OPTIMIZED SYSTOLIC ARRAY FILTERS WITH NOISE CLASSIFICATION FOR EXTRACTING FECG

Preethi D.[1], Sasikala G.[2], Baranidharan V.[3], Gnanaprakash V.[2], Kumaran G.[4] and Mohan Sai Kumar[1]
[1]Department of Electronics and Communication Engineering, Tamil Nadu, India
[2]Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Tamil Nadu, India
[3]Bannari Amman Institute of Technology, Tamil Nadu, India
[4]Saveetha School of Engineering, Tamil Nadu, India
E-Mail: prettz_d@yahoo.in

ABSTRACT

To preserve the lives of both mother and foetus during the early phases of childbirth, heart abnormality diagnosis is essential. In remote places where understanding of maternal care is limited, the death rate caused by carelessness or a failure to detect abnormalities is still a problem. The signal will contain various external or internal noise sources to isolate the key components of the foetal heart rate from the mother's belly during labour. The likelihood that the genuine signal would be misinterpreted and result in a false report will increase in the presence of such noise sources. Although there are many software programs available for extracting the QRS features from foetal ECG signals, it is unavoidable that specialized hardware is required for a significant reduction in both area and power. This paper's main goal is to extract the QRS complex using LDA, then improve Social Spider classifier performance using the suggested TAODV as a distance metric calculator, and then compare against existing methods to discover sounds that are distorting the normal heart rate. Systolic array filter with suggested Glitch Avoidance Circuit employing MUX is simulated using Cadence Virtuoso in 65nm technology to remove noise from the observed QRS complex. Over 100 records with the necessary examples from MIT-BIH Arrhythmia were used in the simulations, and it was discovered that MATLAB 2010b was used to adopt a unique technique for classifying noise. The suggested TAODV-based SSA classifier's accuracy is 96.8%, whereas the accuracy of a filter with a glitch avoidance circuit is 96.13%. The primary benefit of these strategies comprises cutting-edge hardware and computational solutions.

Keywords: QRS Feature extraction using LDA, social spider optimization, systolic array filter, distance metric, glitch avoidance circuit using MUX.

Manuscript Received 19 September 2023; Revised 4 December 2023; Published 10 January 2024

1. INTRODUCTION

Using either invasive, like antepartum surveillance, or non-invasive, like intrapartum surveillance, Electronic Fetal Monitoring (EFM) can be perceived from 20 weeks without amplification to assess the foetal health status [1]. The process of extracting FECG signals from the mother's belly using signal-processing techniques is shown in Figure-1 [2], which is referred to as a non-invasive method for recording FECG without making physical contact with the foetus.

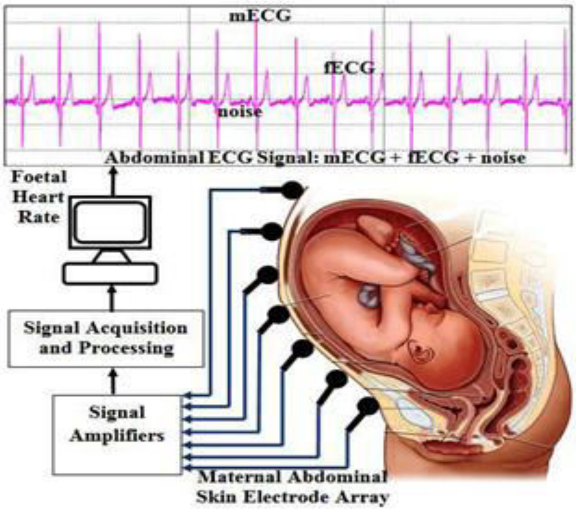

mECG
fECG
noise
Abdominal ECG Signal: mECG + fECG + noise
Foetal Heart Rate
Signal Acquisition and Processing
Signal Amplifiers
Maternal Abdominal Skin Electrode Array

Figure-1. Non-invasive measurement of FECG combined with MECG and noise.

Several methods use a single lead, two leads, or many leads to capture the QRS of the FECG in a non-invasive manner. Unfortunately, these methods of collecting FECG suffer from several issues since they are contaminated by numerous sources of interference. This is because the measurements are collected from the abdomen rather than directly from the foetus. The low signal level and spectral overlap of the MECG and FECG, in addition to these sources of interference, make the extraction more difficult.

The ECG signal is susceptible to low frequency-band at frequencies between 0.5 to 150 Hz, changing its properties as a result of interference from various noise sources. Additionally, the foetal signal energy is predicted to be less than 25% of the entire signal energy whereas the maximum amplitude of the QRS in an abdominal register up to 60 V in maternal recording and foetal recording fluctuates between 100 and 150 V. As a result, it presents many difficulties when trying to accurately extract information about cardiac arrhythmias [3]. Many of the produced noise sources listed below [4] pollute the FECG signals:

a) Interference from power lines: This interference is made up of harmonics of the 50/60 Hz frequency. It has a sinusoidal shape. Normally, the amplitude and frequency of this noise source are constant and won't change during the test. The interference is 20 ms long and has an amplitude of around 50% of the ECG peak.

b) Muscle noise is brought on by the electromyographic activity of the uterine and abdominal muscles. The sounds that muscles in the legs and abdomen make when they move. The interference has a frequency of 10 kHz and an amplitude that is roughly 10% of the ECG peak throughout 50 ms.

c) Electrode contact noise: When an electrode makes poor contact with the skin, electrode contact noise is produced. This transient interference is characterised as a baseline transition and it can happen only once or again quickly. The interference has a frequency of 50 Hz and an amplitude that is roughly 50% of the ECG peak over one second.

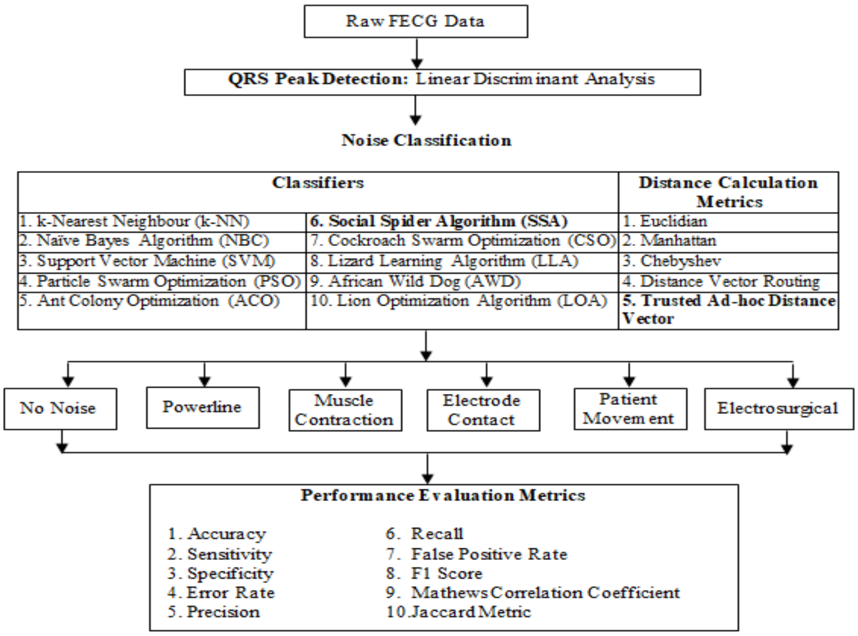
d) Motion artifact: When this noise is added to the system, it causes irregularities in the data and causes the information to be skewed. The interference has a frequency range of 0.5 to 100 Hz with an amplitude of about 500% of the ECG peak spread over 100 to 500 ms.

e) Electrosurgical noise: Electronic equipment typically produces noise, which can be reduced by selecting the right component specifications. The interference comprises of 100 k-1M Hz frequency and amplitude of around 200% of ECG peak over 1-10s.

The basic step in processing ECG data is to extract the QRS wave, which is shown in Figure-2. The standard deviation in the time sequence of the "R-R" interval is "R," which represents the apex of the complex and heartbeat variability [5]. On the MIT-BIH Arrhythmia database, appropriate classifiers are used to categorise and identify the type of noise intruding into the FHR with high accuracy and noise sensitivity upon detecting the QRS peak. The picture below also shows many classifiers and the distance metrics analysed for the performance assessment. Utilizing VLSI technology, the best-performing classifier is built, and the design metrics are assessed.

As all of the processors or embedded chips in ICs depend on a stable power supply to sustain the device's operation, any variation or fluctuation in this could affect the circuit's performance. Due to the high circuit complexity and density of VLSI, it might be difficult to identify the presence of such supply faults [6]. While performing QRS peak identification, these glitches are additionally misunderstood as a source of noise. Therefore, it is crucial to distinguish between supply changes and noise to correct the system problem and implement the FECG denoiser chip.

Work's scope: When considering the entire system, section 2 first discusses the extraction of the QRS peak using linear discriminant analysis (LDA) followed by the enhancement in the performance of classifiers for better noise classification is discussed in section 3. Section 4 discusses the glitch avoidance circuit employing MUX and its corresponding VLSI realisation. Section 5 then summarises the performance validation and conclusion.



Classifiers		Distance Calculation Metrics
1. k-Nearest Neighbour (k-NN)	**6. Social Spider Algorithm (SSA)**	1. Euclidian
2. Naïve Bayes Algorithm (NBC)	7. Cockroach Swarm Optimization (CSO)	2. Manhattan
3. Support Vector Machine (SVM)	8. Lizard Learning Algorithm (LLA)	3. Chebyshev
4. Particle Swarm Optimization (PSO)	9. African Wild Dog (AWD)	4. Distance Vector Routing
5. Ant Colony Optimization (ACO)	10. Lion Optimization Algorithm (LOA)	**5. Trusted Ad-hoc Distance Vector**

1. Accuracy
2. Sensitivity
3. Specificity
4. Error Rate
5. Precision
6. Recall
7. False Positive Rate
8. F1 Score
9. Mathews Correlation Coefficient
10. Jaccard Metric

Figure-2. ECG signal processing flow.

2. IMPLEMENTATION OF QRS PEAK DETECTION ALGORITHMS

Different techniques are used to compute the time intervals in the ECG signal when R peaks in the QRS complex are detected [7]. Each method is carried out using a set of common transformations, but there are some bottlenecks. The main problems with the Pan and Tompkins, Laguna, and Dubois algorithms are, respectively, inconsistent detection rates as P wave or other wave component amplitude and morphology increase, slower processing speeds, and higher error rates as a result of the detection of even signals with smaller amplitudes. This makes Linear Discriminant Analysis (LDA) the preferable algorithm over the others.

2.1 Linear discriminant analysis algorithm

Require: data matrix $Z \in T^{t\times N}$, $z_1 \in R^{t\times 1}$ is the "l-th" column of *Z*, label vector $y_i \in \{1,2,...,C\}$, *l=1,..., N.N_c, c=1,...,C* is the number of samples of each class [8].

Ensure: The projection matrix $M \in R^{p\times t}$

Step 1: Determine the mean vector of each class:

$$v_c = \frac{\sum_{l=1}^{N_C} z_l}{N_C} \in R^{dX1}$$

Step 2: Compute total mean vector: $m =$

$$\frac{\sum_{l=1}^{N} z_l}{N} \in R^{dX1}$$

Step 3: Compute within class scatter:

$$S_U = \sum_{c=1}^{C} \sum_{j=1}^{N_c} (z_j - v_c)(z_j - v_c)^T \in R^{txt}$$

Step 4: Compute between-class scatter,

$$S_b = \sum_{c=1}^{C} N_C (v_c - v)(v_c - v)^T \in R^{t\times t}$$

Step 5: Eigen-decomposition: $[V, D] = eig(S_U^{-1} S_b)$

Step 6: Obtain projection matrix: P being comprised of top-p eigen vectors corresponding to the largest eigen values.

The aforementioned procedure should be used to obtain the correct FECG data for subsequent analysis, such as normalization and QRS peak detection. The ECG amplitude rolls off during normalization so that the standard deviation and mean become unity and zero, respectively, eliminating any traces of DC, while the QRS peak detection aids in extracting crucial feature vectors from each beat to forecast or analyze the characteristics of the specific signal. Additionally, the feature extraction technique is used as a direct input to the classifier to remove unnecessary features and reduce the dimension of the raw signal. Other features that might be extracted include heart beat interval, frequency higher order cumulant features, and Hermite polynomials, similar to QRS heartbeat. As a result, testing time and computational complexity are reduced, and as a result, classification accuracy expectations are dramatically raised.

3. MATHEMATICAL MODELS FOR CLASSIFICATION OF NOISES FROM QRS PEAK

The QRS extraction is followed by noise classification in FECG signal similar to any classifier which has two intermediate steps such as training phase and testing phase with informative and discriminative features. Hence with the trained dataset, extraction and recognition of noise can be characterized. At the training phase, optimal feature vectors from various ECG beats are used for training the noise models, called a model database. Hence during the testing phase, an unknown signal present in FECG other than the normal QRS will be matched across known databases, and the noise that has the best match is concluded to be intruded into the normal FECG.

According to the review conducted, the optimization techniques presented in the next section have not yet been deployed as a classifier. Due to the factors like fitness function, optimization time, and function evaluations, this study seeks to achieve that. The MIT-BIH Arrhythmia database's 30-minute signal is used as the input data for the upcoming algorithm [8]. After that, it is broken up into 10 second segments. This indicates that there are 180 total pieces, each of which lasts for 10 seconds. A sliding window in the method reads two pieces at once. During each window, the algorithm also scans through all the possible high peak values and decides on an average value above which all peaks determined in the further chunks would be labeled as noise peaks. A high amplitude peak can generally be classified as noise if its amplitude is more than 75% larger than the peaks on either side, even if the threshold value entirely depends on the incoming signal. A social spider algorithm should include the following pre-processing steps:

3.1 Social Spider Algorithm

The spiders that have been placed on the web have an additional memory of their search area. To calculate the position and movement of the spider algorithm toward the ideal solution, the following parameters are chosen [9–10]:

Step 1: Calculate the intensity of vibration “I” on knowing the initial values of occurrence of QRS complex and fitness function of“s” within a Manhattan distance of $D(Q_a, Q_b) = \|Q_a - Q_b\|$ with “c” as a constant set by noise source,

$$I(Q_s, Q_s, t) = \log\left(\frac{1}{f(Q_s) - c} + 1\right) \quad \text{.........} (1)$$

Step 2: Standard deviation of position “ σ “ controlled by a user parameter “ r_a “ is,

$$I(Q_a, Q_b, t) = I(Q_a, Q_a, t)\ X\ exp\left(\frac{-D(Q_a, Q_b)}{\overline{\sigma}\ X\ r_a}\right) \text{.......} (2)$$

Step 3: The representation of walk over a distance of R with the initial point “f_o” of a random floating point is,

$$Q_s(t+1) = Q_s + (Q_s - Q_s(t-1))X\ r + \left(Q^{f_0}_{\ s} - Q_s\right)\Theta R \quad \text{....} (3)$$

From equation (10) the direct movement along the upper bound “ $\overline{x_i}$ “ or lower bound “ $\underline{x_i}$ “ over a threshold for noise estimation is given by,

$$Q_s, i(t+1) = \begin{cases} \left(\overline{x_i} - Q_s, i\right) X \quad r & if\ \ Q_s, i(t+1) > \overline{x_i} \\ \left(Q_s, i - \underline{x_i}\right) X\ \ r & if\ \ Q_s, i(t+1) < \underline{x_i} \end{cases} \quad \text{....} (4)$$

The spider movement is then updated and propagates until the entire chunks of data in a record are being traversed.

When the aforementioned algorithms are modified, if the peak is greater than the adaptive detection threshold (TC), the QRS complex is detected as noise (DT). The noise peak level (NPL) and the QRS peak level (QRSPL), respectively, are determined using the mean of an 8-point QRS buffer. The noise buffer stores the eight recordings of the most recent noise peak value using the formula DT = NPL + TC * (QRSPL NPL), much as the buffer collects the eight records of the most recent QRS peak value. These algorithms offer the best performance and an optimal solution for multiple noise classification, taking into account key factors such as traversal speed and parallel computation.

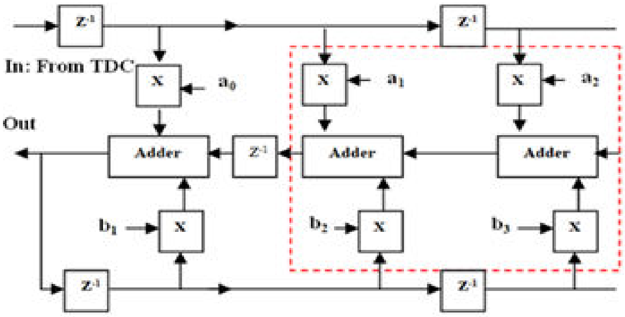


Figure-3. TDC based systolic array filter architecture.

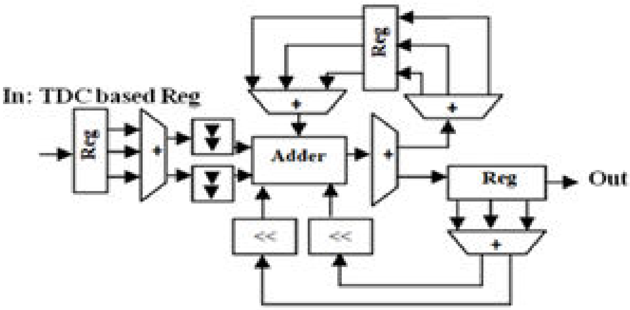


Figure-4. Program element constituents of systolic array.

Filter Architecture having transfer function $H(Z^{-1}) = \frac{\sum\limits_{i=0}^{N} a_i z^{-i}}{1 - \sum\limits_{I=1}^{N} b_i z^{-i}}$.

All of the aforementioned classifiers' optimization times are constrained by each constant, saving O(log n) if. However, the number of local searches in O is used to measure the function evaluations (n). A high amplitude T-wave could be prevented by maintaining the time interval between any two consecutive peaks at less than 225 milliseconds. The optimization time for all the above classifiers are bound by $O(n^2.(1/\rho)^{\varepsilon} + \frac{n/\rho}{\log(1/\rho)}$ for every constant $\varepsilon > 0$ and thus saves O(log n) if $\rho \leq 1/n^2$. On the other hand, the function evaluations are measured by the number of local searches in O(n). The distance between any two consecutive peaks is maintained to be less than 225 milliseconds, such that detecting a high amplitude T-wave as an R-peak could be avoided.

3.1.1 Proposed TAODV as distance metric in classifier

Based on the idea deduced from wireless ad-hoc networks, a modification to the previously stated classifier in terms of distance measurement between two neighbours is presented to prevent misclassification. To make it easier to extract MECG (QRS values-nodes) characteristics to classify noise, the TAODV algorithm [11] is incorporated into classifiers to assess the trust of its neighbours with an established threshold. The features derived from the MECG of the trust model are combined to determine whether they belong to one of the two classes, i.e., noisy or noise-less, based on the feedback gathered from the neighbours. By using relative judgment, it is possible to forecast node behaviour and distinguish between normal and impacted states. The updated weight between the factors of consideration, or "wAB," is the opinion value.

3.1.2 Trust judgment

The values in its opinion about a neighbouring node are used to modify predefined trust judgment rules. The weight representation $w^{AB} = (b^{AB}, d^{AB}, u^{AB})$ includes the opinion of belief, disbelief, or uncertainty. The detected QRS wave is normalized such that the values vary in the range (0, 1). Thus, a threshold of 0.5 (midpoint) is utilized to assess levels of belief, disbelief, and uncertainty.

b^{AB}: To prevent outliers, the belief of node A on node B will be computed based on the compactness of node B's five nearest neighbours. The degree of belief increases with decreasing distance.

d^{AB} : The distance equation |A-B| is used to compute the disbelief from node A to node B. Disbelief increases as distance increases.

u^{AB}: Using entropy, the degree of uncertainty between nodes A and B is calculated. Less entropy means that there is less uncertainty between nodes A and B.

After calculating the weight factor w^{AB} the following decisions are made as in Table-1.

Table-1. Conclusions about the weight factor w^{AB}.

Weight (w^{AB})	**Decision**	**Grouping under class**
w^{AB}> Threshold	Belief	No noise
w^{AB}< Threshold	Disbelief	Noisy region
w^{AB}= Threshold	Uncertainty	Compared against previous judgment: no previous opinion weight is updated as (0, 0, 1) else considered as noise if uncertainty if high

3.1.3 Trust updating

Once choices are taken on the opinions of nodes as shown in Table-2, the value of trust about the neighbours is updated later. When a fresh opinion is obtained, the number of the event's success or failure is transferred back to its initial value (0, 0, 1) to show how unsure it is. Because the judgment table updates itself, the categorization is more accurate than it was with the earlier methods [12-13].

To achieve the best performance, the threshold value listed in Tables 1 and 2 should be set at the maximum value. Utilizing the likelihood function, the appropriate threshold value is determined. The likelihood function, L, given in equation (5) is utilised because the discovered QRS features are Gaussian distributed. The value of trust concerning the neighbours is updated subsequently once decisions are made regarding the opinion of nodes as given in Table-2.

$$L = \frac{\sigma_0^{m_1+m_2}}{\sigma_1^{m_1}.\sigma_2^{m_2}} \qquad \ldots\ldots\ldots\ldots (5)$$

Table-2. Trust updating table.

Opinion	**Trust Table Value**	
	Positive Event (Belief)	**Negative Event (Disbelief)**
Greater than Threshold	**1**	**-**
Less than Threshold	**-**	**0**

Here, class 1 and class 2 QRS characteristics have been discovered using the letters m1 and m2, respectively. The total standard deviation of the two classes is 0, while the standard deviations of class 1 and class 2 individually are 1 and 2, respectively. The following equations are used to compute the standard deviation:

$$\mu = \frac{1}{n}\sum_{i=1}^{n} g_i \quad \text{………… (6)}$$

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n} \left(g_i - \mu\right)^2 \quad \text{………… (7)}$$

The mean and variance are represented by & in equations (6) and (7), respectively, while n stands for the number of features and g for the detected feature values. The probability ratio versus the error rate te of the classifier is taken into account while determining the ideal threshold value. Consider the k-NN classifier, for instance, which uses TAODV as a distance metric in the classification of powerline interference noise. Figure-5 plots the mistake rate vs likelihood ratio.

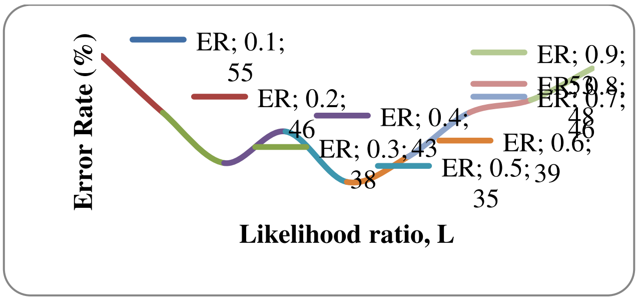


Figure-5. Determining the ideal threshold value for powerline noise classification.

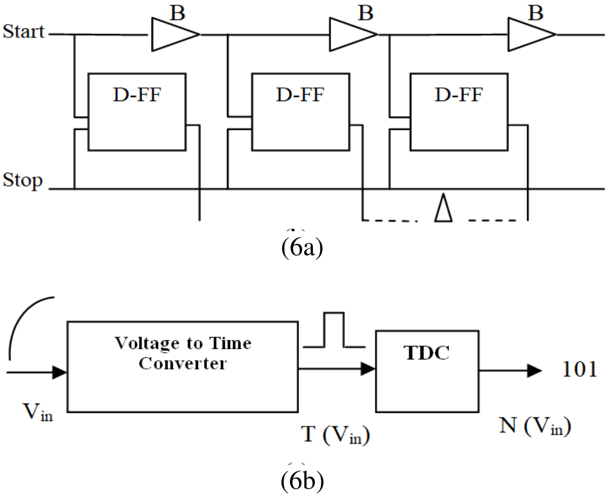
The likelihood ratio of 0.5 results in the lowest mistake rate, which is 35%, as illustrated in Figure 5. This is used as a threshold value for the TAODV distance metric in the k-NN classifier for Powerline Interference noise classification. The best threshold value for TAODV in other classifiers is calculated similarly. The computational results, which characterise the aforementioned techniques and carry out performance validation, are discussed in the next section.

4. GLITCH AVOIDANCE CIRCUIT USING MUX

The QRS detection can be divided into two steps, as described in Section 2. To increase the signal-to-noise ratio, the pre-processing stage highlights the desirable elements. Based on a user-specified threshold or noise, the peak detection stage determines whether an incoming peak is a real QRS complex. The systolic array architecture for these digital filters is used to compute one-dimensional recursive convolution after the pre-processing phase of the QRS detection technique. A peak would have been incorrectly identified as a QRS complex if a false positive beat detection occurred. The peak that was identified as an R-peak could have been a noisy peak or a T-wave with a high 18 amplitude. By effectively de-noising the signal using the filter architecture in Figure 3, noisy peak detection can be avoided. Due to logic errors, the classifier and filter's hardware implementation will undergo voltage variations. Such errors will affect how noise is predicted. To prevent multi path propagation, the glitch avoidance circuit is used as a crucial component of the TDC based systolic array filter architecture depicted in Figure-4.

4.1 Time to Digital Conversion-Based Systolic Array Filter Architecture

The Time to Digital Conversion (TDC) is performed by measuring the time interval between the two time instances concerning the clock. Since analog to digital suffers from various factors such as: i) scaling problems, ii) prone to noise, iii) fidelity decreases on the transmission of signal with high bit rates, and iv) unsuitable for processing data in the time domain, time domain is converted into digital [14-16].



(6a)

(6b)

Figure-6a. Time to Digital Circuit [TDC]; 6b.Block diagram of TDC.

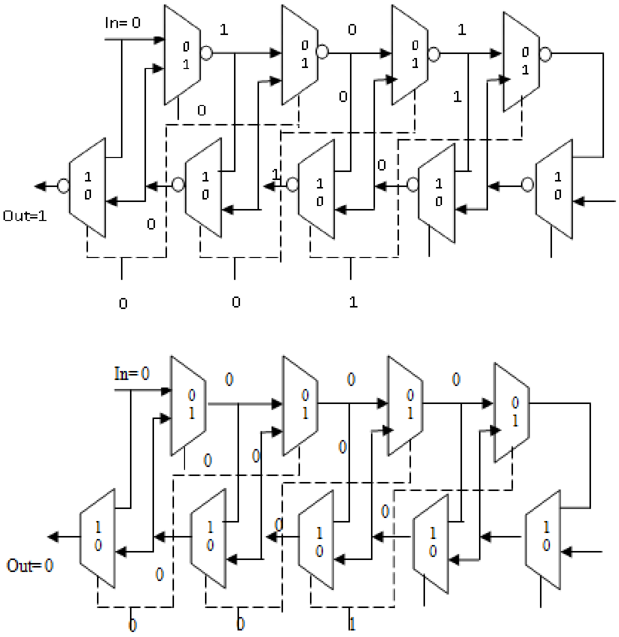


Figure-7a. Inverting Topology of MUX based delay line for 3 bitFigure7b. Non-Inverting Topology of MUX based delay line for 3 bit.

The filter characterised by the transfer function receives the QRS complex extracted from the FECG signal using the classifiers from section 3 as input, where ai (for I = 0 to N) and bi (for I = 1 to N) are real coefficients. The array structure in Figure-4 can be used to achieve an appropriate trade-off between throughput and the amount of hardware needed. The TDC-based circuit reconstructs the single shift register-based programme element of the filter. Components of the time-to-digital circuitry, such as delay flip-flops and buffers (B), are depicted in Figures 6a and 6b. When the start receives a rising edge, the input is delayed before passing to the following flip flops. The conversion process is stopped as soon as the stop signal receives the rising edge because it overtakes the start. The matching thermometer code N is obtained by sampling and decoding the variation in time instances estimated from the Vin (Vin). Resolution of the converter ' Δ' is determined by the difference between two delay flip flops. The block diagram in Figure-8 represents the overall process of temporal to digital conversion. The voltage to time converter serves as the source of the input for TDC, where Vin stands for the input voltage and 'T' is the corresponding time domain representation (Vin).

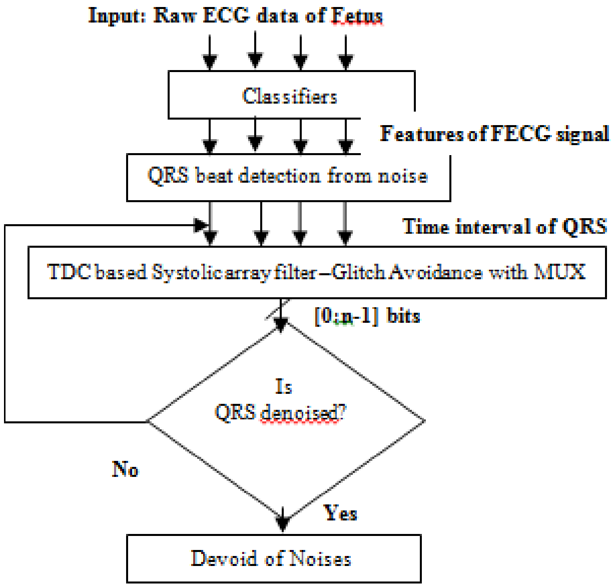

Input: Raw ECG data of Fetus
Classifiers
Features of FECG signal
QRS beat detection from noise
Time interval of QRS
TDC based Systolic array filter–Glitch Avoidance with MUX
[0:n-1] bits
Is QRS denoised?
No
Yes
Devoid of Noises

Figure-8. Block diagram depicting the noise removal process.

4.1.1 MUX based glitch avoidance circuit

Because of the propagation delay caused by clock skew, glitch reduces the performance of filters [17]. Synchronous filters for articulating the circulatory channel to change the frequency are necessary to offer constant output voltage. To transfer signals at the desired pace and prevent glitches, the clock skew must be avoided. Digital delay lines made up of delay components like MUX are therefore utilised to reduce glitches [18-20].

A glitch free MUX based delay line includes several intermediate multiplexers and an output multiplexer. The intermediate multiplexers are fed with a digital control code for selecting an intended delay. After processing of the plurality of delay units, they output a subset of delay signals that includes the signal presently selected with or without an additional delay transmitted through the output multiplexer.

Unlike NAND based Glitch Avoidance Circuit which has a driving circuit with two delay flip flops and three NAND gates for clock synchronization, MUX based delay lines serve as a replacement. Out of the six delay elements in NAND lines, two are inactive dummy cells used for load balancing. Therefore the construction of a multiplexer based delay line with just three multiplexers in place of six NAND gates is utilized. To illustrate, the transistor level implementation of NAND based DCDL in inverting topology requires 10{(6×4)+(8×4)+(2×2)+(3×4)}=720 transistors for the delay line with 10 delay cells including the driving circuit. Whereas for multiplexers just {(21×12)+(21×2)} =294 transistors are required, abruptly reducing the area and power [21-23].

4.1.1.1 Multiplexer based glitch avoidance circuit-inverting topology

The inverting delay line based on the multiplexer shown in Figure 7a are formed by adding an inverter represented by the bubble at the output of the multiplexer and are controlled by one control bit (thermometer code) Si. The control bits given are used as select lines for the multiplexer. Table-3 explains that whenever the control bits are either 0 or 1 the outputs would be with an inversion i.e., 1 or 0 respectively but with a delay. The select line of the MUX at the top leftmost corner is kept at 0 to pass on the data which is always given at the 0th terminal of the MUX both in the non- inverting and inverting topologies. For In= 0 and control code c= 2, the Out= 1 since it is an inverting topology. Though the control code increases by one, there are no glitches in the output as in the case of glitch free NAND based DCDLs.

Table-3. Functionality of inverting and non-inverting MUX.

Terminal	Data	Control Bit	Inv.MUX Output	Non-Inv.MUX Output
0	1	0	0	1
1	1	1	0	1

4.1.1.2 Multiplexer based glitch avoidance circuit-non-inverting topology

The Figure-7b shows the transistor level implementation of a non-inverting MUX based delay line. This line requires {(3×4)+720}=732 transistors where 720 is the number of transistors required to implement an inverting topology of 10 delay cells and (3×4) is the arrangement of three NAND gates at the output of an inverting topology to make their configuration to be non-inverting. In multiplexers (21×12) =252 transistors are utilized. Table-3 explains the operation of the non-inverting multiplexer. Whenever the control bit is "0" the data at the 0^{th} input terminal of the MUX will be directed towards the output and during the control bit of “1” the data at the 1^{st} input terminal of the MUX will be directed towards the output. When In = 0 and control code c = 2, the Out = 0 since it is a non-inverting topology and has no glitches. Though only a single control bit is given as a select line to the multiplexer for controllability there are no glitches at the output.

The DCDL implementation for registers in Systolic Array filters of Figure-4 is shown below in Figure 9toeliminate the varying arrival times of the synchronous clock between the first flip flop and the last flip flop. Owing to the length of the interconnection, the propagation delay of the clock gets altered between the flip flops thereby causing a clock skew. Since the accuracy of the shift registers lies on the synchronization of the clock, the resultant of the phase shift would become detrimental due to an unwanted signal transition at the output of the last flip flop [25-27].

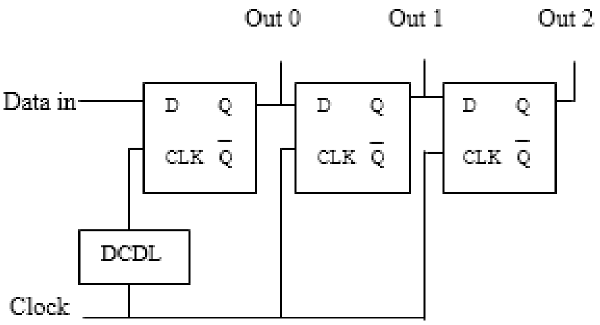


Figure-9. Block diagram depicting the noise removal process.

The delay simulation and FECG signal denoising using systolic filters with glitch avoidance are discussed in the forthcoming section.

5. SIMULATION RESULTS AND DISCUSSION

The effectiveness of QRS peak detection algorithms is assessed based on how accurately the observed peak is identified as noise under static conditions. During simulation, the feature extraction models use the following hypotheses to estimate the mean and variance from the sample data for each class. The total outcomes over a "100" test iterations are tabulated. To the typical data samples collected from MIT-BIH Arrhythmia, appropriate noise of a specific frequency range and duration is purposefully inserted for testing. Future classifier algorithms will use a 30-minute signal from the MIT-BIH Arrhythmia database as their input data. Then, it is split into segments that are each 10 seconds long. This means that there are 180 chunks in total each chunk being of 10s. The algorithm has a sliding window which reads 2 chunks at a time.

5.1 Validation of Classifiers Before and After Using TAODV as Distance Metric

In Table-4, the effectiveness of the LDA-based QRS peak detection method is examined for a variety of distance metrics, including Euclidian, Manhattan, Chebyshev, DVR, and TAODV. It is found that the SSA classifier performs with an overall average accuracy of 86.3% for Euclidian distance metrics, whereas it performs with an average accuracy of 95.0% for TAODV distance metrics. With performance accuracy of 2.9%, 2.9%, and 6.9%, respectively, Euclidian, Manhattan, and Chebyshev of CSO outperform SSA. Similar to how DVR and TAODV of SSA have superior performance accuracy than CSO with 4.1% and 2.6%, respectively. When it comes to categorizing the remaining noises such as those caused by muscle contraction, electrode contact, patient movement, and electrosurgical noise, the SSA classifier performs better than the other classifiers. Ten classifiers are implemented using TAODV, and to improve performance, consensus verification is done for the following classifiers: k-Nearest Neighbour (k-NN), Naive Bayes (NBC), Support Vector Machine (SVM), Ant Colony Optimization (ACO), Cockroach Swarm Optimization (CSO), Lizard Learning Algorithm (LLA), Particle Swarm Optimization (PSO), African Wild Dog (AWD), and Lion Optimization Algorithm (LOA).

The following tuning parameters are assigned for the best classification of the data tabulated: To improve the categorization of noise, the parameter k is supposed to be 15 via the trial-and-error method for the k-NN Classifier. The frequency table and likelihood table are used, as in NBC classification, to forecast the likelihood of a certain class for five different noise sources as the target over their properties. With the vibration velocity set to 1.5 and the weight assumed to be 1.0, the linear kernel function for SVM is decided to be 0.5. After creating a matrix of X rows of observation points and Y columns of predictor points with the value of 1.5 to create the hyperplane, the gradient is set to 0. After creating a matrix of X rows of observation points and Y columns of predictor points with the value of 1.5 to create the hyperplane, the gradient is set to 0. The size N in the instance of ACO is 20 over a weight of 1.5, and iterations were carried out 100 times with tuning parameters of 0.5 each and constant = 1.0, where r0 and q0 are given the value of 0.75. In the SSA, constant rand is set to 0.75 and the tuning parameters intensity are assigned values of 0.5 and 0.75, respectively. The configured value is 0.25. The population size for the CSO is set to N=200, and iterations are conducted about 500 times over the range of (-25, 25) incrementing with a step size of 1.5, initial weight "w=1.2," and vibration velocity of 0.5. Perception distance, step size, intensity, and speed are taken into account for tabulation along with the total number of test runs across 100 iterations. When propagating across the next best position, the hunger threshold is set to 0.5, and the probability regulating parameter "randint()" adjusts the dimension mask at every random walk step to 0.001. For the values of m=1.5 and constant = 1.0, the LLA algorithm is simulated. The propagation for the following-best position is 0.75 with a 0.1-step size. The range of movement or mobility is restricted to a region between (-25, 25). For PSO, a flock size of 20 is chosen with a variable k=0.5 and a velocity of 0.75. The weight is set at 0.25, and it is assumed that the constants c1 and c2 are, respectively, 0.5 and 0.75. The population size is set to 20 for the AWD classifier, and the tuning parameters Lxi and Uxi are assumed to be 0.5 and 0.75, respectively. The value of the constant c remains at 1.0. The flock's range is set between (-25, 25) with a 0.01 step size. The population size for the LOA classifier is set to 20 and the tuning parameters Lxi and Uxi are taken to be 0.5 and 0.75, respectively. A steady rand of 0.5 is maintained. The flock's range is set between (-25, 25) with a 0.01 step size. The Prey is set at 6.0 since the total number of attributes determines it. Hunt ratio equals 0.1.

Table-4. Performance of LDA based QRS Peak detection for classification with different distance metrics.

Noise Source	Distance Metric	Classifier									
		k-NN	NBC	SVM	ACO	LOA	CSO	LLA	PSO	AWD	SSA
Powerline	Euclidian	61.8	66.8	78.1	83.1	74.3	88.1	75.6	76.8	86.8	85.6
	Manhattan	63.1	70.6	80.6	84.3	74.3	89.3	78.1	76.8	88.1	86.8
	Chebyshev	64.3	71.8	81.8	84.3	75.6	91.8	80.6	76.8	90.6	85.6
	DVR	65.6	83.1	88.1	90.6	86.8	91.8	86.8	85.6	93.1	95.6
	TAODV	66.8	88.1	91.8	93.1	89.3	94.3	91.8	90.6	95.6	96.8
Muscle Contraction	Euclidian	63.1	90.6	76.8	81.8	74.3	83.1	71.8	86.8	80.6	93.1
	Manhattan	64.3	71.8	78.1	83.1	75.6	84.3	75.6	88.1	81.8	93.1
	Chebyshev	66.8	74.3	80.6	84.3	76.8	85.6	75.6	88.1	83.1	95.6
	DVR	70.6	80.6	84.3	86.8	85.6	88.1	83.1	90.6	85.6	96.8
	TAODV	71.8	81.8	85.6	89.3	84.3	90.6	83.1	93.1	86.8	96.8
Electrode Contact	Euclidian	61.8	71.8	88.1	84.3	86.8	89.3	75.6	81.8	78.1	91.8
	Manhattan	63.1	74.3	80.6	88.1	78.1	83.1	90.6	85.6	89.3	93.1
	Chebyshev	64.3	76.8	83.1	89.3	80.6	84.3	91.8	86.8	90.6	93.1
	DVR	66.8	80.6	85.6	90.6	84.3	86.8	93.1	88.1	91.8	95.6
	TAODV	70.6	81.8	86.8	93.1	84.3	90.6	95.6	91.8	94.3	96.8
Patient Movement	Euclidian	61.8	70.6	75.3	80.6	83.1	86.8	81.8	76.8	88.1	91.8
	Manhattan	63.1	75.6	76.8	83.1	85.6	86.8	81.8	79.3	89.3	93.1
	Chebyshev	64.3	76.8	81.8	85.6	88.1	89.3	83.1	81.8	90.6	93.1
	DVR	64.3	78.1	85.6	89.3	90.6	91.8	88.1	86.8	93.1	94.3
	TAODV	65.6	83.1	86.8	91.8	94.3	93.1	90.6	89.3	95.6	96.8
Electrosurgical	Euclidian	61.8	71.8	88.1	84.3	86.8	89.3	75.6	81.8	78.1	91.8
	Manhattan	63.1	74.3	89.3	85.6	88.1	90.6	76.8	84.3	79.3	93.1
	Chebyshev	64.3	76.8	90.6	86.8	88.1	91.8	78.1	85.6	79.3	93.1
	DVR	64.3	83.1	91.8	88.1	90.6	93.1	84.3	86.8	85.6	95.6
	TAODV	65.6	88.1	94.3	91.8	93.1	95.6	90.6	89.3	90.6	96.8

Using MATLAB 2013b, a variety of classifier metrics, including Accuracy (Acc.), Error Rate (Er.r.), Sensitivity (Sens.), specificity (Spec.), false positive rate (FPR), precision (prec.), F1 score (f1), Mathews correlation coefficient (MCC), Jacard Metric (JM), and Balanced Classification Rate (BCR), are analyzed with the collected samples. The results are tabulated in tables 5 and

In Table-5, it can be seen that the SSA outperforms the simple k-NN classifier in classifying Powerline noise with accuracy and sensitivity values of 86.3% and 87.5%, respectively.

Table-5. Performance metric of classifiers with Euclidean as distance metric for classifying powerline noise.

Classifier	Performance Metrics									
	Acc.	Er.R.	Sens.	Spec.	FPR	Prec.	F1	MCC	JM	BCR
k-NN	61.8	41.8	59.3	64.3	39.3	62.3	60.8	21.8	43.6	61.8
NBC	66.8	36.8	64.3	69.3	34.3	67.6	65.9	31.8	49.0	66.8
SVM	78.1	25.6	79.3	76.8	26.8	77.4	78.3	54.3	63.8	78.1
ACO	83.1	20.6	81.8	84.3	19.3	83.9	82.8	64.3	69.9	83.1
SSA	88.1	15.6	89.3	86.8	16.8	87.2	88.2	74.3	77.9	88.1
CSO	74.3	29.3	74.3	74.3	29.3	74.3	74.3	46.8	58.7	74.3
LLA	75.6	28.1	74.3	76.8	26.8	76.2	75.2	49.3	59.8	75.6
PSO	76.8	26.8	74.3	79.3	24.3	78.1	76.2	51.9	61.0	76.8
AWD	86.8	16.8	86.8	86.8	16.8	86.8	86.8	71.8	75.7	86.8
LOA	85.6	18.1	89.3	81.8	21.8	83.2	86.1	69.5	74.7	85.6

Table-6 shows that the SSA performs well, with accuracy and sensitivity values of 95.0% and 95.0%, respectively, which are 30.0% and 32.5% higher than the simple k-NN.

After the findings were validated, it was discovered that the SSA-based classifier performed better than the other classifiers regardless of the noise source. Additionally, the samples are purposely introduced to the original data together with noise frequency components to assess the classifier's propensity to recognize FP and FN.

Table-6. Performance metric of classifiers with TAODV as distance metric for classifying powerline noise.

Classifier	Performance Metrics									
	Acc.	Er.R.	Sens.	Spec.	FPR	Prec.	F1	MCC	JM	BCR
k-NN	65.6	38.1	61.8	69.3	34.3	66.7	64.1	29.4	47.1	65.6
NBC	83.1	20.6	81.8	84.3	19.3	83.9	82.8	64.3	69.9	83.1
SVM	90.6	13.1	91.8	89.3	14.3	89.6	90.7	79.3	81.8	90.6
ACO	91.8	11.8	89.3	94.3	9.3	93.9	91.5	81.9	83.2	91.8
SSA	96.8	6.8	96.8	96.8	6.8	96.8	96.8	91.8	92.3	96.8
CSO	95.6	8.1	96.8	94.3	9.3	94.5	95.6	89.3	90.2	95.6
LLA	89.3	14.3	89.3	89.3	14.3	89.3	89.3	76.8	79.6	89.3
PSO	86.8	16.8	86.8	86.8	16.8	86.8	86.8	71.8	75.7	86.8
AWD	93.1	10.6	94.3	91.8	41.8	92.0	93.2	84.3	85.9	93.1
LOA	94.3	9.3	96.8	91.8	11.8	92.3	94.5	86.9	88.2	94.3

SSA, LOA, CSO, AWD, ACO, SVM, LLA, PSO, NBC, and k-NN are in the hierarchy of classifiers depending on how well they perform at classifying powerline noises. The results of similar experiments conducted on the remaining noises, including muscle contraction, electrode contact, patient movement, and electrosurgical noise, were consistent.

5.1.1 VLSI Implementation of classifiers

After exporting the algorithm created in MATLAB 2013b to Xilinx ISE 9.1i, Table-7 shows the design metrics for each classifier. Due to its quick convergence and high global optima solution, the SSA classifier has taken up less space and time than the other classifiers. When compared to k-NN, the area and simulation time have decreased by 8.5% and 46.8%, respectively.

Table-7. VLSI Design metrics of classifiers.

Classifier	Register	LUT	Flip Flop LUT Pairs	Bonded IOB	Fan out of non-clock nets	Time (ns)
k-NN	406	2874	1694	53	3.7	4.7
NB	394	2980	1679	49	3.7	4.7
SVM	289	2608	1432	32	3.3	2.8
ACO	343	2670	1527	45	3.4	3.4
CSO	301	2643	1498	39	3.3	3.1
LOA	267	2589	1390	29	3.2	2.7
LLA	384	2733	1702	43	3.6	4.4
PSO	365	2743	1566	37	3.4	3.8
AWD	377	2767	1535	44	3.6	4.2
SSA	245	2458	1349	27	3.1	2.5

As a result, it can be seen from the simulation results that, under the condition that LDA is employed as the algorithm for detecting the QRS peak and TAODV as the distance measurement metric, the SSA classifier has performed well with the best accuracy and run time. Using the suggested TOADV, the classification area and time have improved over k-NN by 22.1% and 46.7%, respectively. After classification, the upcoming FIR filter denoising options are addressed.

5.2 Validation of Glitch Avoidance Circuit Using MUX

The Figure-10 presents the delay report of existing NAND and MUX based DCDL. The delay is the measure of the input pulse to that of the output pulse concerning time as calculated for Figures 7a and 7b. The simulation time differs from the delay in such a way that it is the time taken for the execution of the entire DCDL constituted of 10 delay elements using Cadence Virtuoso. The delay varies as the control code increases from 1 to 7. Depending on the application or the duration up to which the clock skew has to be eliminated, the control code can be chosen. The advantages of both the existing NAND and Multiplexer based DCDLs are in setting up a variable delay and also the above table indicates that they provide the same delay.

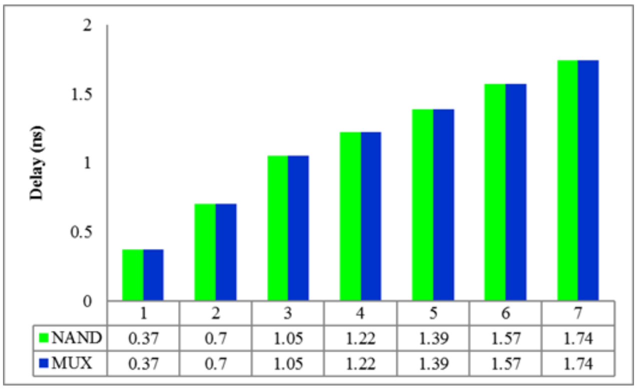


Figure-10. Delay Report of existing NAND and MUX based DCDL.

The CMOS implementation of the existing glitch free NAND based DCDLs reveals that the same signal is shared between two transistors which could be replaced by a single transistor without any deviation in its operation at the same time without any glitches. In the case of multiplexers, three transistors are removed even without which there is no hindrance in the usual operation of the MUX based DCDLs. The following results tabulated in Table-8 are simulated by assigning the value of control code to be 10.From the result tabulated, it is evident that the proposed architecture requires only less number of transistors and hence occupies less area and contributes less delay.

The results in Table-9 depict the evaluation of performance metrics of filters such as area, power, delay, glitches and efficiency with varied multipliers: Array, Wallace Tree, Baugh Wooleyand Booth implemented without Glitch avoidance circuit under Chebyshev window function. The area and power of HPF constructed using Booth multiplier is better than the existing array multiplier by 49.5% and 18.2% respectively whereas the efficiency of AMF constructed using array multiplier is better than the Booth multiplier by 21.9% since the glitch rate is very high at an extent of 39.03%.

Table-8. Analysis report of proposed architecture for DCDLs.

Parameters	**Proposed Architecture for DCDLs**			
	NAND		**MUX**	
	Inverting	**Non-Inverting**	**Inverting**	**Non-Inverting**
Power (watts)	1.09e-002	1.55e-002	4.53e-003	2.99e-003
Active Devices	160	172	252	216
Simulation Time (sec)	2.38	2.48	3.66	2.52

Table-9. Simulation results of systolic array filter with 8 bit multipliers using Chebyshev Window Function (without glitch free circuit).

Filters / **Multiplier**	**Performance Metrics**	**LPF**	**HPF**	**MBF**	**UCF**	**ABF**	**AMF**
Array-multiplier	Area (LUTs × 10^2)	4.05	3.76	3.81	3.84	3.94	4.1
	Power × 10^2(mW)	6.15	5.98	6.16	6.29	6.13	6.17
	Delay × 10^4(ns)	2.21	1.85	1.7	1.55	1.39	1.29
	Glitches/rate (%)	42.12	41.73	39.93	39.13	38.03	37.53
	Efficiency (%)	58.88	59.27	60.07	60.87	61.97	62.47
Wallace –tree	Area (LUTs × 10^2)	3.42	3.1	3.35	3.57	3.76	3.98
	Power × 10^2(mW)	6.14	5.88	5.96	6.04	6.06	6.12
	Delay × 10^4(ns)	2.12	1.67	1.56	1.5	1.27	1.2
	Glitches/rate (%)	36.65	35.93	34.88	34.53	34.23	33.03
	Efficiency (%)	63.35	64.07	65.12	65.47	65.77	66.97
Baugh –Wooley	Area (LUTs x 10^2)	3.18	2.84	3.1	3.34	3.55	3.8
	Power × 10^2(mW)	5.77	5.53	5.68	5.71	5.79	5.89
	Delay × 10^4(ns)	1.94	1.52	1.3	1.23	1.15	1.12
	Glitches/rate (%)	29.35	28.93	29.43	29.55	29.57	29.73
	Efficiency (%)	70.65	71.07	70.57	70.45	70.43	70.27
Booth –multiplier	Area (LUTs × 10^2)	2.45	2.18	2.24	2.36	2.29	2.34
	Power × 10^2(mW)	5.65	5.28	5.44	5.49	5.71	5.79
	Delay × 10^4(ns)	1.69	1.35	1.23	1.21	1.07	0.95
	Glitches/rate (%)	28.85	27.93	27.03	25.63	24.03	21.93
	Efficiency (%)	71.15	72.07	72.97	74.37	75.97	78.07

The results in Table-10 depicts the evaluation of performance metrics of filters such as area, power, delay, glitches and efficiency with varied multipliers: Array, Wallace Tree, Baugh Wooley, and Booth implemented with proposed MUX based Glitch avoidance circuit under Chebyshev window function. The area and power of LPF constructed using the Booth multiplier is better than the array multiplier by 48.3% and 8.0% respectively. Similarly, the efficiency of AMF constructed using Booth multiplier is better than array multiplier by 2.9% since the glitch rate has reduced to 0.01% after using the glitch avoidance circuit.

Table-10. Simulation results of filters with 8 bit multipliers using Chebyshev Window Function (with MUX based glitch avoidance circuit).

Filters / Multiplier	Performance Metrics	LPF	HPF	MBF	UCF	ABF	AMF
Array –multiplier	Area (LUTs × 10^2)	3.62	3.67	3.72	3.75	3.85	4.01
	Power × 10^2(mW)	4.86	4.94	5.12	5.25	5.09	5.13
	Delay × 10^4(ns)	1.47	1.45	1.3	1.15	0.99	0.89
	Glitches/rate (%)	10.6	8.55	8.5	7.19	7.67	7.93
	Efficiency (%)	89.4	91.45	91.5	92.81	92.33	92.07
Wallace –tree	Area (LUTs × 10^2)	3.12	3.14	3.39	3.61	3.8	4.02
	Power × 10^2(mW)	4.84	4.84	4.92	5	5.02	5.08
	Delay × 10^4(ns)	1.38	1.27	1.16	1.1	0.87	0.8
	Glitches/rate (%)	9.17	7.5	6.45	5.23	5.14	6.6
	Efficiency (%)	90.83	92.5	93.55	94.77	94.86	93.4
Baugh –Wooley	Area (LUTs × 10^2)	2.88	2.88	3.14	3.38	3.59	3.84
	Power × 10^2(mW)	4.67	4.69	4.84	4.87	4.95	5.05
	Delay × 10^4(ns)	1.2	1.12	0.9	0.83	0.75	0.72
	Glitches/rate (%)	8.87	5.97	5	4.1	5.1	5.37
	Efficiency (%)	91.13	94.03	95	95.9	94.9	94.63
Booth –multiplier	Area (LUTs × 10^2)	2.15	2.22	2.28	2.4	2.33	2.38
	Power × 10^2(mW)	4.55	4.44	4.6	4.65	4.87	4.95
	Delay × 10^4(ns)	0.95	0.95	0.83	0.81	0.67	0.55
	Glitches/rate (%)	7.97	5.5	3.98	3.54	4.08	3.87
	Efficiency (%)	92.03	94.5	96.02	96.06	95.92	96.13

The LMS algorithm makes sure that the rapid shifts and alterations in the signal are simply removed and eliminated. MBF outperforms LPF in terms of performance, even when the signal is at multiple frequencies with minimal data loss. The ECG signal's peaks may be readily extracted from the distorted signal, providing accurate information on heart rate. Figure 11 depicts an AMF with a regular arrangement of linear filters in a pattern with a transfer function and variable control parameters that are modified using the LMS optimization algorithm for the intended use. Pass frequencies of 7200 and 14400 Hz and stop frequencies of 12000 and 9600 Hz are used to monitor the outcome.

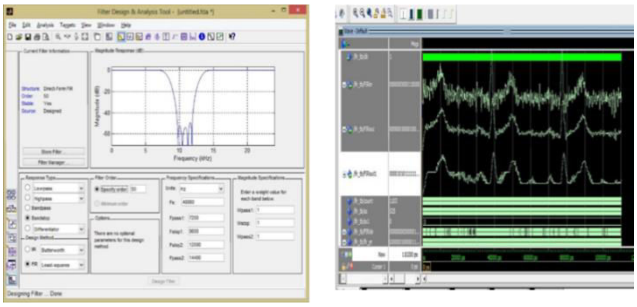


Figure-11. FECG Noise filtering using AMF filter using LMS algorithm.

As two levels are considered for the filter, the coefficients of the input signal designed using the LMS algorithm are compared against the input signal whenever the clock rises. The resultant output signal has filtration of noise at these two levels thereby improving the efficiency. To elaborate on the process, the first level of filtration output repeatedly undergoes noise removal until the 'n' iteration. As the number of levels increases, efficiency is also improved. Hence the number of filtration levels and pass bands is directly proportional to the efficiency of the filter.

6. CONCLUSIONS

By applying a common transformation, such as the Linear Discriminant Analysis Method, the R peaks in the QRS complex can be detected. The classification of FECG sounds is then decided to be an abstract segregation procedure based on the length of the signal in addition to the frequency and amplitude. A tweak to the current classifier in terms of distance measurement is suggested to enhance the performance of the SSA classifier. To facilitate the characteristics that are retrieved for noise classification, the Trusted Ad-Hoc On-Demand Distance Vector routing algorithm (TAODV) is utilised to evaluate the trust of its neighbours with an established threshold. Based on classification accuracy, the performance of the QRS peak detection algorithms and classifiers is assessed concurrently. It has been noted that the distance measurement accuracy of 95.0% for the QRS peak detection based on LDA and TAODV is the best and most reliable. In comparison to k-NN, the SSA classifier's classification area and time have improved by 22.1% and 46.7%, respectively, while utilising the suggested TOADV.

Systolic array filters, which are the inertial component affecting the dynamics of the system, are verified for their effectiveness in filtering out noise. Examples include adaptive band stop, low pass, multiband, unsigned coefficient, and multiband. It is frequently noted that while switching from an array multiplier to a booth multiplier, the area has significantly decreased among the various multipliers used for filter construction. But it's important to remember that when the input bit rate is gradually increased, the number of glitches-and therefore the power also rising. Glitch-free NAND and MUX based filters are presented and simulated for all the performance metrics such as area, power, delay, and efficiency to avoid the increase in glitch rate and subsequently the power. With the suggested MUX-based glitch avoidance circuit, the performance of the Booth multiplier in the adaptive multiband filter has enhanced efficiency of 92.03%, 94.5%, 96.02%, 96.46%, 95.92%, and 96.13%.

REFERENCES

[1] Huliyurdurga Srinivasa Setty Natraj Setty, Shivanand Sanganna Gouda Patil, Raghu Thagachagere Ramegowda, Vijaykumar, Iswarappa Balekundre Vijayalakshmi and Cholenahalli Nanjappa Manjunath. 2017. Comprehensive Approach to Congenital Heart Defects. Journal of Cardiovascular Disease Research. 8(1): 1-5.

[2] Maria G. Signorini, Andrea Fanelli and Giovanni Magenes. 2014. Monitoring Fetal Heart Rate during Pregnancy: Contributions from Advanced Signal Processing and Wearable Technology. Computational and Mathematical models in Medicine, doi: 10.1155/2014/707581.

[3] Matawalli Ajagana Geidam, Rajesh Prasad and Ibrahim A Bello. 2014. Child and Maternal Health Care using Telemedicine: A case study of Yobe State, Nigeria. International Journal of Computer Engineering and Applications. VIII(1): 1-11.

[4] Alvarez L. O. S., Ordónez Y. N. F., Salvador A. G. and Roig J. M. 2013. Noninvasive FECG for estimating the fetal heart rate. Symposium of Signals, Images and Artificial Vision, pp. 1-4, doi: 10.1109/STSIVA.2013.6644934.

[5] Ertugrul D. Ç., Kanmaz H., Yüksel M. U., Elçi A. and Ertugrul M. 2016. Fetal Heart Rate Monitoring System. IEEE 40th Annual Computer Software and Applications Conference, pp. 65-70, doi: 10.1109/ COMPSAC.2016.116.

[6] Assaleh K. 2006. Adaptive Neuro-Fuzzy Inference Systems for Extracting Fetal Electrocardiogram. IEEE International Symposium on Signal Processing and Information Technology, pp. 122-126, doi: 10.1109/ISSPIT.2006.270782.

[7] John S. T., Goyal M. and Singh S. 2017. Ambulatory fetal heart monitoring with QRS detection employing independent component analysis. IEEE Region 10 Conference, doi: 10.1109/ TENCONSpring.2017.8070101.

[8] Alberto Luque-Chang, Erik Cuevas, Fernando Fausto, Daniel Zaldívar and Marco Pérez. 2018. Social Spider Optimization Algorithm: Modifications, Applications, and Perspectives. Mathematical Problems in Engineering. 8: 1-29.

[9] Indu Sainia, Dilbag Singh and Arun Khosla. 2013. QRS detection using K-Nearest Neighbor algorithm (KNN) and evaluation on standard ECG databases. Journal of Advanced Research. 4(4): 331-344.

[10] James J. Q. Yua and Victor O. K. Lia. 2015. A Social Spider Algorithm for Global Optimization. Applied Soft Computing. 1: 1-18.

[11] Menaka Pushpa A. 2010. Trust based secure routing in AODV routing protocol. IEEE International Conference on Internet Multimedia Services Architecture and Applications. pp. 978-983.

[12] Bisengar Ahmed, Zytoune Ouadoudi, Rziza Mohamed and Ouadou Mohamed. 2013. Enhancements in AODV routing using velocity and

distance. Second International Conference on the Innovative Computing Technology. pp. 107-110.

[13] Burke I. D., van Heerden R. and Olivier M. S. 2011. Analysing the fairness of trust-based Mobile Ad hoc Network protocols: Comparing the fairness of AODV and TAODV protocols in scenario driven simulations. Information Security for South Africa, pp. 1-8, doi: 10.1109/ISSA.2011.6027528.

[14] Andrea Bonetti, Adam Teman, Philippe Flatresse and Andreas Bur. 2017. Multipliers-Driven Perturbation of Coeffcients for Low-Power Operation in Reconfigurable FIR Filters. IEEE Transactions on Circuits and Systems I. 64(9): 2388-2400.

[15] Pramodini Mohanty. 2012. An Efficient Baugh-Wooley Architecture for Signed and Unsigned Fast Multiplication. International Journal of Computer Science & Engineering Technology. 3(4): 94-99.

[16] Purvi Jain, Ankur Dixit and Sunil Kumar Mangal. 2016. De-noising & Processing of ECG Signals by using Adaptive Filter. International Journal of Research and Scientific Innovation. 3(5): 1-12.

[17] Gibson N. M., Peasgood W., Herbert J. M., Woolfson M. S. and Crowe J. A. 1995. Iterative method to derive an approximate matched filter template for fetal electrocardiogram signals. Medical Engineering and Physics. 17(3): 188-196.

[18] Jiajia Chen, Chip-Hong Chang, Feng Feng, Weiao Ding and Jiatao Ding. 2015. Novel Design Algorithm for Low Complexity Programmable FIR Filters based on Extended Double Base Number System. IEEE Transactions on Circuits and Systems I. 62(1): 224-233.

[19] Chen P. L., Chung C. C., Yang J. N. and Lee C. Y. 2006. A clock generator with cascaded dynamic frequency counting loops for wide multiplication range applications. IEEE Journal on Solid-State Circuits. 41(6): 1275-1285.

[20] Eriksson H., Larsson-Edefors P., Sheeran M., Själander M., Johansson D. and Schölin M. 2006. Multiplier reduction tree with logarithmic logic depth and regular connectivity. Proceedings in IEEE International Symposium on Circuits and Systems, pp. 4.-8, doi: 10.1109/ ISCAS.2006.1692508.

[21] Davide De Caro. 2013. Glitch-Free NAND-Based Digitally Controlled Delay-Lines. IEEE Transactions on Very Large Scale Integration (VLSI) Systems. 21(1): 55-66.

[22] Davide De Caro, Carlo Alberto Romani, Nicola Petra, Antonio Giuseppe Maria Strollo and Claudio Parrella. 2010. A 1.27 GHz, All-Digital Spread Spectrum Clock Generator/Synthesizer in 65 nm CMOS. IEEE Journal of Solid-State Circuits. 45(5): 1048-1060.

[23] Dheyaa Alhelal and Miad Faezipour. 2017. Denoising and Beat Detection of ECG Signal by using FPGA. International Journal of High Speed Electronics and Systems, 26(03), doi: 10.1142/S012915641740016X.

[24] Guansheng Li, Yahya M. Tousi, Arjang Hassibi and Ehsan Afshari. 2009. Delay- Line- Based Analog-to-Digital Converters. IEEE Transactions on Circuits and Systems-II: Express Briefs. 56(6): 464-468.

[25] Jean Barbier and Montpellier. 2010. Programmable Delay Line Circuit with Glitch Avoidance. Patent published.

[26] Maamoun M. and Iaeng S. Arami. 2017. A 3ps Resolution Time-to-digital Converter in Low-cost FPGA for Laser Rangefinder. Proceedings of the World C1ongress on Engineering. I: 4-9.

[27] Mohammad Maymandi-Nejad and Manoj Sachdev. 2003. A Digitally Programmable Delay Element: Design and Analysis. IEEE Transaction on VLSI. 11(5).