascular assessment. External noises may limit heart sound perception. In addition, heart sound bandwidth is concentrated at very low frequencies, where the human ear has poor sensitivity. Therefore, the acoustic perception of the operator can be significantly improved by shifting the heart sound spectrum toward higher frequencies. This study proposes a real-time frequency shifter based on the Hilbert transform. Key system components are the Hilbert transformer implemented as a Finite Impulse Response (FIR) filter, and a Direct Digital Frequency Synthesizer (DDFS), which allows agile modification of the frequency shift. The frequency shifter has been implemented on a VLIW Digital Signal Processor (DSP) by devising a novel piecewise quadratic approximation technique for efficient DDFS implementation. The performance has been compared with other DDFS implementations both considering piecewise linear technique and sine/cosine standard library functions of the DSP. Piecewise techniques allow a more than 50% reduction in execution time compared to the DSP library. Piecewise quadratic technique also allows a more than 50% reduction in total required memory size in comparison to the piecewise linear. The theoretical analysis of the dynamic power dissipation exhibits a more than 20% reduction using piecewise techniques with respect to the DSP library. The real-time operation has been also verified on the DSK6713 rapid prototyping board by Texas Instruments C6713 DSP. Audiologic tests have also been performed to assess the actual improvement of heart sound perception. To this aim, heart sound recordings were corrupted by additive white Gaussian noise, crowded street noise, and helicopter noise, with different signal-to-noise ratios. All recordings were collected from public databases. Statistical analyses of the audiological test results confirm that the proposed approach provides a clear improvement in heartbeat perception in noisy environments.

**Keywords:** heart sounds; pitch shift; digital signal processor; real-time processing; DDFS; piecewise approximation; VLIW



**Citation:** Muto, V.; Andreozzi, E.; Cappelli, C.; Centracchio, J.; Di Meo, G.; Esposito, D.; Bifulco, P.; De Caro, D. Real-Time Implementation of a Frequency Shifter for Enhancement of Heart Sounds Perception on VLIW DSP Platform. *Electronics* **2023**, *12*, 4359. <https://doi.org/10.3390/electronics12204359>

Academic Editors: Leonardo Pantoli, Costas Psychalinos, Gaetano Palumbo, Egidio Ragonese and Paris Kitsos

Received: 7 September 2023

Revised: 13 October 2023

Accepted: 16 October 2023

Published: 20 October 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Cardiac auscultation is still a very important technique for primary cardiovascular assessment of patients. Heart sounds are generated by the flow of blood through the heart chambers as the heart valves open and close during the heart cycle [1–4]. In healthy subjects, two normal sounds can be perceived for each heartbeat, and are commonly, onomatopoeically described as ‘lub’ and ‘dub’.

Normal body sounds have a very low sound level in the range of 22–30 dB in free space. A simple stethoscope is able to render, for the operator’s ear, a significantly higher sound level of 60–65 dB, while also attenuating the contribution of external noises (thus highly

increasing the signal-to-noise ratio). Despite the simplicity of the instrument, efficiency in mechanical transduction is very important in order to render the lower frequencies, at the edge of audibility, of which heart sounds are composed. The spectral content of the main heart sounds (S1 and S2) is confined to low frequencies, spanning up to about 100 Hz. The human ear threshold of audibility in this frequency range is particularly high, then the perceived heart sounds are particularly feeble. Figure 1 schematically shows the spectrum of heart sounds in relation to the human ear threshold of audibility. Heart sounds are too weak to be heard because the human ear is far less sensitive to them with respect to the speech range (1000–2000 Hz).

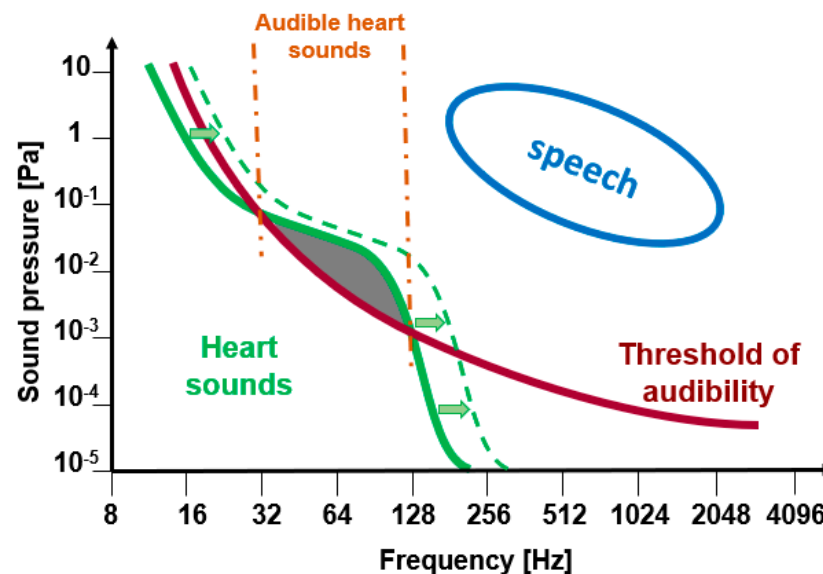


Figure 1. Audibility of heart sounds (inspired by [5,6]).

For this reason, the suppression of external noises from a stethoscope is an important quality parameter along with audio amplification. With the advent of digital stethoscopes, it has been possible to filter and amplify heart sounds to improve their perception [7–10]. Today it is also possible to use special processing [11] to improve diagnostics, as well as to use devices to visualize signals [12]. Moreover, novel promising wearable devices have been proposed, which could enable continuous, long-term monitoring of heart sounds [13–16]. Cardiac auscultation is an invaluable tool to assess the health status of a patient in emergency medicine contexts, specifically in extramural settings, where other cardiac monitoring devices, such as those for electrocardiography and photoplethysmography, or for wireless monitoring of pacemakers [17], are more difficult or impossible to use.

### 1.1. Methods for Noise Suppression

Many researchers have proposed a variety of methods [18] for heart sound denoising, either based on hardware or software solutions. Among software solutions, a first class of algorithms is based on thresholding applied in transform domains, including Short Time Fourier Transform [19–22], Fast Fourier Transform [23–25], Wavelet transform [19,26–35], and Convolution Wavelet Packet Transform [36–39]. However, the choice of the threshold function [35] strongly affects the denoising effect. In addition, with STFT it is impossible to extract some useful clinical information, such as the time distance between aortic valve closure and pulmonic valve closure [19]. A second class of algorithms is based on mode decomposition methods, such as Empirical Mode Decomposition [40–42].

Ensemble Empirical Mode Decomposition [43,44], Complementary Ensemble Empirical Mode Decomposition [45,46], and Variational Mode Decomposition [47–49] are mode decomposition methods that decompose the heart sound signal into a series of intrinsic mode functions (IMFs) and specific algorithms must be designed to select and retain only

the IMFs that capture the main information content of the signal, during the reconstruction of the signal, while discarding those related to noise. However, this class of algorithms, particularly the EMD, is affected by the mode mixing issue, which is caused by the intermittency of signals and noises [42]. In addition, mode decomposition algorithms usually require the a priori selection of the number of IMFs, which affects the decomposition results and determines the impact of mode mixing. EEMD was first conceived to mitigate the mode mixing observed in EMD; however, the reconstructed signal is usually affected by residual noise [49]. CEEMD was proposed precisely to eliminate such residual noise. VMD performs better than EMD and its variants, by realizing an adaptive decomposition of each mode to overcome the mode mixing problem [47]. It has been used to separate heart sounds and lung sounds and to improve the detection of the subtle third heart sound (S3) [48]. However, an accurate selection of the VMD parameters is instrumental in achieving a good decomposition effect [49]. Additional algorithms for heart sound denoising are Total Variation filtering [50,51] and Blind Source Separation [52–54].

Hardware solutions are generally based on the use of additional microphones or PCC sensors, which are used to obtain independent measurements of external noises, and allow enhancing the quality of noisy heart sound recordings via subtractive techniques. This is the case of the spectral subtraction algorithm described in [55], whose performance, however, turned out to be heavily dependent on the characteristics of the additional microphone used to pick up the external noises. A different approach was described in [56], where an array of microphones, appropriately positioned in specific measurement points on the body, and a frequency-domain Independent Component Analysis are used to implement Direction of Arrival and Beamforming techniques for noise suppression in heart sound recordings. The main limitation of such hardware solutions is precisely the need for additional sensors that have to be placed on a subject's body, possibly increasing the subject's discomfort (especially in the case of sensor arrays), and also requiring accurate positioning to ensure effective results.

It is important to highlight that all the methods described above have not been designed for real-time operation, nor to enhance human audibility, but rather to improve computer analysis. For this reason, the ability of these methods to improve human audibility of heart sounds has not been tested on cohorts of human subjects.

### 1.2. Pitch Shift Approaches

Another possibility for improving the perception of heart sounds can be achieved by shifting the heart sounds to higher frequencies. In fact, as the frequency increases, the threshold of audibility lowers, and sounds are perceived even better. Methods based on this approach have proven effective in situations characterized by considerable external noise, such as aboard helicopters [57,58], aircraft [59,60], ambulances [61] or other evacuation vehicles, emergency rooms [62,63], rural clinics [64], and crowded places such as stadiums or crowded streets. Frequency translation can be achieved through digital signal processing techniques. Signal modulation techniques, such as those used for radio or telephone transmissions, can be used for this purpose. Some techniques for translating sounds to the best frequencies perceived by the human ear realize a conspicuous frequency translation of the heart sounds, either to the thousands of Hertz [65] or with a translation of 500 Hz [66], achieving considerable improvements in the acoustic perception of the operator. However, if the pitch is increased significantly, the sounds are perceived as artificial or unreal. Another study [67] has proposed an adjustable frequency modulator with a significantly reduced frequency shift, realized using Hilbert transform. In this case, the Hilbert transformer was implemented as a simple time delay, based on the assumption that heart sounds have a narrow bandwidth. In addition to stethoscopes with frequency-shifting capabilities, hybrid stethoscopes have been proposed, which embed ultrasonic sensors that make the Doppler signal audible, so as to enable cardiac diagnostics in scenarios of intense external noise [68,69]. The enhancements of human audibility of heart sounds provided by the methods proposed in [66,67] have not been tested on an actual cohort of human

subjects. The method proposed in [65] appears to have been tested on a cohort of human subjects, but the kinds and levels of noise adopted have not been specified. The methods proposed in [68,69] require very peculiar stethoscopes with ultrasonic sensors, which are not very common.

In summary, there is a gap in the literature regarding an accurate, real-time method for frequency shifting of heart sounds, where the ability to improve the human audibility of heart sounds has been evaluated by experimental testing on a real cohort of human subjects.

The present study aimed to fill this gap and describes the design, implementation, and experimental testing of a novel, real-time, frequency shifter for heart sounds, implemented on a Very-Long-Instruction-Word (VLIW) Digital Signal Processor (DSP). The proposed method is based on the Hilbert transform and provides two improvements with respect to the solution of [67]: (i) the Hilbert transformer is implemented using an FIR filter to improve signal quality; (ii) a Direct Digital Frequency Synthesizer (DDFS) is used to allow the agile modification of the frequency shift. The proposed implementation allows translating the spectrum of heart sounds even by only a few tens of Hertz. A novel piecewise quadratic technique is also proposed for the implementation of the DDFS and demonstrates the best trade-off between execution time, memory resources, and power consumption. Auscultation tests were performed on 41 volunteers to assess the actual performance of the proposed method. The results confirm the capability of the proposed method to enhance the audibility of heart sounds in the presence of different kinds of noise. The frequency shifter here described could be easily embedded in modern electronic stethoscopes at very modest costs, thus providing valuable support in emergency medicine contexts, such as those described in [58–65], where heart sounds auscultation is fundamental to assess the health status of a patient and could be seriously hindered by environmental noises.

## 2. Materials and Methods

### 2.1. Top-Level Algorithm

The top-level architecture of the algorithm used to implement the frequency shift operation is shown in Figure 2. The operation is achieved using a single-sideband modulator realized using the Hilbert transform. The behavior of this solution can be briefly explained as follows. The Hilbert transform, which can be considered to be a linear system with a frequency response:

$$H(f) = \begin{cases} -j & f > 0 \\ j & f < 0 \end{cases} \quad (1)$$

is used to form the analytical signal  $x_c(n)$  associated with  $x(n)$  as follows:

$$x_c(n) = x(n) + j x_H(n) \quad (2)$$

$x_H(n)$  being the Hilbert transform of  $x(n)$ . As is well-known, the Fourier transform of the associated analytical signal includes only the positive-frequency axis components of  $X(n)$ , as follows:

$$X_c(f) = \begin{cases} 2X(f) & f > 0 \\ 0 & f < 0 \end{cases} \quad (3)$$

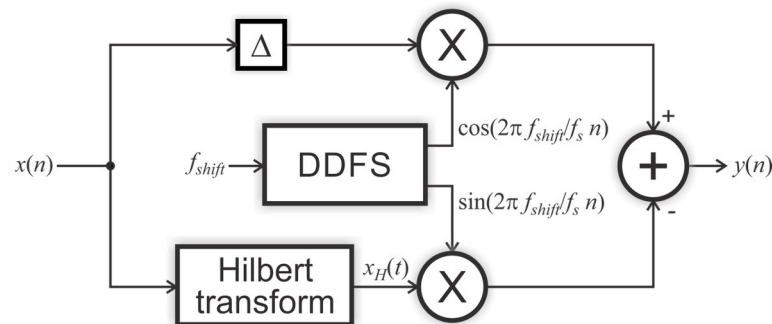
The frequency shift operation, therefore, can be computed by multiplying the complex signal  $x_c(t)$  by  $e^{j2\pi f_{shift}/f_s n}$ ,  $f_{shift}$  being the desired frequency shift and  $f_s$  being the sampling frequency, and taking the real part, in order to re-obtain a real signal with Hermitian Fourier Transform, obtaining the following:

$$Y(f) = \begin{cases} X(f - f_{shift}) & f > f_{shift} \\ 0 & 0 \leq f < f_{shift} \end{cases} \quad (4)$$

where clearly:

$$y(n) = \text{Re} \left[ x_c(n) \cdot e^{j2\pi f_{shift}/f_s n} \right] \quad (5)$$

The architecture of Figure 2 implements (5) using a Direct Digital Frequency Synthesizer (DDFS) to compute the two signals  $\cos(2\pi f_{\text{shift}}/f_s n)$  and  $\sin(2\pi f_{\text{shift}}/f_s n)$ .



**Figure 2.** Top-level architecture used to implement frequency shift operation.

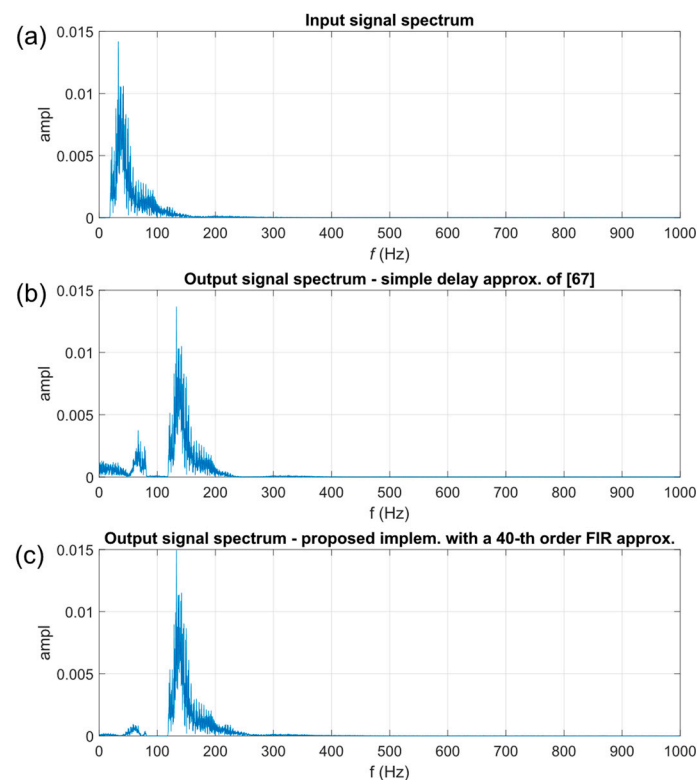
Please note that a similar technique has been devised in [67], where, however, oscillators are used to obtain sine/cosine signals and the Hilbert transform is approximated using a simple delay, chosen in order to realize the required  $\pi/2$  phase shift (see (1)) only at a given frequency,  $f_{\text{mid}}$ , chosen in the middle of the input signal bandwidth, i.e., by calling  $\delta$  the implemented delay,  $2\pi \cdot f_{\text{mid}} \cdot \delta = \pi/2$ . This unfortunately leads to obtaining a not negligible output noise power at frequencies lower than  $f_{\text{shift}}$  (low-sideband noise power) since in the architecture of Figure 2, the final adder which produces  $y(n)$  actually implements a cancellation between the two adder inputs for  $f < f_{\text{shift}}$  and, when the Hilbert transform is implemented with a simple delay, the cancellation is perfectly realized only at frequency  $f_{\text{shift}} - f_{\text{mid}}$ . As an example, Figure 3a shows the spectrum of a cardiac audio signal sampled with a frequency of 2 kHz and used as an input signal for the system of Figure 2, and Figure 3b shows the corresponding elaborated signal when the Hilbert transform is implemented with a simple delay, using  $f_{\text{mid}} = 50$  Hz (this value has been chosen in order to minimize the low-sideband noise power). As it can be observed, the output spectrum nullifies only at frequency  $f_{\text{shift}} - f_{\text{mid}} = 50$  Hz, and no negligible frequency components can be seen in the low-sideband. As reported in Table 1, the overall ratio between the low-sideband noise power and the output power is  $-11.9$  dB.

**Table 1.** Ratio between low-sideband noise power and output power by considering different techniques to implement the Hilbert transform.

Technique	Low-Sideband Noise Power/Output Power
Simple delay approx. [67]	$-11.9$ dB
Proposed with 20-th order FIR filter	$-21.2$ dB
Proposed with 40-th order FIR filter	$-23.3$ dB
Proposed with 60-th order FIR filter	$-30.3$ dB
Proposed with 80-th order FIR filter	$-37.7$ dB
Proposed with 100-th order FIR filter	$-42.9$ dB

To reduce the low-sideband noise, in this paper, we implemented the Hilbert transform using a linear-phase FIR filter, whose coefficients have been obtained using the Parks–McClellan optimal FIR filter design algorithm implemented in Matlab. Please note that in this case, the FIR filter implements the Hilbert transform with a delay given by half of the filter order. To compensate for this delay, a delay  $\Delta$  also equal to one-half the filter order is added in Figure 2 at the input of the multiplier-driven by  $x(t)$ . Figure 3c shows the obtained output spectrum in the case of a 40th-order FIR filter. Table 1 summarizes the achieved ratio between low-sideband noise power and output power in the case of FIR filter orders 20, 40, 60, 80, and 100. It can be observed that a significant reduction of the low-sideband noise power can be achieved with respect to the previous approach of [67]. In the following sections, we will investigate the implementation with a 40th-order filter.





**Figure 3.** Simulation results obtained elaborating a cardiac audio signal (sampling frequency  $f_s = 2$  kHz): (a) input signal spectrum; (b) output signal spectrum obtained using the approach of [67] to compute the Hilbert transform; (c) output signal spectrum obtained using a 40th order FIR filter to compute Hilbert transform.

## 2.2. Implementation

In this paper, we will focus on an implementation using a Texas Instruments VLIW DSP. In the following sections, we will describe in detail the implementation of the two main components of the algorithm of Figure 2 that is the Hilbert transform, implemented using a FIR filter, and the DDFS.

### 2.2.1. FIR Filter Implementation (Hilbert Transform)

As well-known DSPs are specifically designed to implement in an efficient way FIR filters. The exploited implementation (shown in Figure 4) uses a simple direct form implementation of the filter, using an array managed as a linear buffer to store the input samples  $x(n), x(n-1), \dots, x(n-M)$ ,  $M$  being the filter order. Please note that the delay  $\Delta$  shown in Figure 2 can be easily realized by taking the sample  $x(n-M/2)$  from the array.

In the figure, the notation  $Q_{m,n}$  is used to indicate a signed 2's complement representation with a most-significant bit of weight  $2^m$  and a less-significant bit of weight  $2^{-n}$ . As can be observed, the input is assumed contained in the range  $[-1, 1)$  and is represented in  $Q_{0,15}$  using 16 bits. This size fits the data-size of the inputs of the multiplier available in the DSP. The filter coefficient,  $h_i$ , is also assumed represented in  $Q_{0,15}$ , using 16 bits. To manage the overflow, the sum-of-products is computed using the guard-bits approach and the available data-size provided by the DSP is computing the sum using 40 bits in  $Q_{9,30}$ . The filter output is obtained in  $Q_{0,15}$  (note that the 16-bits representation is also needed to compute the multiplication between  $x_H(n)$  and  $\sin(2\pi f_{shift}/f_s n)$  in Figure 2) using a rounding operation on the less-significant bits and a saturation operation to reduce the most-significant bits. The employ of saturation helps in mitigating the limited overflow cases that could happen whether a full-scale, large bandwidth input signal is applied.

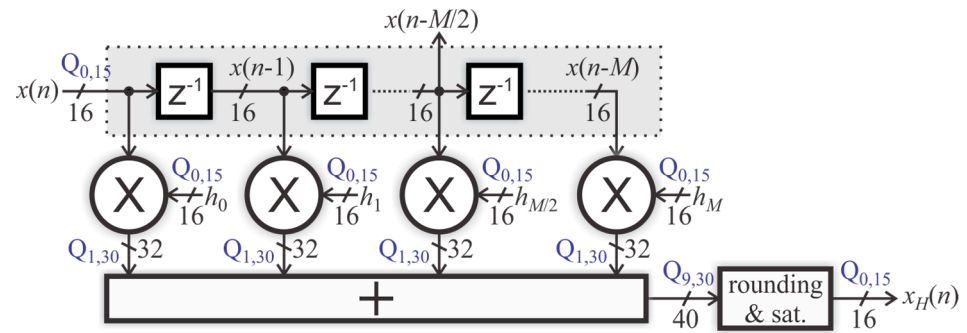


Figure 4. FIR filter implementation (Hilbert transform).

2.2.2. DDFS Implementation

The top-level DDFS implementation is shown in Figure 5. A phase accumulator is used to compute instantaneous phase  $\varphi/2\pi$  encoded in an unsigned  $N$ -bits representation (in our implementation  $N = 32$  to fit DSP register size). This phase is then truncated to  $P$ -bits. The heart of the DDFS is the two blocks that compute sine and cosine functions in  $[0, \pi/4)$  interval that is the two functions:

$$y_s(z) = \text{Sin} \left[ \frac{\pi}{4} \cdot z + \varphi_{LSB} \right] \tag{6}$$

$$y_c(z) = \text{Cos} \left[ \frac{\pi}{4} \cdot z + \varphi_{LSB} \right] \tag{7}$$

for  $z \in [0, 1)$ . The 1's complement, the swapper, and the two inverters (inv.), driven by the mapper, are used to reconstruct the output sine and cosine functions in the whole  $[0, 2\pi)$  interval. Please note that in (6) and (7),  $\varphi_{LSB}$  represents the angle corresponding to the one-half LSB of signal  $\varphi/2\pi$  after  $P$ -bits truncation, which is:  $\varphi_{LSB} = 2\pi \cdot 2^{-P}/2$ .

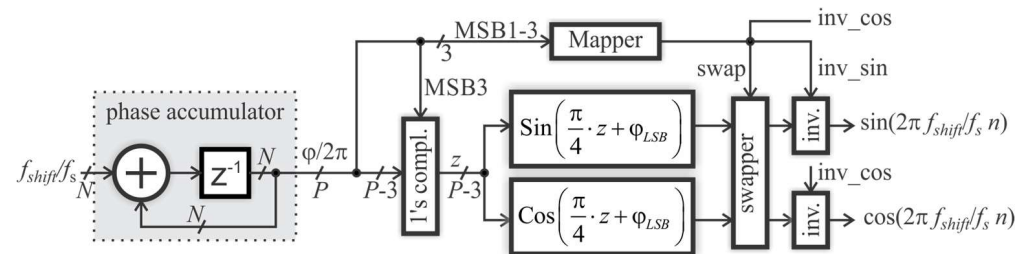


Figure 5. Top-level DDFS implementation.

In the literature, many approaches have been proposed to implement (6) and (7), including simple Lookup-Table (LUT) methods, LUT-compression techniques, piecewise approximations, angle rotation approaches (CORDIC algorithm), and mixed LUT and angle rotation techniques. In this paper, we introduce an extension of the piecewise linear technique proposed in [70] to the piecewise quadratic case.

In piecewise approximation techniques the interval  $[0, 1)$  of  $z$  is subdivided into  $2^H$  sub-intervals, each of length  $h = 2^{-H}$ . Since we consider a piecewise quadratic approximation, in the generic  $i$ -th sub-interval, the two functions  $y_s(z)$  and  $y_c(z)$  are approximated as follows:

$$y_s(z) \simeq a_{0i} + a_{1i} \cdot z_i + a_{2i} \cdot z_i^2 + \varepsilon_{s2i}(z_i) \tag{8}$$

$$y_c(z) \simeq b_{0i} + b_{1i} \cdot z_i + b_{2i} \cdot z_i^2 + \varepsilon_{c2i}(z_i) \tag{9}$$

where  $z_i$  represents the distance between  $z$  and the starting point of the sub-interval:  $z_i = z - i \cdot h$ .

By following the approach of [70], we consider a truncated Chebyshev expansion in order to obtain the coefficients  $a_{ji}$  and  $b_{ji}$  of (8) and (9) which approximate (6) and (7) in the quasi-minimum maximum error sense. To that purpose the variable  $z_i$  is substituted by variable  $w$  so that the interval  $[i \cdot h, (i + 1) \cdot h]$  of  $z_i$  corresponds to the interval  $[-1, +1]$  of  $w$  that is:

$$w = \frac{2}{h} \cdot z_i - 1 \tag{10}$$

Using this substitution, the functions to be approximated in the  $i$ -the sub-interval becomes:

$$y_{si}(w) = \text{Sin} \left[ \frac{\pi}{4} \cdot \left( \frac{h}{2} \cdot w + \frac{h}{2} + i \cdot h \right) + \varphi_{LSB} \right] \tag{11}$$

$$y_{ci}(w) = \text{Cos} \left[ \frac{\pi}{4} \cdot \left( \frac{h}{2} \cdot w + \frac{h}{2} + i \cdot h \right) + \varphi_{LSB} \right] \tag{12}$$

These two functions are expanded in a series of Chebyshev polynomials of the first kind ( $T_n(w)$ ):

$$y_{si}(w) \simeq \alpha_{0i} \cdot T_0(w) + \alpha_{1i} \cdot T_1(w) + \alpha_{2i} \cdot T_2(w) + \alpha_{3i} \cdot T_3(w) + \dots \tag{13}$$

$$y_{ci}(w) \simeq \beta_{0i} \cdot T_0(w) + \beta_{1i} \cdot T_1(w) + \beta_{2i} \cdot T_2(w) + \beta_{3i} \cdot T_3(w) + \dots \tag{14}$$

where we recall that Chebyshev polynomials of the first kind  $T_n(w)$  can be written using the following recursion:

$$\begin{cases} T_0(w) = 1 \\ T_1(w) = w \\ T_{n+1}(w) = 2w \cdot T_n(w) - T_{n-1}(w) \end{cases} \tag{15}$$

Since Chebyshev polynomials are orthogonal in the general sense in  $[-1, +1]$  interval, the coefficient in (13) and (14) can be obtained as follows:

$$\alpha_{ji} = \frac{\int_{-1}^{+1} \frac{1}{\sqrt{1-w^2}} \cdot y_{si}(w) \cdot T_j(w) dw}{\int_{-1}^{+1} \frac{1}{\sqrt{1-w^2}} \cdot T_j^2(w) dw} \tag{16}$$

$$\beta_{ji} = \frac{\int_{-1}^{+1} \frac{1}{\sqrt{1-w^2}} \cdot y_{ci}(w) \cdot T_j(w) dw}{\int_{-1}^{+1} \frac{1}{\sqrt{1-w^2}} \cdot T_j^2(w) dw} \tag{17}$$

By solving the integrals, we obtain:

$$\alpha_{0i} = J_0 \left( \frac{\pi}{8} \cdot h \right) \cdot \text{Sin} \left[ \frac{\pi}{4} \cdot \left( i \cdot h + \frac{h}{2} \right) + \varphi_{LSB} \right] \tag{18}$$

$$\alpha_{1i} = 2 \cdot J_1 \left( \frac{\pi}{8} \cdot h \right) \cdot \text{Cos} \left[ \frac{\pi}{4} \cdot \left( i \cdot h + \frac{h}{2} \right) + \varphi_{LSB} \right] \tag{19}$$

$$\alpha_{2i} = -2 \cdot J_2 \left( \frac{\pi}{8} \cdot h \right) \cdot \text{Sin} \left[ \frac{\pi}{4} \cdot \left( i \cdot h + \frac{h}{2} \right) + \varphi_{LSB} \right] \tag{20}$$

$$\alpha_{3i} = -2 \cdot J_3 \left( \frac{\pi}{8} \cdot h \right) \cdot \text{Cos} \left[ \frac{\pi}{4} \cdot \left( i \cdot h + \frac{h}{2} \right) + \varphi_{LSB} \right] \tag{21}$$

$$\beta_{0i} = J_0 \left( \frac{\pi}{8} \cdot h \right) \cdot \text{Cos} \left[ \frac{\pi}{4} \cdot \left( i \cdot h + \frac{h}{2} \right) + \varphi_{LSB} \right] \tag{22}$$



$$\beta_{1i} = -2 \cdot J_1\left(\frac{\pi}{8} \cdot h\right) \cdot \sin\left[\frac{\pi}{4} \cdot \left(i \cdot h + \frac{h}{2}\right) + \varphi_{LSB}\right] \tag{23}$$

$$\beta_{2i} = -2 \cdot J_2\left(\frac{\pi}{8} \cdot h\right) \cdot \cos\left[\frac{\pi}{4} \cdot \left(i \cdot h + \frac{h}{2}\right) + \varphi_{LSB}\right] \tag{24}$$

$$\beta_{3i} = 2 \cdot J_3\left(\frac{\pi}{8} \cdot h\right) \cdot \sin\left[\frac{\pi}{4} \cdot \left(i \cdot h + \frac{h}{2}\right) + \varphi_{LSB}\right] \tag{25}$$

where  $J_n(x)$  represents the  $n$ -th order Bessel function of the first kind.

The quasi-minimum maximum error quadratic approximation of  $y_s$  and  $y_c$  can be obtained by truncating (13) and (14) to the second order that is:

$$y_s(z) \simeq \alpha_{0i} \cdot T_0(w) + \alpha_{1i} \cdot T_1(w) + \alpha_{2i} \cdot T_2(w)|_{w=\frac{2}{h}z_i-1} \tag{26}$$

$$y_c(z) \simeq \beta_{0i} \cdot T_0(w) + \beta_{1i} \cdot T_1(w) + \beta_{2i} \cdot T_2(w)|_{w=\frac{2}{h}z_i-1} \tag{27}$$

By comparing (26) and (27) with (8) and (9), and using (15) we obtain:

$$a_{0i} = \alpha_{0i} - \alpha_{1i} + \alpha_{2i} \quad a_{1i} = \frac{2}{h} \cdot (\alpha_{1i} - 4\alpha_{2i}) \quad a_{2i} = \frac{8}{h} \cdot \alpha_{2i} \tag{28}$$

$$b_{0i} = \beta_{0i} - \beta_{1i} + \beta_{2i} \quad b_{1i} = \frac{2}{h} \cdot (\beta_{1i} - 4\beta_{2i}) \quad b_{2i} = \frac{8}{h} \cdot \beta_{2i} \tag{29}$$

It is worth noting that the first-order approximation of [70] can be easily re-obtained by imposing  $\alpha_{2i} = \beta_{2i} = 0$  in (28) and (29).

The approximation error in (8) and (9) can be approximated as the first term dropped from the series (13) and (14):

$$\varepsilon_{s2i}(z_i) \simeq \alpha_{3i} \cdot T_3(w)|_{w=\frac{2}{h}z_i-1} \quad ; \quad \varepsilon_{c2i}(z_i) \simeq \beta_{3i} \cdot T_3(w)|_{w=\frac{2}{h}z_i-1} \tag{30}$$

Since every Chebyshev Polynomial  $T_j(w)$  in contained in the interval  $[-1, +1]$ , the maximum errors are given by (see also (21) and (25)):

$$\max|\varepsilon_{s2i}| \simeq |\alpha_{3i}| = 2 \cdot J_3\left(\frac{\pi}{8} \cdot h\right) \cdot \cos\left[\frac{\pi}{4} \cdot \left(i \cdot h + \frac{h}{2}\right) + \varphi_{LSB}\right] \tag{31}$$

$$\max|\varepsilon_{c2i}| \simeq |\beta_{3i}| = 2 \cdot J_3\left(\frac{\pi}{8} \cdot h\right) \cdot \sin\left[\frac{\pi}{4} \cdot \left(i \cdot h + \frac{h}{2}\right) + \varphi_{LSB}\right] \tag{32}$$

In this case, the worst-case error is obtained for the approximation of the sine function in the first sub-interval ( $i = 0$ ). Therefore, the overall maximum approximation error due to piecewise quadratic approximation ( $\varepsilon_{alg}$ ) is given by:

$$\varepsilon_{alg} = 2 \cdot J_3\left(\frac{\pi}{8} \cdot h\right) \simeq \frac{1}{192} \left(\frac{\pi}{4} \cdot h\right)^3 \tag{33}$$

where we have used the approximation:

$$J_n(x) \simeq \frac{1}{n!} \cdot \left(\frac{x}{2}\right)^2 \quad \text{for } x \ll 1 \tag{34}$$

By proceeding in a similar way for a piecewise linear approximation, we find:

$$\varepsilon_{alg} = 2 \cdot J_2\left(\frac{\pi}{8} \cdot h\right) \simeq \frac{1}{16} \left(\frac{\pi}{4} \cdot h\right)^2 \tag{35}$$

Table 2 reports the achieved  $\varepsilon_{alg}$  by varying the number of sub-intervals ( $2^H$ ). Please note that according to DSP data-sizes, we plan to implement a DDFS with 16-bit outputs with a weight of the less-significant bit  $LSB_y = 2^{-15}$ . For this reason, the error values reported in the table are normalized to  $LSB_y$ . By looking at reported values, it can be observed that if we would like to achieve a  $\varepsilon_{alg}$  lower than  $1/4 LSB_y$ , the piecewise linear approximation requires 128 sub-intervals. On the other hand, using the piecewise quadratic approximation developed in this paper only eight sub-intervals can be used; this reduces significantly the size of the arrays needed to store the coefficients (28) and (29).

**Table 2.** Achieved  $\varepsilon_{alg}$  by varying the number of sub-intervals ( $LSB_y = 2^{-15}$ ). The values with grey background are the ones that were used for the actual implementation.

Number of Sub-Intervals ( $2^H$ )	Piecewise Linear Approx	Piecewise Quadratic Approx
	$\varepsilon_{alg}/LSB_y$	$\varepsilon_{alg}/LSB_y$
4	78.893	1.291
8	19.735	0.161
16	4.935	0.020
32	1.234	$2.52 \times 10^{-3}$
64	0.308	$3.15 \times 10^{-4}$
128	0.077	$3.94 \times 10^{-5}$

Figure 6 shows the implementation scheme for the sine computation block of the DDFS shown in Figure 5 (a similar scheme applies also to the cosine block), where Figure 6a considers the piecewise linear technique and Figure 6b focuses on the piecewise quadratic approach. Signal representation is reported using the notation  $U_{m,n}$  and  $Q_{m,n}$  for unsigned and signed representations, respectively. It is assumed that in both cases,  $2^m$  represents the weight of the most-significant bit, while  $2^{-n}$  represents the weight of the less-significant bit. Please note that the number of bits of all signals is chosen in order to meet the fixed-point hardware operators of the DSP (that is 16-bit input and 32-bit output for the multipliers and 32-bit input and output for the adders). If, for the sake of simplicity, we focus on the piecewise quadratic implementation of Figure 6b, since the number of sub-intervals is chosen equal to  $2^H$ , the  $H = 3$  most-significant bits of  $z$  can be used to obtain the index  $i$  which identifies the sub-interval; the remaining bits of  $z$ , represents  $z_i = z - i \cdot h$ . The data-size of  $z$  is chosen in order to meet the 16-bit size of  $z_i$ , since this signal is the input of the following multipliers. The coefficients  $a_{1i}$  and  $a_{2i}$  are stored on 16 bits, resulting in inputs of the multipliers. In this case, the unsigned representation is chosen since  $a_{1i}$  and  $-a_{2i}$  results always positive. The coefficients  $a_{0i}$ , being required only at the input of an adder, can be stored on 32 bits, in  $Q_{0,31}$  representation. A final rounding operation is used to provide the output in  $Q_{0,15}$  representation.

The results of the complete error analysis of the two DDFS implementations (using piecewise linear or piecewise quadratic approximations), are reported in Table 3. It is worth noting that rounding on  $z$  signal is implemented by initializing the phase accumulator in Figure 2 to the rounding constant  $0.5 \cdot 2^{-P}$ . It can be observed that some error components are almost negligible, and this allows us to employ the simpler approach of truncation in place of rounding both at the output of the multiplier computing  $z^2$  in Figure 6b and at the inputs of the adder in Figure 6a,b. The theoretical overall maximum absolute error of piecewise linear DDFS is  $0.581 LSB_y$ , while in the case of piecewise quadratic approximation the error is slightly higher ( $0.722 LSB_y$ ). In both cases, the overall error close to  $0.5 LSB_y$  confirms the accurate sizing described in Figure 6. We have also simulated the two DDFS implementations and the simulated values are reported in the last column of Table 3. It can be observed that in the case of piecewise quadratic DDFS, the simulated maximum error is slightly lower than the theoretical one; this is due to the fact that the theoretical error components considered in Table 3 tend to compensate for each other: as an example the error of  $a_{1i}$  coefficient is maximum for maximum  $z_i$  and is positive while in the same condition, the error due to piecewise quadratic approximation is maximum in modulus

but negative. In the following section, we will consider the simulated values of the DDFS errors.

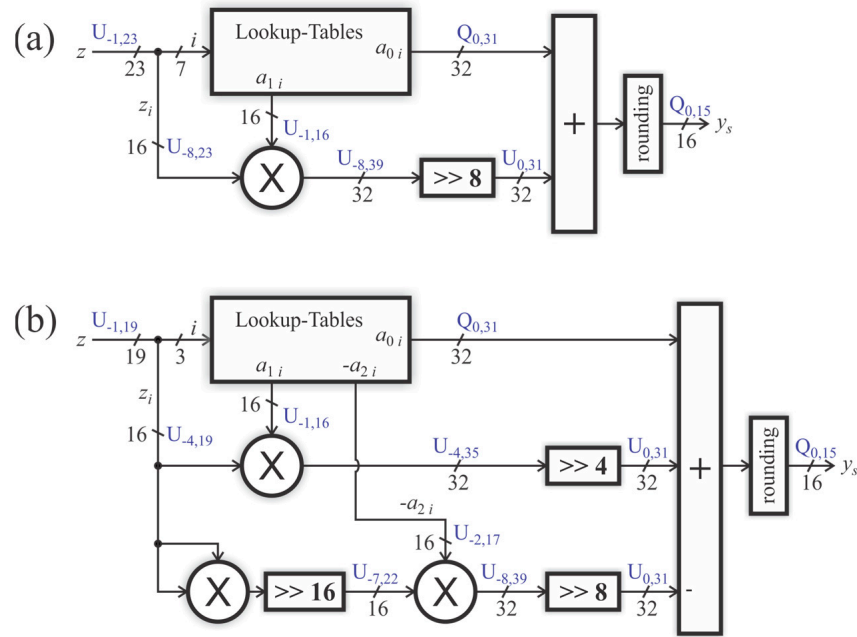


Figure 6. Implementation scheme for sine block of the DDFS in Figure 5: (a) piecewise linear approximation; (b) piecewise quadratic approximation.

Table 3. DDFS error analysis and simulation. The total errors columns are highlighted in grey.

Maximum Absolute Errors Normalized to $LSB_y$									Total Error	Total Error (Simulat.)
Piecewise Approx. ( $\epsilon_{alg}$ )	$a_{0i}$ Coeff.	$a_{1i}$ Coeff.	$a_{2i}$ Coeff.	$z$ Signal	$z^2$ Signal	Inputs of the Adder in Figure 6	Adder Output in Figure 6			
LSB Reduction Method	-	Rounding	Rounding	Rounding	Rounding	Trunc.	Trunc.	Rounding	-	-
Piecewise linear technique	0.077	$7.63 \times 10^{-6}$	0.002	-	0.002	-	$1.53 \times 10^{-5}$	0.500	0.581	0.576
Piecewise quadratic technique	0.161	$7.63 \times 10^{-6}$	0.031	$5.96 \times 10^{-8}$	0.025	0.005	$3.05 \times 10^{-5}$	0.500	0.722	0.684

If we call  $\epsilon_{DDFS}$  the total maximum absolute value of the DDFS error (reported in the last column of Table 3), the overall maximum absolute error of the whole elaboration of Figure 2 can be written as:

$$\epsilon_{elab\ total} = 2 \cdot \epsilon_{DDFS} \cdot \max|x(n)| + \epsilon_{FIR} + 0.5 \cdot LSB_y \tag{36}$$

where  $\epsilon_{FIR}$  represents the total error of the FIR filter used to implement the Hilbert transform and the final contribution of 0.5  $LSB_y$  is due to the final rounding operation needed to obtain the final output  $y(n)$  in  $Q_{0,15}$  representation. Please note, however, that this simple relationship describes a worst-case behavior: as an example, if we look at (31) and (32) the sum of the absolute errors due to piecewise approximation at the two outputs of the DDFS is equal to  $\sqrt{2}$  times the error of the single  $W$  output, not to two times this error.

Since the only approximation in Figure 4 is the final rounding, we have that also  $\epsilon_{FIR}$  is equal to 0.5  $LSB_y$ . As a consequence,  $\epsilon_{elab\ total}$  can be written as:

$$\epsilon_{elab\ total} = LSB_y + 2 \cdot \epsilon_{DDFS} \cdot \max|x(n)| \tag{37}$$

For a worst-case input signal, with maximum value equal to 1,  $\varepsilon_{elab\ total}$  is therefore equal to 2.15 LSB<sub>y</sub> and 2.37 LSB<sub>y</sub> for piecewise linear and quadratic DDFS, respectively. The error becomes lower for an input signal that is not at full-scale. As an example, for a maximum absolute input value of 1/3, the errors for piecewise linear and quadratic DDFS implementations becomes 1.38 LSB<sub>y</sub> and 1.46 LSB<sub>y</sub>, respectively. These theoretical values describe again a worst-case condition since error components in the real system tend not to be at the maximum at the same time or tend to cancel each other out. As an example, by simulating the whole algorithm using a random input signal with 1/3 maximum absolute value, the simulated maximum absolute error results were 1.13 LSB<sub>y</sub> and 1.18 LSB<sub>y</sub>, respectively.

### 2.3. Experimental Auscultation Tests

The actual performance of the proposed frequency shift enhancement was assessed by carrying out specific auscultation tests. To this aim, 41 participants (23 males, 18 females, age  $34.3 \pm 11.6$ ) not skilled in medical auscultation and with no history of hearing impairment were enrolled. Each participant was asked to listen to short audio clips consisting of a heart sound recording (from the public database “Classification of Heart Sound Recordings: The PhysioNet/Computing in Cardiology Challenge 2016” [71]) corrupted by noise, either unprocessed, or processed with the proposed frequency shift enhancement. Three noise sources were considered, namely additive white Gaussian noise (AWGN), crowded avenue noise (publicly available at [72]), and helicopter noise (publicly available at [73]). For each noise source, several noisy audio clips were created by corrupting each heart sound recording, either unprocessed or processed via the proposed algorithm, with different noise levels, so as to obtain values of signal-to-noise ratio (SNR) ranging from 0.025 to 1 (steps of 0.025 up to 0.3, and steps of 0.1 from 0.3 up to 1). In the case of processed heart sounds, two frequency shift ( $f_{shift}$ ) values were considered, namely 50 Hz and 100 Hz. For each noise source, three sets of 19 audio clips were obtained, namely unprocessed (referred to as “no\_elab”), processed with 50 Hz shift (referred to as “elab\_50”), and processed with 100 Hz shift (referred to as “elab\_100”). Therefore, each participant listened to 171 audio clips, divided into three sessions (one for each noise source), with 5-min breaks in between. In each session, the sets of unprocessed and processed audio clips were administered separately, with a random sequence of SNR values. The participants, who were blind to the specific set being administered, were asked to acknowledge their ability to reliably perceive the heart sounds; the lowest SNR ( $SNR_{min}$ ) value for which a participant confirmed the ability to hear the heart sounds was considered to be the minimum hearing level for the related set and noise source. At the end of the experimental auscultation tests, for each participant, the  $SNR_{min}$  values for no\_elab, elab\_50, and elab\_100 sets of clips were obtained for each noise source (i.e., AWGN, crowded avenue, helicopter).

### 2.4. Statistical Analyses

Statistical analyses of the results of the experimental auscultation tests were carried out to verify if the proposed method enhanced the participants’ ability to perceive heart sounds in the presence of different kinds of noise. An enhancement in hearing ability for a specific noise source would result in a reduction in the  $SNR_{min}$  obtained for the processed clips (elab\_50, elab\_100) with respect to the unprocessed clips (no\_elab). Considering that the effects of the proposed frequency shift could change between the participants, specific tests were performed to assess if the differences between the  $SNR_{min}$  values of processed and unprocessed clips were statistically lower than 0. In particular, for each noise source, the Wilcoxon Signed-Rank Test was applied to  $SNR_{min}$  matched pairs of elab\_50 and no\_elab clips, and to  $SNR_{min}$  matched pairs of elab\_100 and no\_elab clips.

### 3. Results

#### 3.1. DSP Implementation Results

The algorithm presented in the previous section has been described in the C programming language and optimized for implementation on Texas Instruments C67xx DSPs (Texas Instruments, Dallas, TX, USA). Table 4 reports the achieved performance by considering the DDFS implementation. In this case, the execution time was obtained by simulating the code on a DSP simulator with an ideal memory model (i.e., without considering cache). It can be observed that a total of five implementations were considered. The first implementation used the piecewise linear approximation technique described in the previous section and exploited a “far allocation” approach to implement the Lookup Table (LUT) needed to store  $a_{ji}$  and  $b_{ji}$  coefficients. The second implementation still used a piecewise linear approximation technique, but in this case, the LUT employed a “near allocation” technique, i.e., it was stored in the BSS data section (as well-known, this approach reduces the time needed to access the coefficients). The third implementation used the piecewise quadratic approach described in the previous section; in this case, only “near allocation” was considered since the required LUT size was very small (only 128 bytes). The last two implementations used the DDFS architecture of Figure 5 and computed the required sine and cosine functions using the optimized floating-point functions available in the FastRTS Library provided with the DSP. In this case, two implementations were considered, one using single-precision sine/cosine functions, and one using double-precision functions. Considering the total memory size required to implement the DDFS, it can be noted that the best solution is the piecewise quadratic implementation, which requires only 384 bytes. The piecewise linear implementation resulted in the largest required memory size, which is dominated by the required LUT size (1536 bytes). In addition, the piecewise linear DDFS using near allocation required also a non-negligible 4.7% of the total BSS section size, which was used precisely to store variables using “near allocation”. By considering, on the other hand, the execution time, the fastest implementation was the piecewise linear DDFS using “near allocation”, which was executed in only 19 clock cycles. Also remarkable is the performance of the piecewise quadratic implementation, which was executed in 21 clock cycles and turned out to be also faster than the piecewise linear DDFS with “far allocation”. The two implementations using library functions, with single and double precision, were much slower than other implementations, requiring 177 and 346.5 clock cycles, respectively. Please note that this last value is not an integer because the execution time turned out to also be data-dependent. It is also interesting to observe that the proposed fixed-point implementations outperform standard floating-point implementations using FastRTS Library, despite the DSP being able to implement floating-point arithmetic in hardware.

**Table 4.** DDFS implementation results (C67xx DSP—ideal memory model).

Sin/Cos Computation Technique	LUT Storage Tech.	LUT Size (Bytes)	Code Size (Bytes)	Total Memory Size (Bytes)	Percentage BSS Section Occupation	Execution Time (Clock Cycles)
Piecewise Lineax Approx.	far alloc.	1536	224	1760	-	25.0
Piecewise Lineax Approx.	near alloc.	1536	224	1760	4.7%	19.0
Piecewise Quadratic Approx.	near alloc.	128	256	384	0.4%	21.0
Library functions—Single-Precision	-	-	928	928	-	177.0
Library functions—Double-Precision	-	-	1120	1120	-	346.5

Table 5 summarizes the performances achieved by considering the whole elaboration algorithm described in the previous sections, by considering the five possible DDFS implementations, also outlined in Table 4. As in Table 4, a DSP simulator with an ideal memory model was considered. By looking at the execution time of the FIR filter, it can be observed that the performances achieved are coherent with the approach used to realize and optimize the filter, which is using a linear buffer algorithm for filter input and implementing the

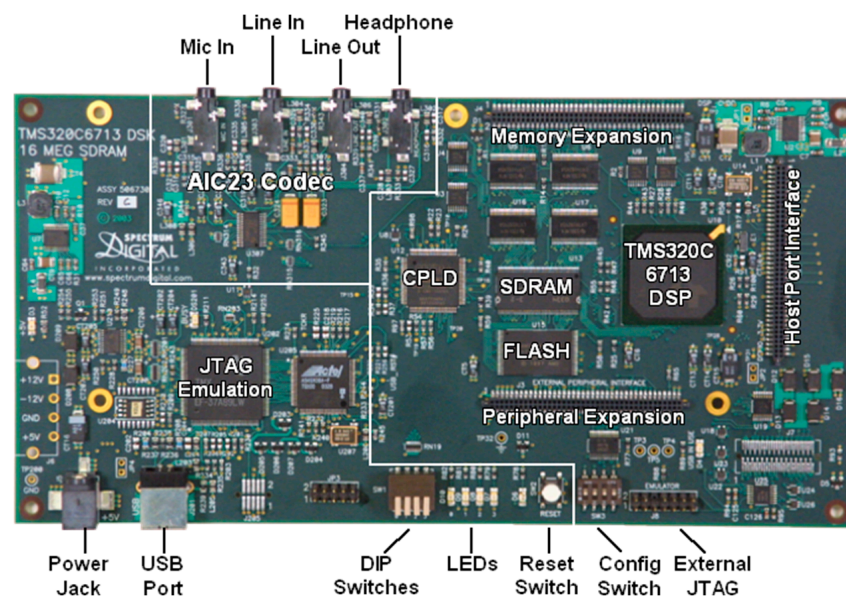


sum-of-products with a software-pipelined loop using  $2 \times$  unrolling (the considered filter order is 40). Overall, the data presented confirm that the piecewise quadratic implementation resulted in the best tradeoff between memory size and execution time, with a required memory size of about 1.1 KB and a total execution time of only 124 clock cycles for the whole algorithm. The fastest implementation is again the one using the piecewise linear technique with “near allocation”, which achieves an execution time of 122 clock cycles, but requires a total memory size of about 2.5 KB. Implementations using FastRTS Library functions, with single- or double-precision arithmetic, are the slowest, with execution times of 280 and 445 clock cycles, respectively.

**Table 5.** Full elaboration algorithm implementation results (C67xx DSP—ideal memory model). The columns related to total memory size and execution time are highlighted in grey.

Sin/Cos Computation Technique	LUT Storage Tech.	Memory Size (Bytes)					Execution Time (Clock Cycles)					
		DDFS LUT	FIR LUT	DDFS Code	FIR Code	Code Other	Total	DDFS	FIR Sum-of-Prod.	FIR Buffer Update	Other	Total
Piecewise Lineax Approx.	far alloc.	1536	84	224	64	640	2548	25.0	28.0	46.0	29.0	128.0
Piecewise Lineax Approx.	near alloc.	1536	84	224	64	640	2548	19.0	28.0	46.0	29.0	122.0
Piecewise Quadratic Approx.	near alloc.	128	84	256	64	640	1172	21.0	28.0	46.0	29.0	124.0
Library functions—Single-Precision	-	-	84	928	64	640	1716	177.0	28.0	46.0	29.0	280.0
Library functions—Double-Precision	-	-	84	1120	64	640	1908	342.0	28.0	46.0	29.0	445.0

To test the developed algorithm, we used a DSK6713 rapid prototyping board, shown in Figure 7. The board is equipped with a Texas Instruments C6713 DSP, which uses a 225 MHz internal core clock frequency and provides a 256 KB internal memory. The board also includes a 16 MB external SDRAM, 512 KB flash memory, and a TLV320AIC23B (AIC23) Audio CODEC. In our experiments, the analog audio input is provided to the Codec using the “line in” 3.5 mm input jack, and the elaborated audio output is obtained using either the “line out” or the “headphone” 3.5 mm output jacks.



**Figure 7.** DSK6713 rapid prototyping board used to test developed algorithm.

The minimum sampling frequency that can be configured in the Codec is 8 kHz; however, in the considered application, a sampling frequency of 2 kHz is more than adequate to sample cardiac audio signal (in fact  $f_s = 2$  kHz is the design value considered in the previous section). Therefore, in the realized tests, the Codec is configured for an 8 kHz



sampling frequency, and digital down and up sampling is implemented on the input and output signals, respectively, in order to keep a sampling frequency of 2 kHz for the core elaboration algorithm (which implement the frequency shift operation). Anti-aliasing and reconstruction filtering of digital down and up conversions is realized using a 39th-order FIR filter.

To realize the experiments, the code has been implemented using Code Composer Studio developing environment of Texas Instruments. The communication between the DSP and the AIC23 Codec of the board is realized using the McBSP interfaces of the DSP and employing a block processing approach with a block size of four samples. The EDMA unit of the DSP is used to transfer the blocks of data between the McBSP interfaces and the internal memory of the DSP, using a ping-pong buffering approach. The block size (equal to 4) has been chosen in order to match the ratio between the Codec sampling frequency (8 kHz) and core elaboration sampling frequency (2 kHz): this results in keeping a stream processing elaboration approach for core elaboration. Since the DSK6713 has on-board JTAG Emulation, the board is connected to the PC running Code Composer Studio by simply using a USB cable. Code Composer Studio is used for debugging and profiling. Realtime audio input and output are achieved using the 3.5 mm jacks of the board.

The results of the profiling achieved using the DSK6713 board are reported in Table 6. If we compare the execution time of the core elaboration with the data obtained in Table 5, we can observe that the execution time obtained by the profiling is very close to the value obtained by the simulator in the case of piecewise DDFS implementations. In this case, in fact, we have verified that the number of cache misses results is very low. A larger number of cache misses was observed in the implementations using single- and double-precision library functions, specifically considering program memory, and this corresponds to an evident increase in the execution time, with respect to the results of Table 5. This behavior can be explained by considering the code size reported in Table 7, and observing that in the case of implementations using single- and double-precision library functions, the code size of frequency shift elaboration (1632 and 1824 bytes, respectively) represents a consistent portion of the DSP Program Cache memory (4 KB). In this case, moreover, the total code size results are also equal to or larger than the Program Cache memory size.

**Table 6.** Experimental profiling results obtained using the DSK6713 board. The column related to total execution time is highlighted in grey.

Sin/Cos Computation Technique	LUT Storage Tech.	Execution Time (Clock Cycles)			
		Core Elaboration (Frequency Shift)	Down/Up Sampling	ISRs	Total
Piecewise Lineax Approx.	far alloc.	128.2	221.0	63.0	412.2
Piecewise Lineax Approx.	near alloc.	122.2	215.0	62.3	399.5
Piecewise Quadratic Approx.	near alloc.	124.1	218.2	62.3	404.7
Library functions—Single-Precision Library	-	285.2	214.1	62.6	561.9
Library functions—Double-Precision	-	510.2	228.2	69.4	807.8

**Table 7.** Required memory size in DSK6713 board implementation. The columns related to total code size and data size are highlighted in grey.

Sin/Cos Computation Technique	LUT Storage Tech.	Code Size (Bytes)				Data Size (Bytes)			
		Core Elaboration (Frequency Shift)	Down/Up Sampling	ISRs	Total	Core Elaboration (Frequency Shift)	Down/Up Sampling	ISRs	Total
Piecewise Lineax Approx.	far alloc.	928	1280	1184	3392	1620	280	70	1970
Piecewise Lineax Approx.	near alloc.	928	1280	1184	3392	1620	280	70	1970
Piecewise Quadratic Approx. Library	near alloc.	960	1280	1184	3424	212	280	70	562
Library functions—Single-Precision Library	-	1632	1280	1184	4096	84	280	70	434
Library functions—Double-Precision	-	1824	1280	1184	4288	84	280	70	434

Please note that in this application the total execution times reported in Table 6 results are much lower than the available time for real-time elaboration, considering the DSP clock frequency (225 MHz) and sampling frequency (2 kHz) is about  $225 \text{ MHz}/2 \text{ kHz} = 112,500$  clock cycles. This calls for a possible sensible reduction in DSP clock frequency that could be beneficial to reduce power dissipation. Since the DSP clock input frequency cannot be changed in the DSK6713 board, we performed this analysis using the DSP power model provided by Texas Instruments [74]. The power analysis results are reported in Table 8. In this table, the first column reports the minimum possible DSP core clock frequency which allows respecting the real-time constraint ( $C/f_{clock} = 1/f_s$ ,  $C$  being the total execution time in clock cycles reported in Table 6). The power analysis is performed by assuming that in each case, the DSP is clocked using the minimum possible clock frequency. To conduct an accurate power analysis, the DSP percentage core utilization has been evaluated using the profiler of the cycle-accurate simulator of C6713 DSP; in this context, the percentage core utilization has been obtained as the ratio between the average number of instructions executed per clock cycle (computed as total number of instructions executed divided by the total cycle count, value also reported in Table 8) and 8, which represents the number of functional units of the DSP (maximum number of instructions per execution packet). The obtained dynamic power dissipation is reported in the last three columns of the table. It can be observed that piecewise techniques allow achieving a remarkable power dissipation reduction with respect to single- and double-precision implementations, with a power saving that is close to 25% with respect to single precision and 44% with respect to double precision. The lowest power dissipation is obtained using piecewise linear implementation with near allocation of the LUT.

**Table 8.** Power Analysis results (DSP TI C6713,  $V_{DD} = 1.2 \text{ V}$ ). The column related to total dynamic power is highlighted in grey.

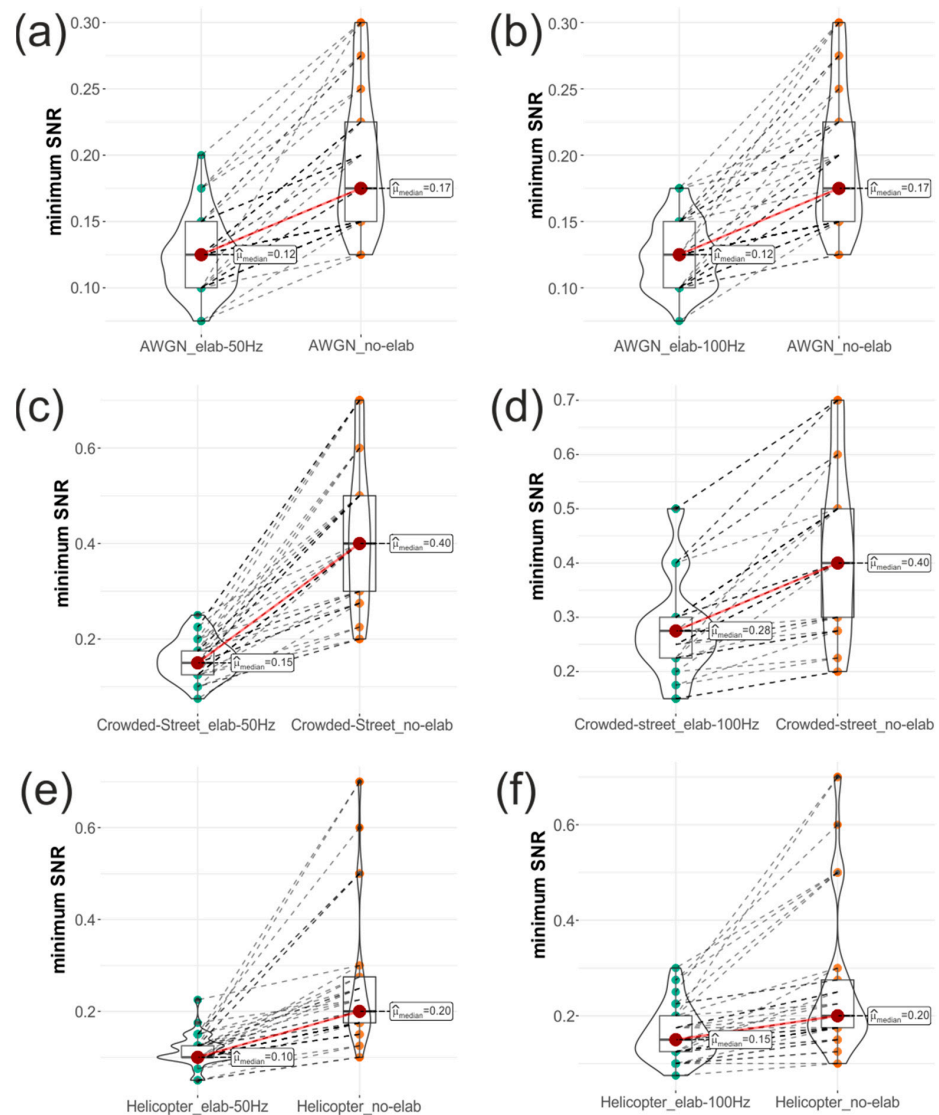
Sin/Cos Computation Technique	LUT Storage Tech.	Minimum DSP Clock Freq. (MHz)	Avg. Istr. Executed per Clock Cycle	DSP Core Utilization (%)	Power estimation		
					Dynamic Power (mW)		
					Activity Power	Clock Tree	Total Power
Piecewise Lineax Approx.	far alloc.	0.824	4.02	50.2%	0.58	1.76	2.35
Piecewise Lineax Approx.	near alloc.	0.799	4.09	51.1%	0.57	1.71	2.28
Piecewise Quadratic Approx.	near alloc.	0.809	4.11	51.4%	0.58	1.73	2.31
Library functions—Single-Precision	-	1.124	2.95	36.8%	0.61	2.40	3.01
Library functions—Double-Precision	-	1.616	2.20	27.5%	0.69	3.46	4.14

### 3.2. Auscultation Tests Results

Table 9 shows some descriptive statistics of the  $SNR_{min}$  values obtained for *no\_elab*, *elab\_50*, and *elab\_100* clips, for each noise source. For each experimental setting, Figure 8 displays the pairwise violin plots of the  $SNR_{min}$  values, by comparing the results obtained for processed and unprocessed audio clips. In particular, the panels of the left column compare the results related to *elab\_50* and *no\_elab* clips, while the panels of the right column compare the results related to *elab\_100* and *no\_elab* clips; the panels of each row show the results obtained for a specific noise source. The plots provide evidence that the distributions of the  $SNR_{min}$  values related to the signals elaborated with the proposed frequency shifter are different, always showing lower central location values and overall, less variability.

**Table 9.** Descriptive statistics of  $SNR_{min}$  values obtained for unprocessed (*no\_elab*) and processed (*elab\_50*, *elab\_100*) audio clips, for each noise source.

Noise Source	Processing	Min	1st Quartile	Median	Mean	3rd Quartile	Max	SD
AWGN	no_elab	0.125	0.150	0.175	0.195	0.225	0.300	0.05038
	elab_50	0.075	0.100	0.125	0.124	0.150	0.200	0.02766
	elab_100	0.075	0.100	0.125	0.124	0.150	0.175	0.02468
CROWDED STREET	no_elab	0.200	0.300	0.400	0.423	0.500	0.700	0.15236
	elab_50	0.075	0.125	0.150	0.158	0.175	0.250	0.04087
	elab_100	0.150	0.225	0.275	0.282	0.300	0.500	0.09087
HELICOPTER	no_elab	0.100	0.175	0.200	0.259	0.275	0.700	0.15550
	elab_50	0.050	0.100	0.100	0.114	0.125	0.225	0.03309
	elab_100	0.075	0.125	0.150	0.161	0.200	0.300	0.05392



**Figure 8.** Pairwise violin plots of the minimum SNR ( $SNR_{min}$ ) values that allowed heart sounds perception: (a) AWGN elaborated signal ( $f_{shift} = 50$  Hz) vs. AWGN not elaborated signal; (b) AWGN elaborated signal ( $f_{shift} = 100$  Hz) vs. AWGN not elaborated signal; (c) Crowded street elaborated signal ( $f_{shift} = 50$  Hz) vs. Crowded street not elaborated signal; (d) Crowded street elaborated signal ( $f_{shift} = 100$  Hz) vs. Crowded street not elaborated signal; (e) Helicopter elaborated signal ( $f_{shift} = 50$  Hz) vs. Helicopter not elaborated signal; (f) Helicopter elaborated signal ( $f_{shift} = 100$  Hz) vs. Helicopter not elaborated signal.

The results of the Wilcoxon Signed-Rank tests are outlined in Table 10. As it can be observed, all tests show very small  $p$ -values ( $p < 0.001$ ) supporting the conclusion that there is large statistical evidence against the null hypothesis that there is no difference between the median  $SNR_{min}$  of the elaborated and not elaborated signals in favor of the alternative that the  $SNR_{min}$  is higher in the case of the not elaborated signals. As higher  $SNR_{min}$  values indicate worse audibility, the tests confirm that the elaboration algorithm reduces the  $SNR_{min}$ , thus improving the audibility of heart sounds. In addition, the effect size has been evaluated by means of the matched pairs rank biserial correlation coefficient ( $r_{rb}$ ). It always reached a value of 1, thus highlighting the high practical relevance of the research outcomes. The analyses confirm the usefulness of the proposed method, which proved capable of enhancing the audibility of heart sounds in the presence of noise.

**Table 10.** Results of the Wilcoxon Signed-Rank Test performed on matched pairs of  $SNR_{min}$  values related to processed (*elab\_50* or *elab\_100*) vs. unprocessed (*no\_elab*) audio clips. The  $z$ -statistic used for computing approximate  $p$ -value ( $z$ -val) and the matched pairs rank biserial correlation coefficient ( $r_{rb}$ ) are also reported.

Noise Source	Comparison	W	z-val	p-Value	$r_{rb}$	# Pairs
AWGN	elab_50 Hz vs. no_elab	861	5.61	$1.005 \times 10^{-8}$	1.00	41
	elab_100 Hz vs. no_elab	861	5.60	$1.057 \times 10^{-8}$	1.00	41
CROWDED STREET	elab_50 Hz vs. no_elab	861	5.58	$1.212 \times 10^{-8}$	1.00	41
	elab_100 Hz vs. no_elab	861	5.59	$1.107 \times 10^{-8}$	1.00	41
HELICOPTER	elab_50 Hz vs. no_elab	861	5.60	$1.066 \times 10^{-8}$	1.00	41
	elab_100 Hz vs. no_elab	780	5.47	$2.210 \times 10^{-8}$	1.00	39

#### 4. Discussion

This study presented a novel algorithm to enhance the acoustic perception of heart sounds in noisy environments in real time. The proposed approach shifts the frequency spectrum of heart sounds to slightly higher frequencies (from tens of Hz up to 100 Hz) in order to achieve two conditions at the same time: moving the heart sounds spectrum in a frequency range corresponding to higher human audibility, and also moving such spectrum away from environmental low-frequency noises (e.g., vehicles engines, crowded streets), which are the ones that most affect the perception of heart sounds. The frequency shift is performed via a single-sideband modulation based on the Hilbert transform. The proposed algorithm was implemented on a VLIW DSP platform, which allowed real-time operation. At the core of the DSP implementation is a novel piecewise quadratic approximation approach for the implementation of the DDFS. This approach was compared with a piecewise linear approximation method and with library functions, in terms of required memory size, execution time, and power dissipation.

Both piecewise approximations provided a remarkable decrease of about 90% in the execution time of the DDFS as compared to library functions with single and double precision. The piecewise quadratic approximation also provided a remarkable reduction in required memory size, namely about 60% with respect to the library functions, and about 80% with respect to the piecewise linear approximation, which in fact, required higher memory resources than the library functions. Considering the whole algorithm implementation, the piecewise quadratic approximation ensured a 30% to 40% reduction in memory size, a 55% to 72% reduction in execution time, and a 25% to 44% reduction in power consumption, as compared to library functions. It also provided a more than 50% reduction in memory size, as compared to the piecewise linear approximation, as well as similar execution time and power consumption. According to these results, the proposed novel piecewise quadratic approximation for the DDFS provided the best tradeoff between memory size, execution time, and power consumption.

The actual performance of the proposed algorithm in enhancing the acoustic perception of heart sounds for a human operator in noisy environments was assessed by carrying out specific auscultation tests on 41 volunteers. Several audio clips consisting of heart

sound recordings (either unprocessed or processed via the proposed frequency shifter) corrupted with different kinds of noise, at different signal-to-noise ratios, were administered to the volunteers. In the case of processed signals, two shift frequencies, namely 50 Hz and 100 Hz, were considered. The volunteers were asked to acknowledge their ability to reliably perceive the heart sounds, in order to determine the minimum SNR values for reliable audibility. The minimum SNR values annotated for all the volunteers were analyzed and Wilcoxon Signed-Rank tests were performed to verify that the minimum SNR values obtained for the audio clips processed with the proposed algorithm were statistically lower than the minimum SNR values obtained for the unprocessed audio clips. The tests confirmed that the proposed algorithm, both with 50 Hz and 100 Hz frequency shifts, enhanced the audibility of heart sounds, as it allowed the volunteers to reliably perceive the heart sounds for lower SNR values, i.e., in the presence of higher noise levels.

This study has some limitations, essentially regarding the cohort of volunteers. Indeed, the enrollment did not account for the age of the volunteers, which has some relationship with the general hearing ability of a subject and may have an impact on the actual enhancement that the proposed method could provide. Future studies will focus on additional experiments on a larger cohort of subjects, with a good representation of sex, age, and other characteristics that may impact the hearing ability of a human subject. Further tests are also envisioned to be performed on a large number of cardiologists skilled in cardiac auscultation, in order to evaluate the ability of the proposed method to maintain essential diagnostic features of heart sounds.

## 5. Conclusions

The frequency shift of heart sounds via single-sideband modulation based on the Hilbert transform enhances the acoustic perception of heart sounds corrupted by different levels of AWGN, crowded avenues, and helicopter noises, according to the results of auscultation tests carried out on 41 volunteers. The algorithm was implemented in real time on a Very-Long-Instruction-Word Digital Signal Processor with hardware floating-point operations support. A novel fixed-point piecewise quadratic approximation approach was proposed for the implementation of the Direct Digital Frequency Synthesizer at the core of the single-sideband modulator, demonstrating to provide the best tradeoff between required memory size, execution time, and power consumption, as compared to a fixed-point piecewise linear approximation, and optimized floating-point library functions provided with the DSP. The proposed frequency shifter could be equipped in modern electronic stethoscopes at very modest costs, and could provide valuable support in all those situations where heart sounds auscultation is fundamental to assess the health status of a patient, and has to be performed in noisy environments, as an example, in emergency medicine contexts such as roadside or helicopter rescue.

**Author Contributions:** Conceptualization, E.A., P.B. and D.D.C.; methodology, V.M., E.A., C.C., P.B. and D.D.C.; software, D.D.C.; validation, V.M., E.A., C.C. and D.D.C.; formal analysis, E.A., C.C. and D.D.C.; investigation, V.M., E.A., C.C., J.C., G.D.M., D.E., P.B. and D.D.C.; data curation, V.M. and D.D.C.; writing—original draft preparation, V.M., E.A., C.C. and D.D.C.; writing—review and editing, V.M., E.A., C.C., J.C., G.D.M., D.E., P.B. and D.D.C.; visualization, C.C. and D.D.C.; supervision, E.A., P.B. and D.D.C. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** All relevant research data will be made available upon request after the publication of the paper.

**Conflicts of Interest:** The authors declare no conflict of interest.



## References

1. Thomas, S.L.; Heaton, J.; Makaryus, A.N. Physiology, Cardiovascular Murmurs. In *StatPearls*; StatPearls Publishing: Treasure Island, FL, USA, 2022.
2. Prakash, R.; Moorthy, K.; Aronow, W.S. First heart sound: A phono-echocardiographic correlation with mitral, tricuspid, and aortic valvular events. *Catheter. Cardiovasc. Diagn.* **1976**, *2*, 381–387. [[CrossRef](#)]
3. Mehta, N.J.; Khan, I.A. Third heart sound: Genesis and clinical importance. *Int. J. Cardiol.* **2004**, *97*, 183–186. [[CrossRef](#)]
4. Pelech, A.N. The physiology of cardiac auscultation. *Pediatr. Clin. N. Am.* **2004**, *51*, 1515–1535. [[CrossRef](#)] [[PubMed](#)]
5. Rushmer, R.F. *Cardiovascular Dynamics*, 4th ed.; W.B. Saunders: Philadelphia, PA, USA, 1976; ISBN 13:9780721678474.
6. Leatham, A. *Auscultation of the Heart and Phonocardiography*; Churchill: Livingstone, Zambia, 1970; ISBN -10 070001490X. ISBN -13 9780700014903.
7. Nowak, L.J.; Nowak, K.M. Sound differences between electronic and acoustic stethoscopes. *Biomed. Eng. Online* **2018**, *17*, 104. [[CrossRef](#)] [[PubMed](#)]
8. Weiss, D.; Erie, C.; Iii, J.B.; Copt, R.; Yeaw, G.; Harpster, M.; Hughes, J.; Salem, D.N. An in vitro acoustic analysis and comparison of popular stethoscopes. *Med. Devices Evid. Res.* **2019**, *12*, 41–52. [[CrossRef](#)] [[PubMed](#)]
9. Silverman, B.; Balk, M. Digital Stethoscope—Improved Auscultation at the Bedside. *Am. J. Cardiol.* **2019**, *123*, 984–985. [[CrossRef](#)] [[PubMed](#)]
10. Kalinauskienė, E.; Razvadauskas, H.; Morse, D.J.; Maxey, G.E.; Naudžiūnas, A. A Comparison of Electronic and Traditional Stethoscopes in the Heart Auscultation of Obese Patients. *Medicina* **2019**, *55*, 94. [[CrossRef](#)]
11. Mohamed, N.; Kim, H.-S.; Kang, K.-M.; Mohamed, M.; Kim, S.-H.; Kim, J.G. Heart and Lung Sound Measurement Using an Esophageal Stethoscope with Adaptive Noise Cancellation. *Sensors* **2021**, *21*, 6757. [[CrossRef](#)]
12. WHO. *World Health Statistics 2021: Monitoring Health for the SDGs, Sustainable Development Goals*; WHO: Geneva, Switzerland, 2021.
13. Andreozzi, E.; Gargiulo, G.D.; Esposito, D.; Bifulco, P. A Novel Broadband Forcecardiography Sensor for Simultaneous Monitoring of Respiration, Infrasonic Cardiac Vibrations and Heart Sounds. *Front. Physiol.* **2021**, *12*, 725716. [[CrossRef](#)]
14. Ha, T.; Tran, J.; Liu, S.; Jang, H.; Jeong, H.; Mitbender, R.; Huh, H.; Qiu, Y.; Duong, J.; Wang, R.L.; et al. A Chest-Laminated Ultrathin and Stretchable E-Tattoo for the Measurement of Electrocardiogram, Seismocardiogram, and Cardiac Time Intervals. *Adv. Sci.* **2019**, *6*, 1900290. [[CrossRef](#)] [[PubMed](#)]
15. Liu, Y.; Norton, J.J.S.; Qazi, R.; Zou, Z.; Ammann, K.R.; Liu, H.; Yan, L.; Tran, P.L.; Jang, K.-I.; Lee, J.W.; et al. Epidermal mechano-acoustic sensing electronics for cardiovascular diagnostics and human-machine interfaces. *Sci. Adv.* **2016**, *2*, e1601185. [[CrossRef](#)]
16. Gupta, P.; Moghimi, M.J.; Jeong, Y.; Gupta, D.; Inan, O.T.; Ayazi, F. Precision wearable accelerometer contact microphones for longitudinal monitoring of mechano-acoustic cardiopulmonary signals. *NPJ Digit. Med.* **2020**, *3*, 19. [[CrossRef](#)]
17. Andreozzi, E.; Gargiulo, G.D.; Fratini, A.; Esposito, D.; Bifulco, P. A Contactless Sensor for Pacemaker Pulse Detection: Design Hints and Performance Assessment. *Sensors* **2018**, *18*, 2715. [[CrossRef](#)]
18. Liu, F.; Wang, Y.T.; Wang, Y.X. Research and Implementation of Heart Sound Denoising. *Phys. Procedia* **2012**, *25*, 777–785. [[CrossRef](#)]
19. Vikhe, P.S.; Nehe, N.S.; Thool, V.R. Heart Sound Abnormality Detection Using Short Time Fourier Transform and Continuous Wavelet Transform. In Proceedings of the 2009 Second International Conference on Emerging Trends in Engineering & Technology, Nagpur, India, 16–18 December 2009; pp. 50–54.
20. Zia, M.K.; Griffel, B.; Semmlow, J.L. Robust detection of background noise in phonocardiograms. In Proceedings of the 2011 1st Middle East Conference on Biomedical Engineering, Sharjah, United Arab Emirates, 21–24 February 2011; pp. 130–133. [[CrossRef](#)]
21. Tang, H.; Li, T.; Park, Y.; Qiu, T. Separation of Heart Sound Signal from Noise in Joint Cycle Frequency–Time–Frequency Domains Based on Fuzzy Detection. *IEEE Trans. Biomed. Eng.* **2010**, *57*, 2438–2447. [[CrossRef](#)] [[PubMed](#)]
22. Gradolewski, D.; Redlarski, G. Wavelet-based denoising method for real phonocardiography signal recorded by mobile devices in noisy environment. *Comput. Biol. Med.* **2014**, *52*, 119–129. [[CrossRef](#)] [[PubMed](#)]
23. Yoganathan, A.P.; Gupta, R.; Udwadia, F.E.; Miller, J.W.; Corcoran, W.H.; Sarma, R.; Johnson, J.L.; Bing, R.J. Use of the fast fourier transform for frequency analysis of the first heart sound in normal man. *Med. Biol. Eng. Comput.* **1976**, *14*, 69–73. [[CrossRef](#)]
24. Debbal, S.; Bereksi-Reguig, F. Computerized heart sounds analysis. *Comput. Biol. Med.* **2008**, *38*, 263–280. [[CrossRef](#)]
25. Naseri, H.; Homaeinezhad, M.; Pourkhajeh, H. Noise/spike detection in phonocardiogram signal as a cyclic random process with non-stationary period interval. *Comput. Biol. Med.* **2013**, *43*, 1205–1213. [[CrossRef](#)] [[PubMed](#)]
26. Rouis, M.; Sbaa, S.; Benhassine, N.E. The effectiveness of the choice of criteria on the stationary and non-stationary noise removal in the phonocardiogram (PCG) signal using discrete wavelet transform. *Biomed. Eng./Biomed. Tech.* **2020**, *65*, 353–366. [[CrossRef](#)]
27. Wang, F.; Ji, Z. Application of the Dual-tree Complex Wavelet Transform in Biomedical Signal Denoising. *Bio-Medical Mater. Eng.* **2014**, *24*, 109–115. [[CrossRef](#)] [[PubMed](#)]
28. Ali, M.N.; El-Dahshan, E.-S.A.; Yahia, A.H. Denoising of Heart Sound Signals Using Discrete Wavelet Transform. *Circuits Syst. Signal Process.* **2017**, *36*, 4482–4497. [[CrossRef](#)]
29. Singh, B.N.; Tiwari, A.K. Optimal selection of wavelet basis function applied to ECG signal denoising. *Digit. Signal Process.* **2006**, *16*, 275–287. [[CrossRef](#)]
30. Debbal, S.M.; Bereksi-Reguig, F. Time-frequency analysis of the first and the second heart beat sounds. *Appl. Math. Comput.* **2007**, *184*, 1041–1052.



31. Gradolewski, D.; Redlarski, G. The use of wavelet analysis to denoising of electrocardiography signal. In Proceedings of the XV International PhD Workshop, Wisła, Poland, 19–22 October 2013; pp. 456–461.
32. Debbal, S.; Bereksi-Reguig, F. Analysis of the second heart sound using continuous wavelet transform. *J. Med. Eng. Technol.* **2009**, *28*, 151–156. [[CrossRef](#)]
33. Messer, S.R.; Agzarian, J.; Abbott, D. Optimal wavelet denoising for phonocardiograms. *Microelectron. J.* **2001**, *32*, 931–941. [[CrossRef](#)]
34. Jain, P.K.; Tiwari, A.K. An adaptive thresholding method for the wavelet based denoising of phonocardiogram signal. *Biomed. Signal Process. Control.* **2017**, *38*, 388–399. [[CrossRef](#)]
35. Zhou, K.L.; Liu, Y.Y. Heart Sound Denoising of New Threshold Wavelet Transform. *Comput. Eng. Des.* **2020**, *41*, 2476–2481.
36. Li, S.; Li, F.; Tang, S.; Xiong, W. A Review of Computer-Aided Heart Sound Detection Techniques. *BioMed Res. Int.* **2020**, *2020*, 1–10. [[CrossRef](#)]
37. Xie-Feng, C.; Bin, J.; He, Y.; YuFeng, G.; ShaoBai, Z. A new method of heart sound signal analysis based on independent function element. *AIP Adv.* **2014**, *4*, 097131. [[CrossRef](#)]
38. Zhao, X.; Ye, B. Convolution wavelet packet transform and its applications to signal processing. *Digit. Signal Process.* **2010**, *20*, 1352–1364. [[CrossRef](#)]
39. Safara, F.; Doraisamy, S.; Azman, A.; Jantan, A.; Abdullah Ramaiah, A.R. Multi-level basis selection of wavelet packet decomposition tree for heart sound classification. *Comput. Biol. Med.* **2013**, *43*, 1407–1408. [[CrossRef](#)] [[PubMed](#)]
40. Huang, N.E.; Shen, Z.; Long, S.R.; Wu, M.C.; Shih, H.H.; Zheng, Q.; Yen, N.-C.; Tung, C.C.; Liu, H.H. The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. *Proc. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci.* **1998**, *454*, 903–995. [[CrossRef](#)]
41. Salman, A.H.; Ahmadi, N.; Mengko, R.; Langi, A.Z.R.; Mengko, T.L.R. Empirical Mode Decomposition (EMD) Based Denoising Method for Heart Sound Signal and Its Performance Analysis. *Int. J. Electr. Comput. Eng.* **2016**, *6*, 2197–2204.
42. Gao, Y.; Ge, G.; Sheng, Z.; Sang, E. Analysis and Solution to the Mode Mixing Phenomenon in EMD. In Proceedings of the 2008 Congress on Image and Signal Processing, Sanya, China, 27–30 May 2008; pp. 223–227. [[CrossRef](#)]
43. Gaci, S. A New Ensemble Empirical Mode Decomposition (EEMD) Denoising Method for Seismic Signals. *Energy Procedia* **2016**, *97*, 84–91. [[CrossRef](#)]
44. Wu, Z.H.; Huang, N.E. Ensemble empirical mode decomposition: A noise-assisted data analysis Method. *AADA Adv. Adapt. Data Anal.* **2009**, *1*, 1–4. [[CrossRef](#)]
45. Yeh, J.-R.; Shieh, J.-S.; Huang, N.E. Complementary ensemble empirical mode decomposition: A novel noise enhanced data analysis method. *Adv. Adapt. Data Anal.* **2010**, *2*, 135–156. [[CrossRef](#)]
46. Dong, L.C.; Guo, X.M.; Zheng, Y.N. Wavelet Packet De-Noising Algorithm for Heart Sound Signals Based on CEEMD. *J. Vib. Shock.* **2019**, *38*, 192–198+222.
47. Dragomiretskiy, K.; Zosso, D. Variational Mode Decomposition. *IEEE Trans. Signal Process.* **2014**, *62*, 531–544. [[CrossRef](#)]
48. Banerjee, S.; Mishra, M.; Mukherjee, A. Segmentation and detection of first and second heart sounds (S1 and S2) using variational mode decomposition. In Proceedings of the 2016 IEEE EMBS Conference on Biomedical Engineering and Sciences (IECBES), Kuala Lumpur, Malaysia, 4–8 December 2016; pp. 565–570. [[CrossRef](#)]
49. Liu, Q.; Xu, Y.; Zhang, L.; Liang, C. Research on Heart Sound Signal Denoising Algorithm Based on Variational Mode Decomposition and Wavelet Threshold. *J. Comput. Commun.* **2021**, *9*, 110–121. [[CrossRef](#)]
50. Rudin, L.I.; Osher, S.; Fatemi, E. Nonlinear total variation based noise removal algorithms. *Phys. D Nonlinear Phenom.* **1992**, *60*, 259–268. [[CrossRef](#)]
51. Varghees, V.N.; Ramachandran, K. A novel heart sound activity detection framework for automated heart sound analysis. *Biomed. Signal Process. Control.* **2014**, *13*, 174–188. [[CrossRef](#)]
52. Haykin, S. *Unsupervised Adaptive Filtering: Volume I Blind Source Separation*; Wiley: New York, NY, USA, 2000.
53. Zheng, H.; Wang, H.; Wang, L.Y.; Yin, G.G. Cyclic system reconfiguration and time-split signal separation with applications to lung sound pattern analysis. *IEEE Trans. Signal Process.* **2007**, *55*, 2897–2913. [[CrossRef](#)]
54. Shah, G.; Koch, P.; Papadias, C.B. On the Blind Recovery of Cardiac and Respiratory Sounds. *IEEE J. Biomed. Health Inform.* **2014**, *19*, 151–157. [[CrossRef](#)]
55. Emmanouilidou, D.; McCollum, E.D.; Park, D.E.; Elhilali, M. Adaptive Noise Suppression of Pediatric Lung Auscultations with Real Applications to Noisy Clinical Settings in Developing Countries. *IEEE Trans. Biomed. Eng.* **2015**, *62*, 2279–2288. [[CrossRef](#)]
56. Khan, A.K.; Onoue, T.; Hashiodani, K.; Fukumizu, Y.; Yamauchi, H. Signal and noise separation in medical diagnostic system based on independent component analysis. In Proceedings of the 2010 IEEE Asia Pacific Conference on Circuits and Systems, Kuala Lumpur, Malaysia, 6–9 December 2010; pp. 812–815. [[CrossRef](#)]
57. Hunt, R.C.; Bryan, D.M.; Brinkley, V.S.; Whitley, T.W.; Benson, N.H. Inability to assess breath sounds during air medical transport by helicopter. *JAMA* **1991**, *265*, 1982–1984. [[CrossRef](#)]
58. Nelson, G.; Rajamani, R.; Erdman, A. Noise control challenges for auscultation on medical evacuation helicopters. *Appl. Acoust.* **2014**, *80*, 68–78. [[CrossRef](#)]
59. Tourtier, J.P.; Fontaine, E.; Coste, S.; Ramsang, S.; Schiano, P.; Viaggi, M.; Libert, N.; Durand, X.; Chargari, C.; Borne, M. In flight auscultation: Comparison of electronic and conventional stethoscopes. *Am. J. Emerg. Med.* **2011**, *29*, 932–935. [[CrossRef](#)] [[PubMed](#)]

60. Fontaine, E.; Coste, S.; Poyat, C.; Klein, C.; Lefort, H.; Leclerc, T.; Dubourdieu, S.; Briche, F.; Jost, D.; Maurin, O.; et al. In-flight auscultation during medical air evacuation: Comparison between traditional and amplified stethoscopes. *Air Med. J.* **2014**, *33*, 283–285. [[CrossRef](#)] [[PubMed](#)]
61. Brown, L.H.; E Gough, J.; Bryan-Berg, D.M.; Hunt, R.C. Assessment of breath sounds during ambulance transport. *Ann. Emerg. Med.* **1997**, *29*, 228–231. [[CrossRef](#)]
62. Zun, L.S.; Downey, L. The Effect of Noise in the Emergency Department. *Acad. Emerg. Med.* **2005**, *12*, 663–666. [[CrossRef](#)] [[PubMed](#)]
63. Mallinson, T. Prehospital cardiac auscultation: Friend or foe? *J. Paramed. Pract.* **2010**, *2*, 256–259. [[CrossRef](#)]
64. McLane, I.; Emmanouilidou, D.; E West, J.; Elhilali, M. Design and Comparative Performance of a Robust Lung Auscultation System for Noisy Clinical Settings. *IEEE J. Biomed. Health Inform.* **2021**, *25*, 2583–2594. [[CrossRef](#)] [[PubMed](#)]
65. Holloway, G.A., Jr.; Watkins, D. An electronic frequency shifting stethoscope for heart sounds. *J. Bioeng.* **1978**, *2*, 59–64. [[PubMed](#)]
66. Jung, D.K. Reinforcing Stethoscope Sound using Spectral Shift. *J. Sens. Sci. Technol.* **2021**, *30*, 47–50. [[CrossRef](#)]
67. Aumann, H.M.; Emanetoglu, N.W. Stethoscope with digital frequency translation for improved audibility. *Healthc. Technol. Lett.* **2019**, *6*, 143–146. [[CrossRef](#)]
68. Houtsma, A.J.M.; Curry, I.P.; Sewell, J.M.; Bernard, W.N. Auscultation in high-noise environments using hybrid electromechanical and ultrasound-Doppler techniques. *J. Acoust. Soc. Am.* **2006**, *120*, 3361. [[CrossRef](#)]
69. Gaydos, S. Clinical auscultation in noisy environments. *J. Emerg. Med.* **2012**, *43*, 492–493. [[CrossRef](#)]
70. Strollo, A.G.M.; De Caro, D. Direct Digital Frequency Synthesizers exploiting Piecewise Linear Chebyshev Approximation. *Microelectron. J.* **2003**, *34*, 1099–1106. [[CrossRef](#)]
71. Classification of Heart Sound Recordings: The PhysioNet/Computing in Cardiology Challenge. Available online: <https://physionet.org/content/challenge-2016/1.0.0/> (accessed on 10 February 2021).
72. Available online: <https://pixabay.com/it/sound-effects/crowded-avenue-people-talking-vendors-shouting-musicians-playing-part-1-7099/> (accessed on 15 July 2022).
73. Available online: <https://pixabay.com/it/sound-effects/helicopter-8030/> (accessed on 15 July 2022).
74. TMS320C6711D, C6712D, C6713B Power Consumption Summary; Texas Instruments Application Report SPRA889A. 2005.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.