This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2780190, IEEE Access

# Inequality Indexes as Sparsity Measures Applied to Ventricular Ectopic Beats Detection and its Efficient Hardware Implementation

Hamza Baali, Xiaojun Zhai, Hamza Djelouat, Abbes Amira, *Senior Member, IEEE* and Faycal Bensaali, *Senior Member, IEEE*

***Abstract*— Meeting application requirements under a tight power budget is of a primary importance to enable connected health internet of things (IoT) applications. This paper considers using sparse representation and well-defined inequality indexes drawn from the theory of inequality to distinguish ventricular ectopic beats (VEBs) from non-VEBs. Our approach involves designing a separate dictionary for each arrhythmia class using a set of labelled training QRS complexes. Sparse representation, based on the designed dictionaries of each new test QRS complex is then calculated. Following this, its class is predicted using the winner-takes-all principle by selecting the class with the highest inequality index. Our experiments showed promising results ranging between 80% and 100% for the detection of VEBs considering the patient-specific approach, 80% using cross-validation and 70% on unseen data using independent sets for training and testing respectively. An efficient hardware implementation of the alternating direction method of multipliers (ADMM) algorithm is also presented. The results show that the proposed hardware implementation can classify a QRS complex in 69.3 *ms* that use only 0.934 W energy.**

***Index Terms*— Inequality Indexes; Dictionary Learning; ADMM; Arrhythmia; Classification; Connected Health; QRS**

## I. Introduction

Driven by demographic changes, especially the growing ageing population and the prevalence of chronic diseases, existing healthcare systems are shifting from hospital-centred to connected health models. Real-time health monitoring using wearable devices outside the clinical setting represents one of the cornerstones in these new models and has been attracting considerable attention from both academia and industry in recent years [1]. Typically, the sensed data is sent via a short-range low-power wireless communication protocol to a battery powered internet of things (IoT) edge device equipped with routable connectivity. Its automatic analysis is then performed either locally on an edge device located in the vicinity of the patient or in the cloud [2]. The former approach is preferred to avoid big data issues such as latency related to cloud computing [3].

Among the most crucial vital signs in wearable devices, electrocardiogram (ECG) is a non-invasive diagnostic and prognostic tool to evaluate the status of the heart [4]. It reflects abnormal cardiac activities in both electrical generation and conduction at different levels in the heart as deviations from the normal intervals and waveforms morphologies. The term arrhythmia is used to refer to these deviations. Over the years, a large amount of research has been dedicated to developing accurate algorithms for automated arrhythmia classification. In particular, different approaches for feature extraction have been reported in the literature.

Heuristic descriptors such as ECG morphology and heartbeat intervals are among the commonly used features [5-7]. Other methods include signal modelling techniques whereby the model parameters serve as features [8-11]. In addition, many statistical parameters have been considered as features including high order statistics and correlation and Shannon entropies [12-14].

Researchers have also examined the use of a large number of classifiers for arrhythmia classification. These include, linear discriminant analysis (LDA) [5-6], neural networks (NN) [15-16], self-organising maps [9], support vector machine (SVM) [17], active learning [18] and combination of different classifiers [19-20].

However, despite their good accuracy, most of the aforementioned approaches are more suitable for offline applications due to their high computational costs [21]. The need to reduce the power consumption and to extend the battery life time for long-term health monitoring is driving the development of a new class of power-aware signal processing and classification techniques [21-22].

Sparse representation (SR) has emerged in recent years as a powerful tool for efficient data processing. It attempts to exploit the compressibility of the true signal in a transform domain by solving a sparsity seeking optimisation ($\ell_1$-Regularised Least Squares) or a greedy algorithm. Classifiers based on SR have been attracting growing interest from the research community and have shown to provide state-of-the-art performances across many applications. Two basic formulations are usually adopted in the literature,

This paper was submitted for the review on September 9th, 2017. This research work was supported by National Priorities Research Program (NPRP) grant No. 9-114-2-055 from the Qatar National Research Fund (a member of Qatar Foundation).

Hamza Baali, Hamza Djelouat, A. Amira and F. Bensaali are with the College of Engineering, Qatar University, Qatar. (e-mail: hamza.baali@qu.edu.qa; hamza.djelouat@qu.edu.qa; abbes.amira@qu.edu.qa; f.bensaali@qu.edu.qa).

X. Zhai is with the Department of Electronics, Computing and Mathematics, University of Derby, Derby, DE22 1GB, U.K. (e-mail: x.zhai@derby.ac.uk).

reconstructive and discriminative approaches [23]. In the reconstructive approach, class specific dictionaries (specifying transforms) are learned using labelled data. Each testing signal is then assigned to a particular class either based on the best reconstruction error or based on the best actual/penalised cost function. The latter approach takes into account both the reconstruction error and the sparsity of the signal representation. Discriminative formulations use augmented objective functions with a discrimination term that enforces the intra-classes discrimination in the projection subspace. These approaches, however, come with the intensive computational complexity of the sparse representations at test time which is a major obstacle that hinders their applicability in large-scale problems, or in applications with limited power supply [24]. The phase of dictionary learning is usually performed offline, and energy consumption is not of primary concern. Real-time detection of events of interest, however, should be performed in real-time at minimum power.

In this paper, we propose the use of two inequality measures drawn from the theory of inequality, namely the Gini and the Pietra indexes as alternative measures of sparsity for effective and simple arrhythmia classification. These indexes are quasi-convex, their values are independent of the size of the data vector, they are scale-invariant, they are independent of the total energy of the signal, and they depend on the energy concentration in a small number of coefficients [25-27].

We are looking for simplicity in the sense that these indexes allow an online implementation as their implementation is based on accumulators. Furthermore, we use a decomposition technique that significantly reduces the processing time as well as the complexity of the $\ell_1$-regularised least squares (LS) optimisation problem based on the alternating direction methods of multipliers (ADMM) algorithm**.**

An efficient hardware implementation of the algorithm is also presented. Particularly, energy cost and real-time reconstruction on a parallel hardware implementation of the proposed approaches running on a Zynq System-on-Chip (SoC) device are investigated.

The paper is organised as follows: Section II (a) formulates the sparse dictionary learning and sparse representation problem. The K-SVD and the alternating direction method of multipliers (ADMM) are presented in Sections II (b) and II (c) respectively. A fast method of the ridge regression updates of the ADMM is presented in Section II (d**)**. The Lorenz curve and the inequality indexes are discussed in Section II (e) while the proposed classification algorithm is described in Section II (e). A hardware implementation of the ADMM algorithm is presented in section III. Classification results are presented in Section IV, in which we evaluate and discuss three classification scenarios. Section V concludes the paper and highlights some perspectives of future work.

## II. Methodology

### *a. Sparse Data Representation*

Let $\boldsymbol{x} \in \mathbb{R}^N$ be a vector of observations. A sparse representation of $\boldsymbol{x}$ consists of approximating it using a weighed sum of a few dictionary atoms $\boldsymbol{d}_i \in \boldsymbol{\mathcal{R}}^N$ such that:

$$\boldsymbol{x} = \boldsymbol{D}\boldsymbol{w}, \qquad (1)$$

where $\boldsymbol{D} = [\boldsymbol{d}_1, \boldsymbol{d}_2, \dots, \boldsymbol{d}_K] \in \mathbb{R}^{N\times K}$ is an over-complete dictionary and $\boldsymbol{w} \in \mathbb{R}^K$ is a sparse vector. The dictionary $\boldsymbol{D}$ can be either predetermined or learned from a given training set of the observed data $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_N]$.

The latter approach seeks to learn a dictionary $\boldsymbol{D}$ and the corresponding sparse representation $\boldsymbol{W} = [\boldsymbol{w}_1, \boldsymbol{w}_2, \dots, \boldsymbol{w}_K]$. This process can be expressed as an optimisation problem with respect to $\boldsymbol{D}, \boldsymbol{W}$ and the learning scalar parameter $\lambda > 0$ such that [28-29]:

$$\{\boldsymbol{W_{opt}}, \boldsymbol{D_{opt}}\} = \operatorname{argmin}_{\boldsymbol{W},\boldsymbol{D}} \ \frac{1}{2} \ \|\boldsymbol{X} - \boldsymbol{D}\boldsymbol{W}\|_2^2 + \lambda\|\boldsymbol{W}\|_p$$
$$p \in \{0,1\} \qquad (2)$$

A practical iterative approach often used to solve (2) involves: 1) Keep $\boldsymbol{D}$ fixed and solve for $\boldsymbol{W}$, and 2) Keep $\boldsymbol{W}$ fixed and find $\boldsymbol{D}$.

For the case $p = 1$ and for a fixed dictionary $\boldsymbol{D}$, the above optimisation leads to the convex $\boldsymbol{\ell}_1$ regularised least squares problem (3).

$$\boldsymbol{W_{opt}} = \operatorname{argmin}_{\boldsymbol{W}} \ \tfrac{1}{2} \ \|\boldsymbol{X} - \boldsymbol{D}\boldsymbol{W}\|_2^2 + \lambda\|\boldsymbol{W}\|_1 \qquad (3)$$

Iterative methods such as least angle regression (LARS), (ADMM), or linear programming can be used to find a solution of (3) [30-31].

For a given sparse representation $\boldsymbol{W}$, dictionary learning is generally performed using the K-SVD algorithm described below [27].

### *b. The K-SVD Algorithm*

In general, the K-SVD algorithm alternate between two steps. The first is considered as a generalisation of the famous K-means clustering algorithm where the input vector is represented as a linear combination of the dictionary atoms (codewords in the case of K-means). The second step updates sequentially the dictionary atoms and the sparse representations while in K-means only the dictionary is updated. The following algorithm results:

Data: $\{\boldsymbol{x_i}\}_{i=1}^N$, number of iterations: $L$

**Initialise the dictionary** $\boldsymbol{D}^0$ with columns normalised.

For $J$ =1 to number of iterations ($L$)

**1.** Find the sparse representation $\{\boldsymbol{w_i}\}_{i=1}^N$ for each $\{\boldsymbol{x_i}\}_{i=1}^N$ using appropriate algorithm.

**2.** For each atom $\boldsymbol{d}_k$ $(k = 1, \dots K)$ in the dictionary $\boldsymbol{D}^{j-1}$ (obtained from the previous iteration)

**a)** Define the set $r_k$ of training samples that use this atom

$$r_k = \{i|1 \le i \ge N, w_i(k) \ne 0\}$$

**b)** Compute the residual error matrix $\boldsymbol{E}_k$

$$\boldsymbol{E}_k = \boldsymbol{X} - \sum_{j\ne k} \boldsymbol{d_j}\,\boldsymbol{w_j^T}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2780190, IEEE Access

Form a restricted matrix $\boldsymbol{E}_k^R$ from columns with indices $r_k$ and apply SVD to it:

$$\boldsymbol{E}_k^R = \boldsymbol{USV^T}$$

**c)** Update the atom $\boldsymbol{d_k}$ to be the first column of $\boldsymbol{U}$ and update the coefficient vector $\boldsymbol{w}_k^R$ to be the first column of $\boldsymbol{V}$ weighted by the largest singular value $\boldsymbol{S}$ (1,1).

**3.** Set $J = J + 1$

### a. *The ADMM-LASSO Algorithm*

ADMM is a general optimisation technique that gives a flexible framework to solve a variety of convex optimisation problems including the LASSO problem. In particular, ADMM introduces an auxiliary variable $\boldsymbol{z}$ to split the LASSO objective terms into two parts, a decoupled $\ell_2$ – loss objective function of $\boldsymbol{w}$ and a coupled $\ell_1$ – consensus constraint function of $\boldsymbol{z}$ to obtain [30]:

$$\text{Minimise } \frac{1}{2}\ \|\boldsymbol{x} - \boldsymbol{Dw}\|_2^2 + \lambda\|\boldsymbol{z}\|_1$$
$$\text{Subject to } \boldsymbol{w} - \boldsymbol{z} = \boldsymbol{0} \qquad (4)$$

The augmented Lagrangian for the optimisation problem (4) takes the form:

$$\boldsymbol{L}_\rho(\boldsymbol{w}, \boldsymbol{z}, \boldsymbol{u}) = \frac{1}{2}\ \|\boldsymbol{x} - \boldsymbol{Dw}\|_2^2 + \lambda\|\boldsymbol{z}\|_1 + \rho\boldsymbol{u}^T(\boldsymbol{w} - \boldsymbol{z}) + \rho/2\|\boldsymbol{w} - \boldsymbol{z}\|_2^2 \qquad (5)$$

where $\rho > 0$ is a penalty parameter and $\rho\boldsymbol{u}$ is the dual variable.

The ADMM algorithm consists of the three main recursive steps: First, the augmented Lagrangian (5) is minimised with respect to: $\boldsymbol{w}^{(k+1)} := \text{argmin}_w\, \boldsymbol{L}_\rho(\boldsymbol{w}, \boldsymbol{z}^{(k)}, \boldsymbol{\mu}^{(k)})$. Second, it is minimised with respect to: $\boldsymbol{z}^{(k+1)} := \text{argmin}_z\, \boldsymbol{L}_\rho(\boldsymbol{w}^{(k+1)}, \boldsymbol{z}, \boldsymbol{\mu}^{(k)})$. Finally, the dual variable is updated via a simple linear update [30].

The first minimisation step updates $\boldsymbol{w}$ and takes the form of a ridge regression (Tikhonov-regularised least squares) problem. This later has the explicit solution:

$$\begin{aligned}\boldsymbol{w}^{(k+1)} &= \underset{w}{\text{argmin}}\, \boldsymbol{L}_\rho\left(\boldsymbol{w}, \boldsymbol{z}^{(k)}, \boldsymbol{\mu}^{(k)}\right) \\ &= (\boldsymbol{D^T D} + \rho\boldsymbol{I})^{-1}\left(\boldsymbol{D^T x} + \rho\left(\boldsymbol{z}^{(k)} - \boldsymbol{\mu}^{(k)}\right)\right) \qquad (6)\end{aligned}$$

The second minimisation step results in an efficient soft thresholding update for $\boldsymbol{z}$:

$$\begin{aligned}\boldsymbol{z}^{(k+1)} &= \underset{z}{\text{argmin}}\, \boldsymbol{L}_\rho\left(\boldsymbol{w}^{(k+1)}, \boldsymbol{z}, \boldsymbol{\mu}^{(k)}\right) \\ &= \boldsymbol{T}_{\lambda/\rho}(\boldsymbol{w}^{(k+1)} + \boldsymbol{\mu}^{(k)}) \qquad (7)\end{aligned}$$

$\boldsymbol{T}_{\lambda/\rho}(\cdot)$ is the term-by-term soft thresholding operator such that:

$$T_{\lambda/\rho}(\boldsymbol{v})[n] = \begin{cases} \boldsymbol{v}[n] - \lambda/\rho, & \boldsymbol{v}[n] > \lambda/\rho \\ 0, & |\boldsymbol{v}[n]| \le \lambda/\rho \\ \boldsymbol{v}[n] + \lambda/\rho, & \boldsymbol{v}[n] < -\lambda/\rho \end{cases}$$

Or equivalently:

$$S_{\lambda/\rho}(v)[n] = sign(v[n]) * \max(v[n] - \lambda/\rho, 0)$$

Finally, the update of the dual variable $\boldsymbol{u}$ can be expressed explicitly as:

$$\boldsymbol{u}^{(k+1)} = \boldsymbol{u}^{(k)} + \boldsymbol{w}^{(k+1)} - \boldsymbol{z}^{(k+1)} \qquad (8)$$

It is worth noting that in the case of a wide matrix $\boldsymbol{D}$ $(\mathbb{R}^{N\times K})$, the Sherman-Morrison-Woodbury inversion formula can be used to substitute $(\boldsymbol{D^T D} + \rho\boldsymbol{I})^{-1}$ by a smaller matrix$(I + \frac{1}{\rho}\boldsymbol{DD^T})^{-1}$. In addition, updating the penalty parameter $\rho$ at each iteration can improve the convergence and reduce its dependency on the initial guess $\rho_0$.

### b. *Efficient Implementation of $\boldsymbol{w}$ –Update*

In practice, the first step (updating $\boldsymbol{w}$) represents a computational bottleneck especially for large matrices, which call for computationally more efficient methods to solve these systems of equations. In our implementation, we used a decomposition technique that exploits the special structure of the matrix $(\boldsymbol{D^T D} + \rho\boldsymbol{I})$ to update its inverse [32].

In particular, applying the eigenvalues decomposition to $\boldsymbol{D^T D}$ gives:

$$\boldsymbol{D^T D} = \boldsymbol{U\Sigma U}^T \qquad (9)$$

where $\boldsymbol{U}$ is the orthogonal eigenvector matrix of $\boldsymbol{D^T D}$, while $\boldsymbol{\Sigma}$ is the real valued diagonal positive semidefinite matrix of its eigenvalues $\sigma_i$, and $^T$ denotes the transpose operation.

It is easy, using some basic linear algebra, to show that the matrices $\boldsymbol{D^T D}$ and $(\boldsymbol{D^T D} + \rho\boldsymbol{I})$ have the same eigenvectors and that the eigenvalues of the later matrix are simply: $\boldsymbol{\Sigma} + \rho\boldsymbol{I}$. Therefore:

$$(\boldsymbol{D^T D} + \rho_k\boldsymbol{I})^{-1} = \boldsymbol{U^T}(\boldsymbol{\Sigma} + \rho_k\boldsymbol{I})^{-1}\boldsymbol{U} \qquad (10)$$

The eigenvectors (matrix $\boldsymbol{U}$) and the eigenvalues (diagonal matrix $\boldsymbol{\Sigma}$) are pre-computed offline and the inverse update $(\boldsymbol{D^T D} + \rho_k\boldsymbol{I})^{-1}$ can be calculated quickly. In particular, the inverse of the diagonal matrix $(\boldsymbol{\Sigma} + \rho_k\boldsymbol{I})$ is the reciprocals of its diagonal entries. One of the main advantages of using this approach is that it does not need to calculate inverse of the matrix within the learning iterations, which significantly reduce the processing time as well as the complexity of the implementation. The parameter $\rho_k$ is updated in accordance to the rule described in [30]. In addition, the speed of the algorithm is further improved by exploiting the parallelism of the matrix-vector multiplication as well as the vector operations. Moreover, the vector matrix multiplication $\boldsymbol{D^T x}$ in equation (6) is calculated only once at the outset.

### c. *Parameterised-Lorenz Curve and Inequality Measures*

In this section, we introduce two popular inequality measures drawn from the theory of income distribution, namely the Pietra (PI) and the Gini (GI) indexes.

The standard definitions of these inequality measures are not valid for vectors with negative entries; therefore, absolute values are used when applied as sparsity measures.

Let $\boldsymbol{w}$ be a vector with its entries arranged in a descending order such that: $\boldsymbol{w} = [w_{\{1\}}, w_{\{2\}}, \dots, w_{\{K\}}]$. The parameterised-Lorenz curve shown in Fig. 1 is an increasing convex graphical representation of the cumulative sum of $\boldsymbol{w}_{\{j\}}$ normalised by the $\boldsymbol{\ell}^1$ norm of $\boldsymbol{w}$ given by [27]:

$$\Lambda\left(\frac{i}{K}\right) = \frac{1}{\|\boldsymbol{w}\|_1}\sum_{j=1}^{i}\left|\boldsymbol{w}_{\{j\}}\right|\ , for\ i = 0, \dots, K \qquad (11)$$

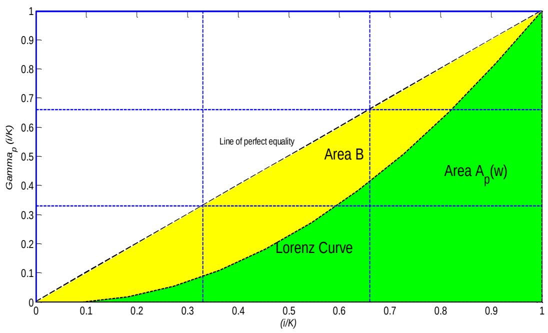


Fig. 1. Lorenz curve.

The GI is defined as the ratio of the area between the 45° line and $A_p(\boldsymbol{w})$ and the area beneath the 45° line. This is mathematically expressed as:

$$\mathrm{GI}(\boldsymbol{w}) = \frac{B}{A_p(\boldsymbol{w}) + B} = 2B = 1 - 2A_p(\boldsymbol{w}) \qquad (12)$$

GI takes values between 0 (the least sparse scenario where all the elements of $\boldsymbol{w}$ are equal) and 1 (the most sparse scenario where all the signal energy is concentrated in one coefficient). In addition, the Gini index is quasi-convex in $|\boldsymbol{w}\{j\}|$ , its value is independent of the size of the vector, and it is scale-invariant and independent of the total energy of the signal.

The PI, also known as the Relative Mean Deviation (RMD), is a quasiconvex function of $\boldsymbol{w}$ defined as the maximum vertical deviation between the equality line joining (0, 0) to (1, 1) and the Lorenz curve. In mathematical terms, the un-normalised PI can be expressed as one half of the relative mean deviation:

$$P(\boldsymbol{w}) = \frac{\sum_{i=1}^{K} |w_i - \bar{\boldsymbol{w}}|}{2\bar{\boldsymbol{w}}} \qquad (13)$$

### d. Classification Algorithm

The main idea is that a relatively high value of the GI/PI index of a sparse represented signal will be associated with the dictionary learned from data belonging the same class. The maximum index is then used to predict the correct class of a test data set. The structure of the proposed classification algorithm consists of the following steps:

Step 1: Learn $q$-dictionaries $(\boldsymbol{D}_1, \boldsymbol{D}_2, \ldots, \boldsymbol{D}_q)$ ($q$ is the number of classes) by solving (2). Each dictionary is learned using the training data belonging to a specific class.

Step 2: Find $q$ different sparse vectors $(\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_q)$ of each new test signal, $\boldsymbol{x}_t$, by solving (3) for each dictionary $\boldsymbol{D}_i$ $(i = 1,2, .. q)$

Step 3: Use the GI/PI index to predict $\boldsymbol{x_t}$ class label such that:
Class label $(\boldsymbol{x}_t)$ = argmax $(\psi(\boldsymbol{w}_i))$ $i = 1,2, .. q$, where
$\psi: \mathbb{R}^N \rightarrow \mathbb{R}$ is the Gini or Pietra index.

### e. Data Collection and Pre-processing

To evaluate the effectiveness of the proposed approach, validation was carried out using the MIT-BIH Arrhythmia Database. The data is bandpass filtered at 1-100 Hz and sampled at 360 Hz to facilitate the elimination of the power line interference using notch-filter. All the heartbeats in the database were manually annotated and made available for the users. In addition, fiducial point times are provided to locate the R-peaks [33].

A second-order notch-filter and two cascaded median filters of lengths 108 $(0.3f_s)$ and 216 $(0.6f_s)$ samples were used to remove power line interference and baseline wander from the ECG signal. The first median filter aimed at removing the QRS complexes and the P-waves from the ECG, while the second filter was employed to remove the T waves. The output of the second filter is subtracted from the original ECG signal to obtain a corrected baseline ECG.

The Association for the Advancement of Medical Instrumentation (AAMI) recommendation emphasises the discrimination of Ventricular Ectopic Beats (VEBs) from the other three classes namely, Normal beats (N), Supraventricular Ectopic Beats (SVEBs) and Fusion beats (F), [34]. Each of the aforementioned categories contains different types of arrhythmias as summarised in Table 1.

TABLE I: ARRHYTHMIA CATEGORIES

| | Sub-category | Beat types |
|---|---|---|
| Category (1) | VEB | Premature ventricular contraction, Ventricular escape. |
| Category (2) | N | Normal beats, Left bundle branch Block, Right bundle branch block, Atrial escape beat and Nodal escape beat. |
| | S | Atrial premature beats, Aberrated atrial premature beats, Nodal premature beats and Supraventricular premature beats. |
| | F | Fusion of ventricular and normal beats. |

The exemplary data in Fig. 2, taken from record 106, shows the within the class variability of the VEB class.

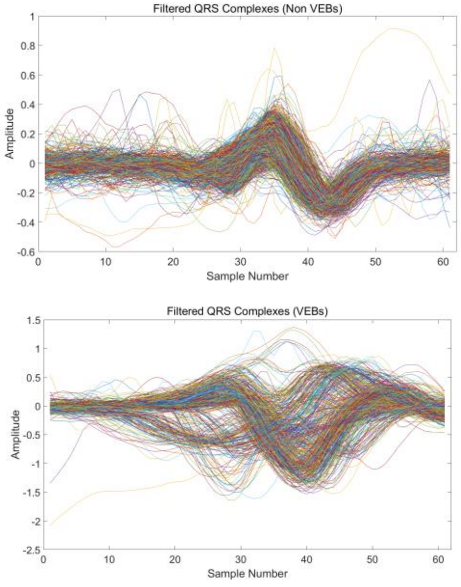


Fig. 2. Exemplary waveforms of the two classes of QRS complexes taken from record 106**.**

## III. HARDWARE IMPLEMENTATION

The Zedboard is used for hardware implementation, which contains two subsystems: Programmable Logic (PL) and

Processing Systems (PS). Both PL and PS are integrated into a Xilinx Zynq-7000 XC7Z020 SoC [35].

The sparsifying matrices or dictionaries $\boldsymbol{D}_i$ were first learned using ECG QRS complexes training data in MATLAB simulation environment and its right singular vectors and singular values are stored in a set of data files.

The ADMM algorithm is first implemented in C++, and then synthesised and translated into a hardware description language (HDL). A set of pragma directives are used to optimise the codes for hardware implementation, where the overall goal of the optimisation is to achieve the high throughput architecture with minimal usage of hardware recourses [36].

Since arrays are implemented as block random access memory (block-RAM) which has a maximum of two data ports. This limits the throughput of a read/write (or load/store) intensive algorithm. However, the bandwidth can be improved by splitting the array (a single block-RAM resource) into multiple smaller arrays (multiple block-RAMs), which effectively increases the number of ports. Therefore, the array ***U*** [*N*][*N*] is partitioned into *f* small arrays in both dimensions, where each array has a size of *N/f* × *N/f* (*#pragma HLS ARRAY_PARTITION block factor=f*). Fig. 3 shows the partitioned arrays.

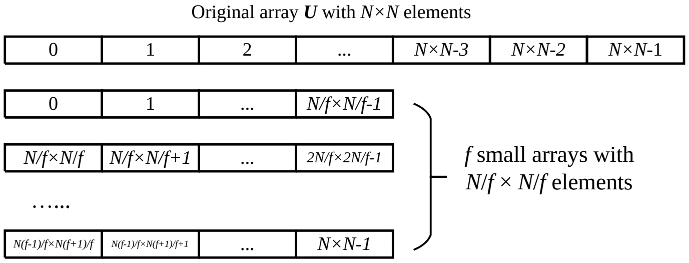


Fig.3. Partitioned arrays.

The proposed implementation uses 32-bit floating point arithmetic. C/Resistor–Transistor Logic (RTL) simulation is performed before exporting the RTL as a Vivado's IP core. The pre-synthesis resource report is used for design resource exploration and performance estimation. The RTL is exported as IP core to be synthesised and implemented in Vivado (v2016.3). The proposed algorithm is implemented on a Xilinx Zynq-7000 XC7Z020 all programmable SoC.

The solution is then exported as an IP core connected with AXI4-Stream interface to the accelerated coherency port (ACP) on AP SoC PS. The connection is made through a direct memory access (DMA) core in the PL subsystem. SDSoC (v2016.3) is used to interface the AP SoC PL hardware, the peripheral, the DMA engine, an AXI timer as well as other data mover logics. The SDSoC is also used to design the AP SoC PS software to manage the peripherals and loading the testing data from the external SD card, as shown in Fig. 4.

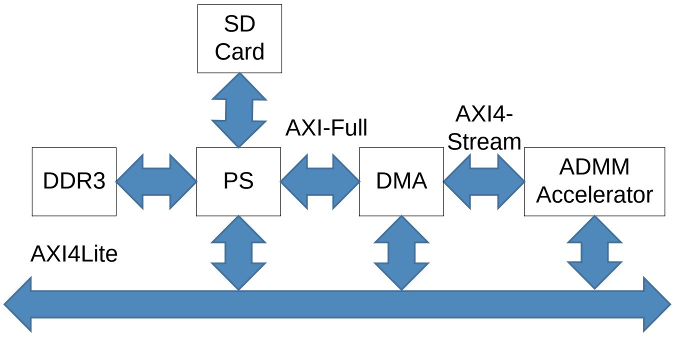
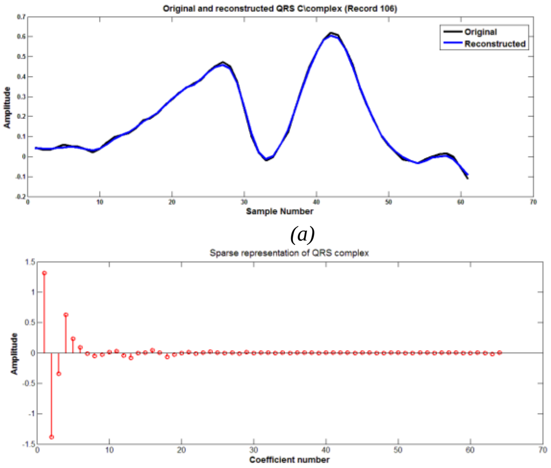


Fig.4. Implementation overview.

Fig. 5 (a) and Fig. 5 (b) represent the results of sparse representation and reconstruction of an abnormal QRS complex extracted from record 106 in the database.



*(a)*

*(b)*

Fig. 5. (a) Original and reconstructed QRS complex from record 106, (b) Sparse representation.

## IV. Performance Evaluation

### *a) Hardware implementation results*

This subsection begins by reporting the Field-programmable Gate Array (FPGA) hardware resources usage. Subsequently, the software implementation (on CPU) and the hardware implementation of the proposed algorithm are compared. Finally, the comparison of the proposed solution with existing hardware-based implementations is provided.

The resource utilisation of the proposed implementation is shown in Table II.

We propose five HW implementations, presented and named from factor = 1, 2, 4, 8, 16 in Table II, where the factor value decides how many small arrays will be partitioned from the original array. The required hardware resources are also presented (look-up-table (LUT), flip-flop (FF), BRAM_18K and DSP48E blocks). The number of elements and ratio compared to available resources are given. All configurations have been set at 100 MHz.

TABLE II: FPGA RESOURCE UTILISATION

| | Array Partition Factor | | | | |
|---|---|---|---|---|---|
| **Resource** | **1** | **2** | **4** | **8** | **16** |
| **BRAM_18K** | 186 (17%) | 634 (58%) | 8 (4%) | 84 (7%) | 126 (11%) |
| **DSP48E** | 41 (4%) | 51 (5%) | 75 (8%) | 115 (12%) | 190 (21%) |
| **Flip-Flop** | 11,863 (2%) | 14,224 (3%) | 137,216 (31%) | 147,076 (33%) | 180,139 (41%) |
| **LUT** | 18,597 (8%) | 32,727 (14%) | 47,900 (21%) | 61,004 (27%) | 103,055 (47%) |

The first implementation (factor = 1) does not include any synthesis directive for array partition for all the arrays in the top function. It has the least resources usage compared to other implementations. As a result of this, the loops cannot be fully pipelined or unrolled due to the limitation of the memory ports. Therefore, the processing time is the highest one. Along with the increase of the value of factor, the resource usage is increased dramatically, especially the usages of LUT and FF. This is due to that the increase of the number of arrays and multiplier/adders needed in the pipelines. The processing time is reduced along with the increased usage of pipeline and parallelism in the implementations, once it reaches its optimal, the downtrend speed is slowed.

As it can be seen from Table II, since all the arrays in the loops have been partitioned with different factors, the usage of BRAMs and DSPs is dominant due to the duplications of the arrays and multiplications. The usage of FF and LUT is increased significantly when the partition factor is greater than 4. In addition, the DATAFLOW pragma has been used to pipeline the operation blocks; a ping-pong buffer has been placed between every two operation blocks to maintain the data rate. Therefore, additional memory blocks are added to the design. The major benefit of doing this is to make the functions or loops at the top level operate in parallel and improve the overall throughput of the design. Fig. 4 shows the PL implementation results of resources utilisation and processing time with different optimisation pragmas.

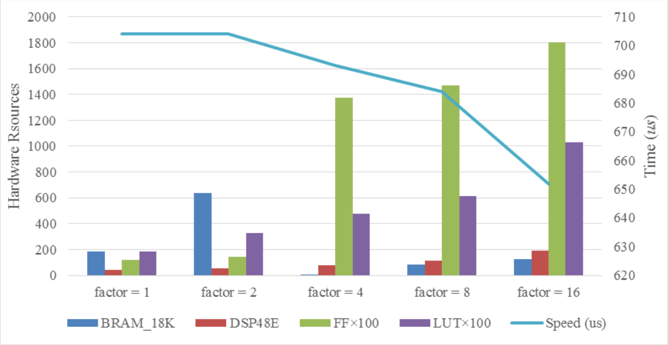


Fig. 6. Implementation results and timing of the hardware IP.

Compared to the average processing time per iteration using PC software implementation, the hardware implementation of the proposed approach has been reduced from 1649 us to 693 us which makes it twice faster. On average, the algorithm converges in 100 iterations.

The details of estimated power consumption of the hardware implementation of the proposed algorithm are summarised in Table III. Compared to the Processing System 7 (PS7), the custom logic blocks consume only small portions of the total power consumption due to the efficient usage of the on-chip resources. To partition the arrays, more logic and memory blocks are used, which is reflected in the energy utilisation results of signals and BRAMs. The total on-chip power consumption estimations for implementing the classifier is 0.934 W out of 2.593 W.

TABLE III. ESTIMATION OF POWER CONSUMPTION

| | **Utilization Details** | **Proposed approach** | |
|---|---|---|---|
| | | **Power (W)** | **Utilization (%)** |
| Dynamic Power Consumption | Clock | 0.210 | 9 |
| | Signals | 0.176 | 8 |
| | Logic | 0.145 | 6 |
| | DSP | 0.038 | 2 |
| | BRAM | 0.108 | 5 |
| | PS7 | 1.660 | 70 |
| Static Power Consumption | Device Static | 0.257 | 10 |

### *b) Classification performances*

To assess the merits of the proposed algorithm, we performed the classification using all the records containing VEBs beats from the MIT-BIH database [14] using three classification scenarios. In the first scenario, we randomly selected 100 QRS complexes from each subject, representing each category for training and the remaining beats for testing using 10 runs experiments (we used patient-specific approach). We used a window of 61 samples centred at the R-peak to extract the QRS complexes. In the second scenario, we randomly selected 1200 QRS complexes from each category from a pool of data containing all the records to learn the dictionaries. The remaining heartbeats are used for testing. In the last scenario, the database is split into two independent datasets. The first dataset is composed of patients' records with identification numbers 106,116,119 and 208. The second data set is composed of patient's records 201, 203, 207, 215 and 232. The first dataset was used to learn the dictionaries while the second was held for testing.

Fig.7 is a graphical representation of the Lorenz curve of a VEB test QRS complex (Category 2) in the transform (sparse) domain.

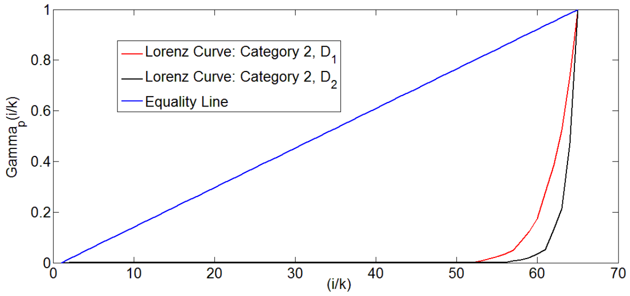


Fig. 7. Lorenz curves: (a) Test QRS complex for Category 2

TABLE IV: PERFORMANCE OF THE PROPOSED CLASSIFICATION ALGORITHM

| Record number | Number of beats | | Accuracy (%) | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Class 1 | Class2 | Training set | | Test set | | Training set | | Test set | |
| | | | Gini based classification | | | | Pietra based classification | | | |
| 106 | 1500 | 517 | 96.00±2.58 | 89.52±3.03 | 94.00±3.12 | 89.49±2.98 | 95.90±2.80 | 92.50±1.17 | 92.10±3.60 | 88.51±1.17 |
| 116 | 2292 | 109 | 99.90±0.31 | 99.45±0.27 | 99.80±0.63 | 98.88±3.513 | 100.00±0.00 | 99.82±0.051 | 99.90±0.31 | 100.00±0.00 |
| 119 | 1532 | 444 | 100.00 ± 0.00 | 99.99 ± 0.02 | 100.00 ± 0.00 | 99.97 ± 0.09 | 100.00±0.00 | 99.99±0.02 | 100.00±00 | 99.97±0.09 |
| 201 | 1756 | 198 | 100.00±0.00 | 99.19 ± 0.55 | 100.00 ± 0.00 | 99.48 ± 0.53 | 99.90±0.31 | 98.97±0.65 | 100.00±0.00 | 99.48±0.53 |
| 203 | 2522 | 444 | 92.00 ± 2.82 | 84.77 ± 3.95 | 82.50 ± 7.63 | 74.53 ± 7.25 | 94.80±3.22 | 88.60±2.78 | 85.50±6.18 | 81.51±6.05 |
| 207 | 1638 | 210 | 99.30 ± 0.67 | 97.45 ± 0.62 | 97.00 ± 2.10 | 95.09 ± 2.50 | 99.40±0.69 | 98.50±0.88 | 95.60±3.30 | 92.54±3.17 |
| 208 | 1954 | 988 | 95.50 ± 2.17 | 88.03 ± 2.35 | 99.40 ± 0.51 | 97.90 ± 1.16 | 94.30±2.94 | 89.35±2.52 | 98.60±0.84 | 98.31±0.59 |
| 215 | 3188 | 164 | 99.90±0.31 | 98.03±0.52 | 99.10±0.87 | 98.90±1.28 | 100.00±0.00 | 99.47±0.22 | 98.00±0.81 | 99.37±0.80 |
| 223 | 2121 | 473 | 99.70±0.67 | 98.66±0.42 | 93.90±4.81 | 92.89±5.56 | 99.60±0.51 | 99.07±0.19 | 95.40±2.41 | 94.95±2.54 |
| All records cross validation | 18503 | 3547 | 90.25 ± 3.36 | 89.25±3.71 | 81.54±3.39 | 80.69±3.53 | 92.72±2.11 | 91.71±2.28 | 82.50±4.82 | 81.26±5.02 |
| Independent sets | 18503 | 3547 | 95.77±4.94 | 84.54±1.88 | 75.90±8.09 | 70.46±8.32 | 93.91±2.19 | 87.33±2.97 | 71.32±10.84 | 66.10±13.11 |

The good separation capability of the GI and the PI indexes between the VEB class and the other arrhythmia classes is evident in this test data. To be more specific, the black Lorenz curve represents the sparse representation of the VEB QRS complex calculated using a dictionary learned from VEB data ($\boldsymbol{D}_2$). Meanwhile, the red Lorenz curve represents its sparse representation using a dictionary learned from data belonging to the other classes ($\boldsymbol{D}_1$).

The details of the distribution of heartbeats in each record are summarised in Table IV. The first column represents the identification number of the patients in the dataset. The subsequent two columns provide the number of beats in each category. The last columns provide the binary classification results of VEB (V) against the other classes (N, S and F) in terms of accuracy for the training and test sets respectively.

It is clear from the results that the selection of the training and the test sets plays a key role in the performances of the classifier. In the first scenario, the proposed algorithms (Gini based/Pietra based) achieved very good classification accuracies, though little ECG data was used for training. The relatively low classification accuracy of record 203 records compared to other records due to the morphological and temporal similarities between QRS belonging to the two categories of arrhythmias, the relatively small number of beats used for training (6.74 % of the total number of beats) and the large variation in the morphology of PVC beats.

The relatively low classification accuracy of record 203 records compared to other records due to the morphological and temporal similarities between QRS belonging to the two categories of arrhythmias, the relatively small number of beats used for training (6.74 % of the total number of beats) and the large variation in the morphology of PVC beats.

As expected, the achieved accuracies with the second scenario, widely used in the literature, were above 80% on the test set which is lower than the first scenario. The last scenario is the most challenging. The performances are comparable with those of more complex approaches such as linear discriminant classifier [5], SVM [7], weighted SVM [7] and hierarchical SVM [17] which achieved overall accuracies in the range of 80%. It is worth noting that the execution time of the Pietra and Gini indexes is neglected compared to the execution time of the ADMM algorithm.

## V. CONCLUSION

This paper presented an efficient algorithm for arrhythmia classification. The results were very encouraging as they are comparable to many state-of-the-art algorithms. The strength of the method comes from the well-definiteness (Quasi-convexity) of the used indexes and their simplicity of implementation. The use of matrix decomposition technique that exploits the special structure of symmetric matrices facilitated the hardware implementation of the ADMM algorithm. The validation was carried out on the MIT-BIH Arrhythmia Database using learned specifying matrices. Our implementation is suitable for low-power real-time applications and can be extended to several other sparse based classification problems.

## REFERENCES

[1] Teng, Xiao-Fei, et al. "Wearable medical systems for p-health." *IEEE reviews in Biomedical engineering* 1 (2008): 62-74.'

[2] D. Minoli, Building the Internet of Things With IPv6 and MIPv6: The Evolving World of M2M Communications. New York, NY, USA: Wiley, 2013.

[3] Samie, Farzad, et al. "Computation offloading and resource allocation for low-power IoT edge devices." *Internet of Things (WF-IoT), 2016 IEEE 3rd World Forum on*. IEEE, 2016.

[4] R. Plonsey and R. Barr, Bioelectricity: A Quantitative Approach. Springer Science & Business Media, 2007.

[5] Chazal, P. D., O'Dwyer, M., and Reilly, R. B. (2004). " Automatic Classification of Heartbeats Using ECG Morphology and Heartbeat Interval Features". *Biomedical Engineering, IEEE Transactions, 51*(7), pp. 1196-1206.

[6] Chazal, P. d., and Reilly, R. B. (2006). "A Patient-Adapting Heartbeat Classifier Using ECG Morphology and Heartbeat Interval Features". Biomedical Engineering, IEEE Transactions 53(12), pp. 2535-2543.

[7] Lannoy, G. d., Franc, D., Delbeke, J., and Verleysen, M. (2011). "Weighted SVMs and Feature Relevance Assessment in Supervised Heart Beat Classification". Communications in Computer and Information Science 127(4), pp. 212-223.

[8] Ge, D., Srinivasan, N., and Krishnan, S. M. (2002). "Cardiac Arrhythmia Classification Using Autoregressive Modeling". *BioMedical Engineering OnLine 1, 5*(1).

[9] Ham, F. M., and Han, S. (1996). " Classification of Cardiac Arrhythmias Using Fuzzy ARTMAP". *Biomedical Engineering, IEEE Transactions, 43*(4), pp. 425-429.

[10] Lin, K.-P., and Chang, W. H. (1989). " QRS Feature Extraction Using Linear Prediction". *Biomedical Engineering, IEEE Transactions, 36*(10), pp. 1050-1055.

[11] Chen, S.-W. (2000). "A Two-Stage Discrimination of Cardiac Arrhythmias Using a Total Least Squares-Based Prony Modeling Algorithm". *Biomedical Engineering, IEEE Transactions 47*(10), pp. 1317-1327.

[12] Chuang-Chien, C. H. I. U., Tong-Hong, L. I. N., and Liau, B.-Y. (2005). " Using Correlation Coefficient in ECG Waveform for Arrhythmia Detection". *Biomedical Engineering: Applications, Basis and Communications 17*(3), pp. 147-152.

[13] Osowski, S., and Linh, T. H. (2001). " ECG Beat Recognition Using Fuzzy Hybrid Neural Network". *Biomedical Engineering, IEEE Transactions, 48*(11), pp. 1265-1271.

[14] Tsipouras, M. G., and Fotiadis, D. I. (2003). " An Efficient System for the Detection of Arrhythmic Segments in ECG Recordings Based on Non-Linear Features of the RR Interval Signal". *In Computers in Cardiology, IEEE*, pp. 533-536.

[15] Minami, K.-i., Nakajima, H., and Toyoshima, T. (1999). " Real-Time Discrimination of Ventricular Tachyarrhythmia with Fourier-Transform Neural Network". *Biomedical Engineering, IEEE Transactions 46*(2), pp. 179-185.

[16] Özbay, Y., and Tezel, G. (2010). " A New Method for Classification of ECG Arrhythmias Using Neural Network with Adaptive Activation Function". *Digital Signal Processing 20*(4), pp. 1040-1049.

[17] Park, K. S., Cho, B. H., Lee, D. H., Song, S. H., Lee, J. S., Chee, Y. J., Kim, I. Y., and Kim, S. I. (2008). " Hierarchical Support Vector Machine Based Heartbeat Classification Using Higher Order Statistics and Hermite Basis Function". *In Computers in Cardiology, IEEE*, pp. 229-232.

[18] Wiens, J., and Guttag, J. V. (2010). " Active Learning Applied to Patient-Adaptive Heartbeat Classification". *Advances in Neural Information Processing Systems, 23*, pp. 2442-2450.

[19] Castillo, O., Melin, P., Ramírez, E., and Soria, J. (2011). " Hybrid Intelligent System for Cardiac Arrhythmia Classification with Fuzzy K-Nearest Neighbors and Neural Networks Combined with a fuzzy System". *Expert Systems with Applications, 39*, pp. 2947–2955.

[20] Dokur, Z., and Olmez, T. (2001). " ECG Beat Classification by A Novel Hybrid Neural Network". *Computer Methods and Programs in Biomedicine 66*( 2-3 ), pp. 167-181.

[21] Bote, Jose Manuel, et al. (2017)."A modular low-complexity ECG delineation algorithm for real-time embedded systems." *IEEE Journal of Biomedical and Health Informatics*

[22] H.Ghasemzadeh, N.Amini, R.Saeedi, and M.Sarrafzadeh, (2015). “Power-aware computing in wearable sensor networks: an optimal feature selection,” IEEE Trans. Mobile Comput., vol. 14, no. 4, pp. 800–812.

[23] Tošić, Ivana, and Pascal Frossard. (2011) "Dictionary learning." Signal Processing Magazine, IEEE 28, no. 2: 27-38.

[24] A. Fawzi, M. Davies, and P. Frossard, “Dictionary learning for fast classification based on soft-thresholding,” Int. J. Comput. Vis., vol. 114, no. 2-3, pp. 306–321, 2015.

[25] Zonoobi, Dornoosh, Ashraf Kassim, and Yedatore V. Venkatesh. (2011) "Gini index as sparsity measure for signal reconstruction from compressive samples." IEEE Journal of Selected Topics in Signal Processing, 5(5): 927-932.

[26] Baali, Hamza, Mesbah, Mostefa, (2015) “Ventricular Ectopic Beats Classification using Sparse Representation and Gini Index” The 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society.

[27] S. Rickard. (2006). Sparse sources are separated sources. Proceedings of the 16th Annual European Signal Processing Conference, Florence, Italy.

[28] M. Aharon, M. Elad, and A. Bruckstein, (2006) “K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation,” *IEEE Trans. Sig Proc*, vol. 54, no. 11, pp. 4311–4322,

[29] Skretting, Karl, and Kjersti Engan. (2010). "Recursive least squares dictionary learning algorithm." *IEEE Trans. Sig Proc* 58.4: 2121-2130.

[30] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, *3*(1), 1-122.

[31] T. Hastie, R. Tibshirani, J. Friedman. (2009). *The elements of statistical learning* (Vol. 2, No. 1). Springer, New York.

[32] M. V. Afonso, J. M. Bioucas-Dias, and M. A. T. Figueiredo. Fast image recovery using variable splitting and constrained optimization. IEEE Trans. Image Process, 19(9):2345–2356, September 2010.

[33] G.B.Moody and R.G. Mark. (1990). The MIT-BIH arrhythmia database on CD-ROM and software for use with it. *Proceedings of IEEE Computers in Cardiology*, 185-188

[34] Recommended Practice for Testing and Reporting Performance Results of Ventricular Arrhythmia Detection Algorithms Association for the Advancement of Medical Instrumentation, Arlington, VA, 1987, ANSI ECAR:1987

[35] Xilinx Inc. (2015). *Zynq-7000 All Programmable SoC Data Sheet*. Available: www.xilinx.com

[36] Xilinx Inc. (2015). *Vivado HLS User Guide*. Available: www.xilinx.com