FPGA-BASED IMPLEMENTATION OF THE INSTANTANEOUS FREQUENCY ESTIMATION OF PHONOCARDIOGRAPHIC SIGNALS

Andrés Hernández-Esteban [*] **Fco. Javier Toledo-Moreo** [*]
Juan Martínez-Alajarín [*] **J. Javier Martínez-Alvarez** [*]
Ramón Ruiz-Merino [*]

[*] *Dpto. Electrónica, Tecnología Computadoras y Proyectos*
Universidad Politécnica de Cartagena
30202 Cartagena, Murcia
Spain

Abstract: The instantaneous frequency can be used to provide information about how the frequency content of the phonocardiogram signal varies in time, in order to characterize the heart sounds and murmurs. The instantaneous frequency of a signal can be calculated from the discrete Hilbert transform, computed through the moving discrete Hartley transform, which reduces the computation time. To compute in real time the instantaneous frequency, the algorithms have been implemented in a FPGA device, exploiting the high performance and flexibility of reconfigurable hardware. The results obtained from the FPGA show high accuracy in comparison to those computed with Matlab®.

Keywords: Frequency signal analysis, signal processing algorithms, computer-aided system design, Hilbert transform, instantaneous frequency, FPGA, phonocardiographic signal

1. INTRODUCTION

The introduction of the Field-Programmable Gate Array (FPGA) devices about 15 years ago gave rise to the reconfigurable computing concept. First FPGAs generations were quite limited in their capacities. Nowadays, the most advanced manufacturing technologies are used to feature devices with millions of gates of programmable logic, with dedicated hardware resources, with the widest range of system connectivity solutions, enabling more complex and powerful systems on a single chip.

Reconfigurable hardware offers a tradeoff between the very specialized solution of Application Specific Integrated Circuits (ASICs) and the inefficiency of General Purpose Processors (GPPs), combining the desirable high performance of ASICs with the characteristics of system flexibility of GPPs. Like ASICs, it involves hardware implementation and consequently parallelism and high performance. Like GPPs, it provides reconfigurability, and hence flexibility and rapid prototyping. The key of reconfigurable hardware lies on that the flexibility is provided by the hardware design rather than by software-programmable hardware.

On the other side, the phonocardiographic signal or phonocardiogram (PCG) is the graphic representation of the heart sounds generated by the heart, and corresponds to what the clinical hears during auscultation. After some decades in which auscultation of the heart lost importance in favour of more accurate and expensive techniques, in the last few years, automated phonocardiogram analysis has been rediscovered as a useful and low cost technique to assess valvular diseases. It is well established that time domain only is not enough to represent the information content of the PCG, so joint time-frequency

techniques must be considered. Although traditionally other time-frequency techniques have been used (short time Fourier transform, Wigner-Ville distribution, wavelets, ...) (Obaidat 1993, Jing *et al.* 1995, Debiais *et al.* 1997, Tovar-Corona and Torry 1998), the instantaneous frequency (IF) has been also proposed as a useful technique to characterize the variation of the frequency content of the PCG in time in order to distinguish between several valvular diseases (Yoshida *et al.* 1997, Sharif *et al.* 2000).

In this paper a hardware implementation of the instantaneous frequency of the phonocardiographic signal is proposed, based on the definition given by Ville (Ville 1948) and using the discrete Hilbert transform, computed through the moving discrete Hartley transform (MDHT) (Prabhu and Sundaram 1996), to compute the IF in real time. The hardware design has been realized with a system level modeling tool for DSP (Digital Signal Processing).

This paper is organized as follows: in Section 2 an introduction to the Hilbert transform and the instantaneous frequency computation is presented. Section 3 describes the hardware implementation of the algorithms in the FPGA device, and Section 4 presents the obtained results, and compares them with those obtained in Matlab. Finally, some closing remarks are indicated in Section 5.

2. ANALYTIC SIGNAL AND INSTANTANEOUS FREQUENCY

2.1 The analytic signal

The analytic signal $z(t)$ of a real signal $x(t)$ is defined as:

$$z(t) = x(t) + j\,\mathcal{H}[x(t)] \tag{1}$$

where $\mathcal{H}[\cdot]$ is the Hilbert transform operator. For a discrete signal $x(n)$, the corresponding analytic signal is defined as:

$$z(n) = x(n) + j\,H[x(n)] \tag{2}$$

where $H[\cdot]$ is the discrete Hilbert transform (Kak 1970, Čížek 1970).

The discrete Hilbert transform can be computed also through the discrete Fourier transform. (Pei and Jaw 1989) proposed a fast algorithm to compute the Hilbert transform through the discrete Hartley transform (DHT), using only real-domain operations. (Prabhu and Sundaram 1996) proposed also another fast algorithm to compute the Hilbert transform through the moving discrete Hartley transform (MDHT), which reduced the computational complexity of (Pei and Jaw 1989). The algorithm presented in (Prabhu and Sundaram 1996) uses a recursive relation to compute the present value from previous ones.

Consider a signal $x(n)$ divided into overlapping blocks of N samples. The signal $x(n)$ is weighted by a window function $w(m)$, which is shifted L samples in time through $x(n)$. Thus, two consecutive blocks overlaps $N - L$ samples. If the input signal is multiplied by the window function, it results:

$$x(L \cdot n + m) \cdot w(m) \tag{3}$$

Thus, for each sample n from the previous windowed sequence, a new sequence of length V shifted L samples in time is obtained. If $w(m)$ is the rectangular function, defined as:

$$w(m) = \begin{cases} 1, & 0 \le m \le V - 1 \\ 0, & \text{elsewhere} \end{cases} \tag{4}$$

and $N = V$, the windowed sequence for each n is:

$$x(L \cdot n + m) \cdot w(m) = \begin{cases} x(L \cdot (n-1) + m + 1), \\ \qquad\qquad m = 0, 1, 2, \ldots, N-2 \\ x_{add}, \\ \qquad\qquad m = N - 1 \end{cases} \tag{5}$$

where x_{add} is the new sample.

The Hilbert transform computed through the MDHT, x', described in (Prabhu and Sundaram 1996), is computed as:

$$x'(L \cdot n, m) = \begin{cases} x'(L \cdot (n-1), m + 1), \\ \qquad\qquad m = 1, 3, \ldots, N-1 \\ x'(L \cdot (n-1), m + 1) + Y(L \cdot n) \cdot C(m), \\ \qquad\qquad m = 0, 2, \ldots, N-2 \end{cases} \tag{6}$$

which is the equation that has been implemented to compute the discrete Hilbert transform of $x(n)$, where

$$Y(L \cdot n) = x(L \cdot n + V) - x(L \cdot n) \tag{7}$$

and

$$C(m) = \frac{2}{N} \cdot \cot\left(\frac{\pi(m+1)}{N}\right) \tag{8}$$

For $N = V$ and $L = 1$, equation 7 transforms in

$$Y(n) = x(n + N) - x(n) \tag{9}$$

which is the same as

$$Y(n) = x_{add} - x(0) \tag{10}$$

from the windows sequence.

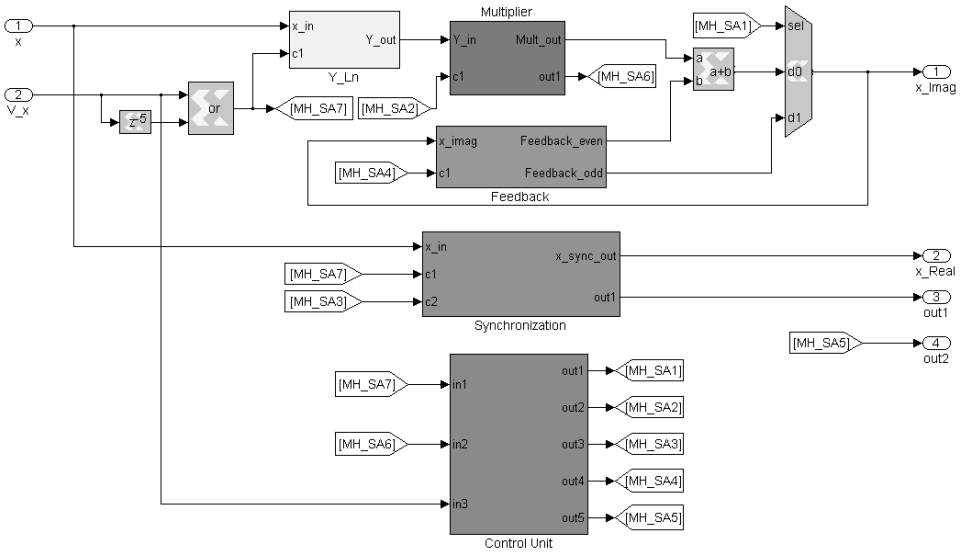

Fig. 1. Analytic signal module

2.2 The instantaneous frequency

The instantaneous frequency of a real signal $x(t)$ was defined by Ville (Ville 1948) as the derivative of the phase ($\phi(t)$) of the associated analytic signal $z(t)$:

$$\mathrm{IF}(t) = \frac{1}{2\pi}\frac{d}{dt}[\arg z(t)] = \frac{1}{2\pi}\frac{d\phi(t)}{dt} \qquad (11)$$

For a discrete time signal, the expression for the IF is similar to equation 11, but using discrete derivatives of the phase instead of $d\phi(t)/dt$, for example, the central finite difference (CFD) (Boashash 1992):

$$\mathrm{IF}(n) = \frac{1}{4\pi}(\phi(n+1) - \phi(n-1)) \qquad (12)$$

3. SYSTEM DESCRIPTION

The design has been described using Xilinx System Generator™ (Xilinx Inc. 2004). System Generator is a system level modeling tool for FPGA hardware design. It is a software platform that uses Simulink® to represent a high level description of a digital signal processing system, and extends it to constitute a useful environment for hardware design. Through different abstraction levels it is possible to implement efficient FPGA designs easily.

Moreover, Matlab has been the chosen platform to carry out the previous stages of study of the phonocardiographic signals, and the analysis and development of the different expressions that have led to the final hardware implementation. Therefore, the integration of the hardware design process within the overall research is facilitated by System Generator.

On the other hand, the use of the available records of phonocardiographic signals for design simulation is accomplished faster and more efficiently with System Generator, which also provides hardware co-simulation. This allows that the design running in the FPGA can be incorporated in a Simulink simulation.

The process can be divided into four steps:

- Conversion of the analog signal from the stethoscope to digital
- Computation of the analytic signal, using the discrete Hilbert transform
- Computation of the phase
- Computation of the instantaneous frequency, using the central finite difference.

The last three steps, which make up the digital processing, are described in the following subsections.

3.1 Computation of the analytic signal

As mentioned in subsection 2.1, the analytic signal consists of two parts: the real part, which is the same input signal, and the imaginary part, obtained using the Hilbert transform of the input signal. Figure 1 shows the design structure in System Generator. Next, the blocks in this figure are described:

- Y_Ln block. It computes the arithmetic function of equation 7. It is implemented with a simple hardware subtractor based on the Xilinx Smart-IP™ Adder/ Subtractor Core. The block latency depends on the window size (N): the bigger window size, the larger number of samples of the input signal must be acquired and the higher latency.
- Multiplier block. This block computes the term $C(m)$ of the equation 6 corresponding to the multiplication of equation 8. The values obtained for the cotangent term have been stored in a ROM memory, implemented with BlockRAM resources. These are specific purpose memory resources included in Xilinx Virtex II FPGA devices.

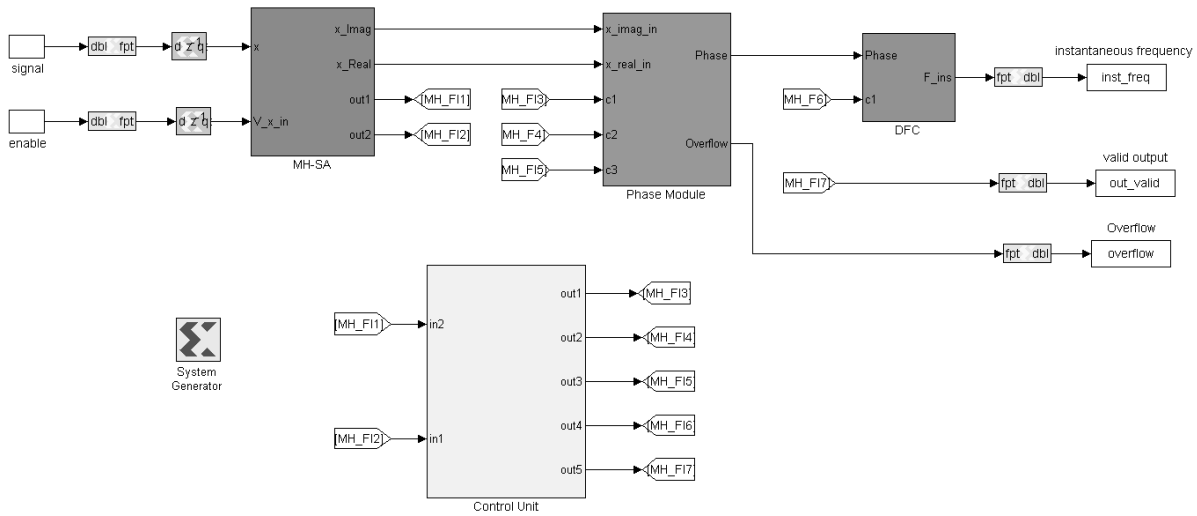

Fig. 2. Instantaneous frequency module

System Generator multiplier block has been configured to implement a parallel multiplier using the embedded multipliers in the Xilinx Virtex II FPGAs. Operating on the full width data with the multiplier blocks allows to achieve the fastest hardware multiplication compared to multiplication in logic.

- Feedback block. Since the discrete Hilbert transform computes a new value of the analytic signal from previously calculated values, a feedback block must be included. As equation 6 shows, it must be taken into account if the sample number is even or odd.
- Synchronization block. Since the imaginary part of the analytic signal required the described processing, but not the real part, a block is needed to ensure that both components are synchronized at the output. The hardware is based on the Xilinx Addressable Shift Register, which implements a delay line of configurable length. In this case the length is fixed to the window size.
- Control Unit. This block generates the control signals of the module.

3.2 Computation of the phase

This module calculates the analytic signal phase and its correction by means of the unwrap function. The phase is obtained using a CORDIC algorithm. The instanced block implements a rectangular to polar coordinate conversion using a fully parallel CORDIC algorithm in circular vectoring mode.

The unwrap function corrects phase angles by adding multiples of $\pm 2\pi$ where needed, to smooth the transitions and make the phase continuous. For hardware implementation, and taking the advantage that it is a combinational function, the function is defined in Matlab `.m` code, inside an MCode Block. When hardware is generated by System Generator this code is translated in a straightforward way to behavioural VHDL.

3.3 Computation of the instantaneous frequency

This blocks calculates the instantaneous frequency using the central finite difference, according to equation 12. Hardware just consists of delay elements, a subtractor and a multiplier by a constant implemented with logic.

4. RESULTS

The design has been functionally simulated in System Generator exactly the same as any model described in Simulink. To test the accuracy of the proposed system the instantaneous frequency has been computed in Matlab. These results have been used as reference for error estimation. The RMSE (Root Mean Square Error) between the simulation in Matlab and the functional simulation in Simulink, was 0.0038. The RMSE is defined as:

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{x}_i)^2} \qquad (13)$$

where x_i is the Matlab simulation signal, $\hat{x}_i$ is the functional simulation in ModelSim, and n is the number of samples.

Figure 3 shows the phonocardiographic signal, the computation of the instantaneous frequency with the System Generator model in Simulink and the error with respect to the results obtained in Matlab. The output and internal data are coded with 26 bits. The window size used is 128. Except for the central peak around sample number 9000, the two signals are virtually identical. The small error is mainly due to the

Table 1. Design and timing summaries of the final hardware implementation with a window size of 128 samples.

Design summary			Timing summary	
Logic utilization			Minimum period	27.019 ns
Number of 4 input Look Up Table	2030	19%	Maximum frequency	37 MHz
Number of Slice Flip Flops	2168	21%		
Logic Distribution				
Number of occupied Slices	1907	37%		
Total Number 4 input LUTs	3525	34%		
Number of bonded IOBs	49	28%		
Specific purpose resources				
Number of Block RAMs	2	5%		
Number of MULT18X18s	4	10%		
Number of GCLKs	1	6%		

change of the arithmetic data type, from double precision floating point in Matlab to signed fixed point in the System Generator model.

Functional simulation with an HDL simulator (like ModelSim) is not necessary, since System Generator provides bit-true results and, therefore, finite precision fixed-point operations are modeled precisely.

Once the functional simulation is successfully completed, the design should be exported to the Xilinx® ISE design environment, where it would be implemented. After mapping, placing and routing processes the information for timing simulation would be available and this could be run on, e. g., the ModelSim simulator.

However, it is possible to incorporate the design running in the FPGA into a Simulink simulation by means of the System Generator hardware co-simulation capability. In this situation, input data are generated in Simulink, processed in hardware and the results are reported to Simulink again. This allows the design to be tested in actual hardware. This also implies that simulation is dramatically sped up. The communication between the PC and the FPGA is established by a generic interface provided by System Generator that uses JTAG and the Xilinx Parallel Cable IV.

The hardware co-simulation results were practically identical to those obtained with Simulink. It took 15 seconds for a 2.2 seconds input signal (17450 samples at 8 kHz), while the corresponding timing simulation in the ModelSim simulator took more than 52 hours, both in a 3.00 GHz Pentium IV.

Finally, the design's bitstream file obtained in the final step of the implementation process is downloaded onto the FPGA. The implementation process has been completed in the Xilinx ISE 6.2.03i version. The targeted Xilinx XC2V1000-4FG256 FPGA is available in the Virtex II evaluation kit (Avnet Electronics Marketing 2004), a platform suitable for the development and test of Xilinx FPGA-based applications. This board is also populated with a Xilinx XC18V04VQ44C configuration EEPROM, where the configuration bitstream can be stored, an RS-232 serial port, several connectors (including AvBus connector) and general purpose inputs and outputs.

In the final prototype, the data are directly received from the stethoscope. Its analog output is converted to digital using the UCB1400 audio codec from Philips. This is a 20-bit stereo audio codec which integrates an AC97 2.1 interface and supports programmable sample rate and input/output gain control. The UCB1400 is populated in the Audio/Video Module from Avnet Inc., a non stand-alone module designed to supply data to a host FPGA-based module like the Virtex II evaluation kit, via the AvBus connector. The codec configuration is executed from the FPGA by means of a state machine described in VHDL.

5. CONCLUSIONS

In this paper a hardware implementation of the instantaneous frequency of the phonocardiogram using the discrete Hilbert transform has been presented. A system level modeling tool for DSP, System Generator, integrated with Simulink and provided by Xilinx, has been used to develop the system, resulting in a faster design cycle.

Co-simulation in the FPGA, simulation in Simulink and the timing simulation in ModelSim provide virtually identical results, and they are very similar to those computed with Matlab.

Future work is needed to improve the hardware design in terms of power consumption and area, in order to use this design in portable devices, integrating it with the rest of the diagnosis system. In the final system, the instantaneous frequency module will provide information content of the PCG with respect to time, which aids in the characterization of the cardiac sound events.

ACKNOWLEDGEMENT

This work has been supported by Fundación Séneca of Región de Murcia and Ministerio de Ciencia y

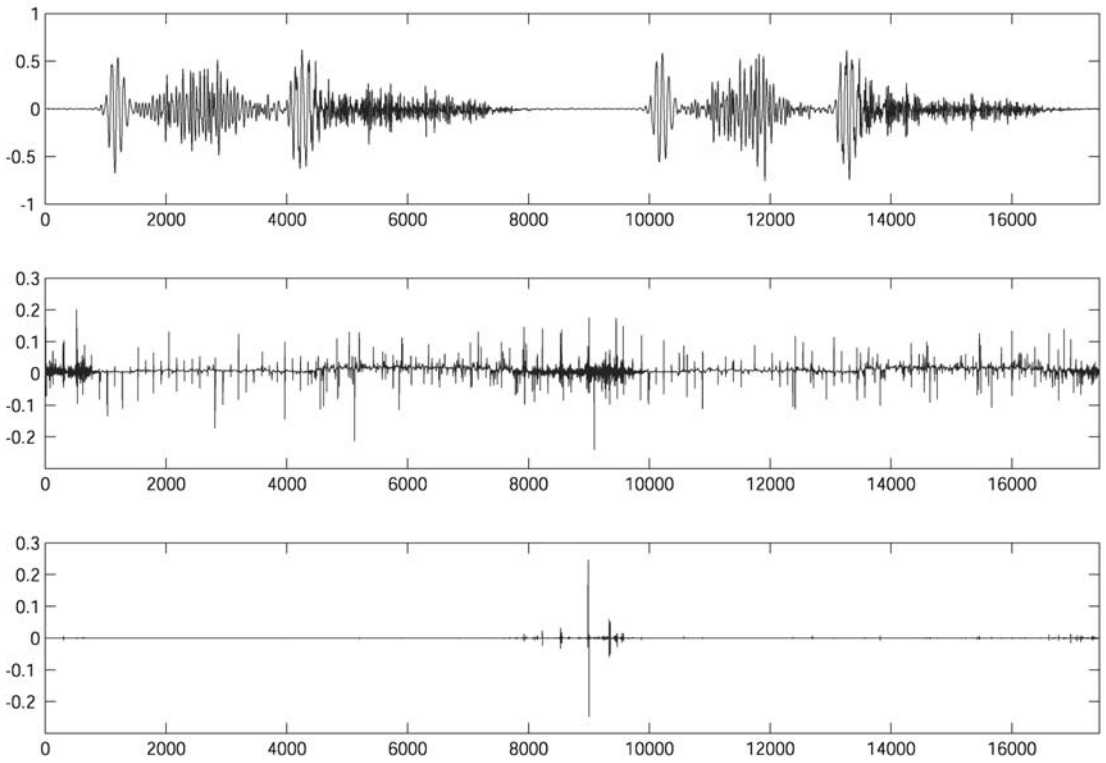

Fig. 3. Phonocardiographic signal (above), functional simulation in Simulink (middle), and error between the simulation in Matlab and in Simulink (below)

Tecnología of Spain, under grants PB/63/FS/02 and TIC2003-09400-C04-02, respectively.

REFERENCES

Avnet Electronics Marketing (2004). *Development and evaluation kits*. (http://www.em.avnet.com).

Boashash, B. (1992). Estimating and interpreting the instantaneous frequency of a signal—Part 2: Algorithms and applications. *Proceedings of the IEEE* **80**(4), 540–568.

Čížek, V. (1970). Discrete Hilbert transform. *IEEE Transactions on Audio and Electroacoustics* **AU-18**(4), 340–343.

Debiais, F. , L. G. Durand, Z. Guo and R. Guardo (1997). Time-frequency analysis of heart murmurs. Part ii: Optimisation of time-frequency representations and performance evaluation. *Medical & Biological Engineering & Computing* **35**, 480–485.

Jing, W. , W. Liangjun, W. Jue and C. Jingzhi (1995). Heart sounds analysis in joint time-frequency domain. In: *Proceedings of the First Regional Conference IEEE Engineering in Medicine and Biology Society, and 14th Conference of the Biomedical Engineering Society of India*. Vol. 2. pp. 19–20.

Kak, S. C. (1970). The discrete Hilbert transform. *Proceedings of the IEEE* **58**, 585–586.

Obaidat, M. S. (1993). Phonocardiogram signal analysis: techniques and performance comparison. *Journal of Medical Engineering & Technology* **17**(6), 221–227.

Pei, S.-C. and S.-B. Jaw (1989). Computation of discrete Hilbert transform through fast Hartley Transform. *IEEE Transactions on Circuits and Systems* **36**(9), 1251–1252.

Prabhu, K. M. M. and R. S. Sundaram (1996). Fast algorithm for pseudodiscrete Wigner-Ville distribution using moving discrete Hartley transform. *IEE Proceedings – Vision, Image and Signal Processing* **143**(6), 383–386.

Sharif, Z. , M. S. Zainal, A. Z. Sha'ameri and S. H. S. Salleh (2000). Analysis and classification of heart sounds and murmurs based on the instantaneous energy and frequency estimations. In: *Proceedings of the IEEE TENCON*. Vol. 2. pp. 130–134.

Tovar-Corona, B. and J. N. Torry (1998). Time-frequency representation of systolic murmurs using wavelets. In: *Computers in Cardiology*. pp. 601–604.

Ville, J. (1948). Theory et applications de la notion de signal analytique. *Cables Trans.* **2A**(1), 61–74.

Xilinx Inc. (2004). *Xilinx System Generator v6.2 User Guide*. (http://www.xilinx.com).

Yoshida, H. , H. Shino and K. Yana (1997). Instantaneous frequency analysis of systolic murmur for phonocardiogram. In: *Proceedings of the 19th International Conference, Engineering in Medicine and Biology Society*. Vol. 4. Chicago, IL., USA. pp. 1645–1647.