International Journal of Engineering & Technology, 9 (2) (2020) 367-377

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Design and implementation of accurate frequency estimator depend on deep learning

Mohamed Saber [1]*, El-Sayed M. El-kenawy [2]

[1] *Dep. Of Communications & Computers, Faculty of Engineering, Delta University for Science & Technology Gamasa city, Mansoura, Egypt*
[2] *Department of Communications and Electronics Delta Higher Institute for Engineering & Technology (DHIET), Mansoura, Egypt*
**Corresponding author E-mail: Mohamed.saber@deltauniv.edu.eg*

Abstract

An Accurate, efficient, and stable system to estimate the unknown input frequency of a sinusoidal signal is presented. The proposed design solves the main drawback of the existing phase-based estimator which called a derivative estimator depend on deep learning. These limitations are the inability to estimate low frequencies and the large estimation errors for the frequencies near the Nyquist rate. A Brief mathematical analysis in discrete-time of the proposed system is presented. Proposed estimator performance when the input is a single sinusoid, multiple sinusoids in the presence of additive white Gaussian noise (AWGN) are provided. The accuracy of the proposed estimator is the result of dividing the dynamic range of estimation to three regions (low frequencies, middle frequencies, high frequencies) and specify a different formula to calculate the estimated frequency in each region. The boundaries of each region are determined by using a Grey wolf optimizer (GWO) which training bidirectional recurrent neural networks (BRNN) to select the best weights for the estimated frequency. The simulation results ensure the accuracy and validity of the proposed estimator compared to the traditional one. The hardware implementation of enhanced estimator using field-programmable gate array (FPGA), consumed 265 mW, and worked at 375 MHz.

Keywords: *Frequency Estimation; Phase-Based Estimator; Deep Learning; FPGA; Neural Networks; GWO.*

1. Introduction

Estimating the unknown frequency of sinusoidal signals in noisy environments is an important task in signal processing, seismic, radar, sonar, communications, instrumentation, biomedical, and other applications [1-5]. Many algorithms have been proposed to estimate frequency. One common algorithm is a maximum likelihood estimator (MLE), which estimates the frequency by obtaining the maximum of the periodogram for the input signal using fast Fourier transform (FFT). MLE achieves the Cramer-Rao lower bound at low/moderate signal-to-noise ratio [6]. The main drawback of the MLE estimator is that implementation requires extensive computations, which need extra hardware and leads to high power consumption. Many improvements are presented to the MLE algorithm to reduce the cost of implementation [7-10].
Another algorithm to estimate frequency uses the phase of FFT to estimate the frequency. Weighted linear predictor (WLP), and weighted phase average (WPA), are two fast frequency estimators based on cross-correlation and phase of the input signal [11]. However, they suffer from low performance in case of low SNR. Another method that uses the derivative of the input signal to estimate the frequency is proposed in [12]. The limitation of the derivative algorithm is the inability to estimate the low frequency and the accuracy of estimation decrease for the frequencies near the Nyquist rate. Frequency offset in communication circuits, and seismic application required the accurate estimations for low frequency.
Deep learning nowadays has many applications such as medical diagnoses [13], information security [14], distributing graphic rendering [15], e-market trust measurement [16], [17], and signal and image processing [18], [19]. In optimization, the best optimal solution is a main goal , according to the problem description how to select the available solutions from a given problem .This paper proposed an accurate frequency estimator for all dynamic range of input frequencies (0: half the sampling frequency) based on derivative method and grey wolf optimization technique which used for training bidirectional neural network to select the best weights which represents the boundaries of low frequencies range, middle frequencies, and high frequencies ranges.
This paper is organized as follows; Section 2 explains the grey wolf optimization technique. Basics of derivative frequency estimator present in section 3. Section 4 introduces the mathematical model analysis for the proposed method in the discrete-time domain, the analysis in the case of white additive gaussian noise (AWGN) is also provided with tracking the input frequency. Section 5 provides the simulations of the proposed estimator. Finally, section 6 presents the FPGA implementation of the propped estimator.

2. Grey wolf optimizer (GWO)

Grey wolves are one of the zenith predators, and zenith predators are at the highest point of the natural pecking order. In the real live Grey wolves commonly bring closer in groups. The average of group numbers may be from 5 to 12. The fascinating thing about grey wolves is that they have an extremely severe social driving pecking order. The leaders called alphas, are a male and a female. The alpha is responsible for settling on significant choices to the group such as hunting, resting place, etc. The whole group should obey the alpha's orders [20]. The alpha wolf is additionally called the leader wolf since the whole group should ought to comply with the alpha's requests. Beta is the second level in the hierarchy of grey wolves. Beta is the second level in the chain of command of dim wolves. The betas are second rate wolves that are liable for helping the alpha in settling on the correct choices or other gathering exercises. Omega is the least positioning of the grey wolf. The omega assumes the job of a substitute. mega wolves consistently need to capitulate to all other predominant wolves. Also, they are the last wolves that are permitted to eat. As shown in Fig. 1.

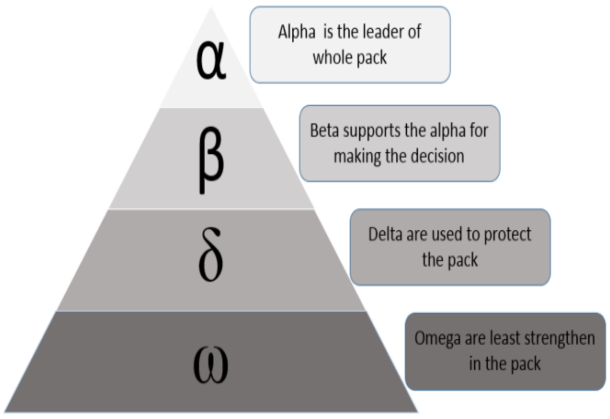
α
Alpha is the leader of whole pack
β
Beta supports the alpha for making the decision
δ
Delta are used to protect the pack
ω
Omega are least strengthen in the pack

Fig. 1: Hierarchy of Gray Wolves.

The position of each wolf is updated using the following equations:

$$\vec{D} = \left|\vec{C}\,.\,\overrightarrow{X_p}\,(t) - \vec{X}\,(t)\right| \qquad (1)$$

$$\vec{X}\,(t+1) = \overrightarrow{X_p}\,(t) - \vec{A}\,.\,\vec{D}$$

Where t refers to the current iteration, $\vec{A}$ and $\vec{C}$ are coefficient vectors, $\overrightarrow{X_p}$ is the preposition, and $\vec{X}$ is the position of the gray wolf. The vectors are calculated using the following equation:

$$\vec{A} = 2\vec{s}\,.\,\overrightarrow{r_1} - \vec{s} \qquad (2)$$

$$\vec{C} = 2.\overrightarrow{r_2}$$

Where, s is linearly decreased from 2 to 0 over the number of iterations, and r_1 and r_2 are random vectors in [0, 1]. Grey wolf optimization Flowchart shown in Fig. 2.

Initialize the grey wolf population Xi (i = 1, 2, ..., n)
Calculate α, β, and δ
Calculate a as per Eq (1)
for each search agent
Update A, and C
Calculate Xα, Xβ, and Xδ
Calculate fitness
Update A, a, C
Evaluate all particles using the objective function
Update the positions of the three best agents α;β;δ
t=t+1
return α

Fig. 2: Gray Wolf Optimization Flowchart.

2.1. Deep learning

One of the maximum promising answers for big data issues is deep learning due to its big functionality of extracting the meaningful facts from massive information sets. Nowadays, Deep learning to know have been carried out to many fields such as machine learning, computer vision and ect. Where they have produced notable effects in comparison to traditional device studying techniques. Artificial Neural Networks (ANNs) are emulating the way of data processing and verbal exchange distributed nodes in organic worried systems. ANNs especially consists of a set of linked units known as synthetic neurons, Neurons usually have a state, and it's far typically represented by actual numbers between 0 and 1. The electricity of the signal that neurons ship downstream may be accelerated or decreased, and it's far especially depending on the weights of the neurons and synapses, which varies as getting to know proceeds. Neurons are typically prepared in layers;

the form of input processing operations finished in every layer can range between different layers. Signals are traveling from the first layer (input) to the closing layer (output), sometimes, its travers the layers a couple of times.[21] The ANNs performance is hugely tormented by the learning process and parameters optimization. Multilayer perceptron (MLP) is one of the most typically implemented ANNs. ANN specifically suffer from three primary issues: its convergence is very slow, it's far quite dependent on initial values, and it could be trapped without problems in local optima. Local optima can be defined as the satisfactory solution within a collection of neighbouring solutions, which can be defined as the most beneficial answer amongst all possible solutions inside the whole to overcome all of those the (BRNN) has been used.

2.2. Bidirectional recurrent neural networks (BRNN)

GWO is utilized to improve the presentation of the BRNN classifier and dispose of the excess and immaterial highlights. BRNN is chosen dependent on its presentation in the past works, which has demonstrated to outflank another AI classifier. Moreover, to give solid outcomes and maintain a strategic distance from overfitting issues (BRNN) can be considered as an extension of the unidirectional recurrent neural networks by adding a second hidden layer, where the connections between the hidden to hidden layers are in the opp site direction. Consequently, this model can exploit data from both directions, the past (u) and the future (y). The output u_t can be calculated by

$$Q(u_t|\{v_i\}_{i\neq t}) = \sigma(W_u^y h_t^y + W_u^z h_t^z + z_u) \tag{3}$$

Where

$$h_t^y = \tanh\left(W_h^y h_{t-1}^y + W_x^y x_t + z_h^y\right)$$

$$h_t^z = \tanh(W_h^z h_{t+1}^z + W_v^z v_t + z_h^z)$$

Wu were the weights that connects the hidden layer to output layer, Wh were the weights that connects the hidden layer to hidden layer and Wv were the weights that connects the input layer to the hidden layer. Zu were the biases of the output layer, and Zh were the biases of the hidden layer. For the final nonlinearity σ we can use sigmoid, tanh, and Relu as an activation function due to this structure, the RNN will calculate the output ut based on the transmitted information through the hidden layers regardless of whether it depends directly or indirectly on the values $\{v_i\}_{i=1}^t = \{v_1, \dots \dots .., v_t\}$.

2.3. BRNN parameter optimization

Weights in a BRNN have two main roles; the first one is, deciding how many the output is affected by the input, and the second one is, controlling the learning rate of the hidden layers. Exactly as the slope in linear regression, where the output is calculated by multiplying the weights to the inputs, then added up. Weights are numerical values that control how many neurons are affecting each other. For any neuron, if the inputs are V_1, V_2, and V_3, and weights applied to them are w_1, w_2, and w_3. The output is:

$$u = f(v) = \sum_{j=1}^{n} v_j w_j \tag{4}$$

Where n is the number of inputs. Generally, the weighted sum can be calculated by performing this array multiplication. Bias is an additional variable that can be used to adjusting the output along with the weighted sum of the inputs to the neuron. The final output of a neuron is:

$$u = f(v) = \sum_{j=1}^{n} v_j w_j + z \tag{5}$$

Where z is the bias.

3. Proposed estimator

In the beginning, we will explain how to extract the frequency of the input sinusoidal signal to be in the amplitude of the signal using a derivative estimator in the discrete-time domain. Let

$$x(nT_s) = a\, cos\,(2\pi f n T_s + \theta) \tag{6}$$

Where$a: amplitude, f: frequency, f_s: sampling\ frequency\ (Hz), T_s: sampling\ time\ (s), \theta: phase.$
Let us define a delayed sample of $x(n)$ which will be

$$x((n-1)T_s) = a\, cos\,(2\pi f(n-1)T_s + \theta) \tag{7}$$

The difference between the two signals using trigonometric identities will be

$$x_1(nT_S) = x(nT_S) - x\left((n-1)T_S\right) = \ a\, cos(2\pi f n T_S + \theta) - a\, cos(2\pi f(n-1)T_s + \theta)$$

$$= a\, cos(2\pi f T_S + \theta) - a\, cos(2\pi f T_s + \theta)\, cos(2\pi f T_S) - a\, sin(2\pi f n T_S + \theta)\, sin\,(2\pi f T_S)$$

$$= a\, cos(2\pi f n T_S + \theta)\,[1 - cos(2\pi f T_S)] - a\, sin(2\pi f n T_S + \theta)\, sin\,(2\pi f T_S)$$

$$= 2a\, cos(2\pi f n T_s + \theta)\, sin^2(\pi f T_s) - 2a\, sin(2\pi f n T_S + \theta)\, sin(\pi f T_S)\, cos\,(\pi f T_s)$$

$$= 2a\, sin(\pi f T_S)\, [cos(2\pi f n T_S + \theta)\, sin(\pi f T_s) - sin(\pi f T_s)\, cos(\pi f T_S)]$$

$$= 2a\, sin(\pi f T_S)\, sin\, (\pi f T_S - 2\pi f n T_S - \theta) \qquad (8)$$

Let the amplitude of the difference signal in Eqn. (8) be $b = 2a\, sin\, (\pi f T_S\,)$, so we can get

$$\frac{b}{2a} = sin\, (\pi f T_S) \qquad (9)$$

Now we can estimate the input frequency to be

$$\hat{f} = \frac{f_s}{\pi} sin^{-1}\left(\frac{b}{2a}\right) \qquad (10)$$

Eqn. (10) describes the conventional derivative estimator output for all the dynamic range $(0{:}\frac{f_s}{2})$. Unfortunately, there are two limitations in the method:

1) Can't estimate low frequencies (near zero) correctly, this limits the operation of this estimator in many applications such as frequency offset in communication and seismic systems.
2) Estimation error increased when the input frequency is near to half sampling frequency $(\frac{fs}{2})$.

The proposed estimator presents a modification to the derivative estimator to solve the drawbacks and improve the performance and accuracy of estimation, which making the proposed one suitable for many applications. A block diagram of the proposed frequency estimator is shown in Fig. 3; it receives the input signal and passes it to one-unit delay.

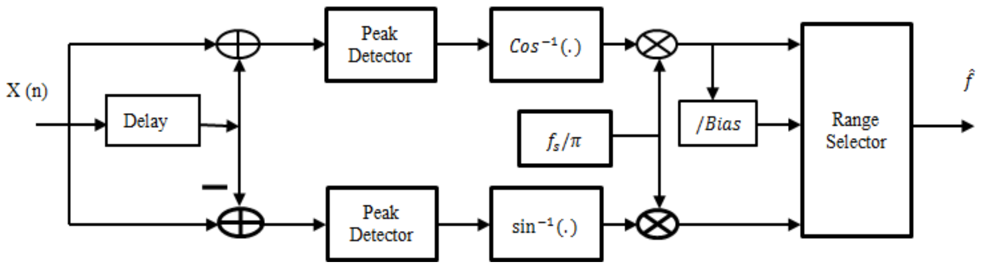

Fig. 3: Block Diagram of Proposed Frequency Estimator.

The mathematical analysis of the proposed estimator can be obtained as follows:
Considering the previous mathematical equations from Eqn. (6) until Eqn. (10), let us define Eqn. (10) to be the estimation of low frequencies $(\hat{f}_L)$

$$\hat{f}_1 = \frac{f_s}{\pi} sin^{-1}(\frac{b}{2a}) \qquad (11)$$

Let us define $x_2(nT_S)$ to be

$$x_2(nT_S) = x(nT_s) + x\big((n-1)T_S\big) = \ a\, cos(2\pi f n T_S + \theta) + a\, cos(\omega(n-1)T_S + \theta)$$

$$= a\, cos(2\pi f T_S + \theta) + a\, cos(2\pi f T_S + \theta)\, cos(2\pi f T_S) + a\, sin(2\pi f n T_S + \theta)\, sin\, (2\pi f T_s)$$

$$= a\, cos(2\pi f n T_s + \theta)\, [1 + cos(2\pi f T_S)] + a\, sin(2\pi f n T_S + \theta)\, sin\, (2\pi f T_S)$$

$$= 2a\, cos(2\pi f n T_S + \theta)\, cos^2(\pi f T_S) + 2a\, sin(2\pi f n T_s + \theta)\, sin(\pi f T_S)\, cos\, (\pi f T_S)$$

$$= 2a\, cos(\pi f T_S)\, [cos(2\pi n T_S + \theta)\, cos(\pi f T_S) + sin(2\pi f n T_S + \theta)\, sin(\pi f T_s)]$$

$$= 2a\, cos(\pi f T_s)\, cos\, (2\pi f n T_S + \theta - \pi f T_S) \qquad (12)$$

Let $c = 2a\, cos\, (\pi f T_S\,)$

We can now define the estimation of high frequencies $(\hat{f}_H)$ to be

$$\hat{f}_2 = \frac{f_s}{\pi} cos^{-1}(\frac{c}{2a}) \qquad (13)$$

Mathematically, we can describe the amplitude of the stationary signal for N samples by the variance (σ^2)

$$var(x) = \sigma_x^2 = \frac{1}{N}\sum_{n=0}^{N-1} x^2(n) \cong \frac{a^2}{2} \qquad (14)$$

$$var(x_1) = \sigma_{x_1}^2 = \frac{1}{N}\sum_{n=0}^{N-1} x_1^2(n) \cong \frac{b^2}{2} \qquad (15)$$

$$var(x_2) = \sigma_{x_2}^2 = \frac{1}{N}\sum_{n=0}^{N-1} x_2^2(n) \cong \frac{c^2}{2} \qquad (16)$$

We can rewrite Eqn. (14), Eqn. (15), and Eqn. (16)

$$a = \sqrt{2\sigma_x^2} \tag{17}$$

$$b = \sqrt{2\sigma_{x_1}^2} \tag{18}$$

$$c = \sqrt{2\sigma_{x_2}^2} \tag{19}$$

Now we can rewrite Eqn. (11) and Eqn. (13) to be

$$\hat{f}_1 = \frac{f_s}{\pi} sin^{-1}\left(\frac{b}{2a}\right) = \frac{f_s}{\pi} sin^{-1}\left(\frac{\sqrt{\sigma_{x_1}^2}}{2\sqrt{\sigma_x^2}}\right) \tag{20}$$

$$\hat{f}_2 = \frac{f_s}{\pi} cos^{-1}\left(\frac{c}{2a}\right) = \frac{f_s}{\pi} cos^{-1}\left(\frac{\sqrt{\sigma_{x_2}^2}}{2\sqrt{\sigma_x^2}}\right) \tag{21}$$

As explained before we have three regions in the dynamic range of the frequency estimator $(0{:}\frac{f_s}{2})$; low-frequency region, medium frequency region, and high-frequency region. We now have Eqn. (20) and Eqn. (21) describes only two solutions to estimate the frequency, by trial and error for each equation with a range of frequencies near zero we find that the low range frequencies can be estimated with the following formula

$$\hat{f}_{low} = \frac{\hat{f}_2}{bias} = \frac{f_s}{\pi} cos^{-1}\left(\frac{c}{2a}\right) = \frac{f_s}{\pi} cos^{-1}\left(\frac{\sqrt{\sigma_{x_2}^2}}{2\sqrt{\sigma_x^2}}\right) \tag{22}$$

Where, $bias = cos^{-1}(0) \times \frac{\pi}{2} = \left(\frac{\pi}{2}\right)^2 = 2.45$

The middle region can be estimated correctly using the formula

$$\hat{f}_{middle} = \hat{f}_1 = \frac{f_s}{\pi} sin^{-1}\left(\frac{\sqrt{\sigma_{x_1}^2}}{2\sqrt{\sigma_x^2}}\right) \tag{23}$$

The high-frequency region can be calculated correctly using the formula

$$\hat{f}_{high} = \hat{f}_2 = \frac{f_s}{\pi} cos^{-1}\left(\frac{\sqrt{\sigma_{x_2}^2}}{2\sqrt{\sigma_x^2}}\right) \tag{24}$$

Eqn. (22), Eqn. (23), and Eqn. (24) can be summarized as follows:

$$\hat{f} = \begin{cases} \frac{f_s}{2.45\pi} cos^{-1}\left(\frac{\sqrt{\sigma_{x_2}^2}}{2\sqrt{\sigma_x^2}}\right), for\ w_0 < f < w_1 \\ \frac{f_s}{\pi} sin^{-1}\left(\frac{\sqrt{\sigma_{x_1}^2}}{2\sqrt{\sigma_x^2}}\right), for\ w_1 \le f < w_2 \\ \frac{f_s}{\pi} cos^{-1}\left(\frac{\sqrt{\sigma_{x_2}^2}}{2\sqrt{\sigma_x^2}}\right), for\ w_2 \le f < w_3 \end{cases} \tag{25}$$

The problem now is how to determine the boundaries of each region, and this is done using a grey wolf optimizer. To calculate the best weights (w_0, w_1, w_2, w_3) for frequency estimation, grey wolf optimizer has used for training bidirectional neural networks to achieve the minimum mean absolute error (MAE) between the input unknown frequency and $\hat{f}_L, \hat{f}_H$. To find the weight w_0, w_1 we use the following equation

$$MAE = \frac{1}{m}\sum_{t=1}^{m}\left|f - \frac{\hat{f}_2}{2.45}\right| \tag{26}$$

The MAE achieves 0.000005 with $w_0 = 0\ and\ w_1 = 0.099956f_s \approx 0.01\ f_s$. To find w_2 we use

$$MAE = \frac{1}{m}\sum_{t=1}^{m}\left|f - \hat{f}_1\right| \tag{27}$$

The MAE achieves 0.000003 with $w_1 = 0.01\ and\ w_2 = 0.399987f_s \approx 0.4\ f_s$. To find w_3 we use

$$MAE = \frac{1}{m}\sum_{t=1}^{m}\left|f - \hat{f}_2\right| \tag{28}$$

It achieves MAE= 0.000007 with $w_2 = 0.4\ f_s\ and\ w_3 = 0.5f_s$
So we can write the estimated frequency as follows:

$$\hat{f} = \begin{cases} \frac{f_s}{\pi} sin^{-1}\left(\frac{\sqrt{\sigma_{x_1}^2}}{2\sqrt{\sigma_x^2}}\right) + \frac{bias}{10}, for\ 0 < f < 0.0001f_s \\ \frac{f_s}{\pi} sin^{-1}\left(\frac{\sqrt{\sigma_{x_1}^2}}{2\sqrt{\sigma_x^2}}\right), for\ 0.0001f_s \le f < 0.4f_s \\ \frac{f_s}{\pi} cos^{-1}\left(\frac{\sqrt{\sigma_{x_2}^2}}{2\sqrt{\sigma_x^2}}\right), for\ 0.4f_s \le f < 0.5f_s \end{cases} \tag{29}$$

Eqn. (29) clears the main difference between the proposed estimator and the traditional derivative estimator. The traditional one defines one equation for the all estimation range ($0{:}\frac{f_s}{2}$) as shown in Eqn. (10), which lacks accuracy and gives large estimation errors in the low-frequency region and the high-frequency region. While the proposed estimator defines three frequency regions determined by the sampling frequency (f_s) to estimate the input frequency. The first region is the region of low frequency ($0 < f < 0.01f_s$), and the third region is a high-frequency region ($0.4f_s < f < 0.5f_s$) while the rest of the range is the middle frequency region($0.01f_s < f < 0.4f_s$). Each frequency region has its accurate estimation equation so that the full estimation range is covered with accurate estimations in all the dynamic range.

a) Effect of Additive Gaussian Noise (AWGN)

This section presents the analysis for the estimator when AWGN added to the input signal; it will be

$$Q(n) = x(n) + y(n) = a\ cos(\omega t + \theta\ ) + y(n) \tag{30}$$

Where $y(n)$ is AWGN.
The signal-to noise ratio (SNR) is the ratio between the signal energy and noise energy in dB scale. $Q(n), x(n), and\ y(n)$ are zero-mean signals; the variance can be used to represent the energy.

$$SNR = 10log\ (\frac{\sigma_x^2}{\sigma_y^2}) \tag{31}$$

At very low SNR we get:

$$\sigma_x^2 \cong \sigma_y^2$$

And

$$\sigma_{x\ 1}^2 \cong \sigma_{x\ 2}^2 \cong 2\sigma_y^2$$

Applying (20), (21) we get

$$\hat{f}_1 = \frac{f_s}{\pi} sin^{-1}\left(\frac{\sqrt{\sigma_{x_1}^2}}{2\sqrt{\sigma_x^2}}\right) = \frac{f_s}{\pi} sin^{-1}\left(\frac{\sqrt{2\sigma_y^2}}{2\sqrt{\sigma_y^2}}\right) = \frac{f_s}{\pi} sin^{-1}\left(\frac{1}{\sqrt{2}}\right) = \frac{f_s}{4} \tag{32}$$

$$\hat{f}_2 = \frac{f_s}{\pi} cos^{-1}\left(\frac{\sqrt{\sigma_{x_1}^2}}{2\sqrt{\sigma_x^2}}\right) = \frac{f_s}{\pi} cos^{-1}\left(\frac{\sqrt{2\sigma_y^2}}{2\sqrt{\sigma_y^2}}\right) = \frac{f_s}{\pi} cos^{-1}\left(\frac{1}{\sqrt{2}}\right) = \frac{f_s}{4} \tag{33}$$

Eqn. (32) & Eqn. (33) is shown that at low SNR the proposed estimator gives a fixed output frequency equal to $\frac{f_s}{4}$.

3.1. Tracking the input frequency

The ability to track the change in input frequency is a required feature for many applications such as demodulation in communication. The proposed estimator can track the frequency change.

$$x(n) = \sum_{m=1}^{k} x_m(n) \tag{34}$$

$$X_m(n) = a_m cos\ (\frac{2\pi f_m}{f_s} n + \theta_m) \tag{35}$$

Where a_m : the amplitude of x_m signal at frequency f_m
The sinusoidal component of the frequency f_m are:

$$x_1(n) = x(n) - x(n-1) = \sum_{m=1}^{k} x_{1m}(n) \tag{36}$$

$$x_2(n) = x(n) + x(n-1) = \sum_{m=1}^{k} x_{2_m}(n) \quad (37)$$

Now the frequency of f_m can be estimated using the following equations

$$\hat{f}_1 = \frac{f_s}{\pi} sin^{-1}\left(\frac{b_m}{2a_m}\right) \quad (38)$$

$$\hat{f}_2 = \frac{f_s}{\pi} cos^{-1}\left(\frac{c_m}{2a_m}\right) \quad (39)$$

Where b_m amplitude of x_{1_m} sinusoidal signal. c_m amplitude of x_{2_m} signal

4. Simulation results

Different simulations have been done to inspect the performance of the proposed simulator. We let the sampling frequency $f_s = 10\ kHz$.

4.1. Accuracy of the proposed estimator

The estimation accuracy is the main feature of the proposed estimator. Fig. 4 shows the estimator's response to the change in the input frequency with step 100 Hz. The estimator has an accurate estimation without any errors in the full estimation range [0:5000 Hz].

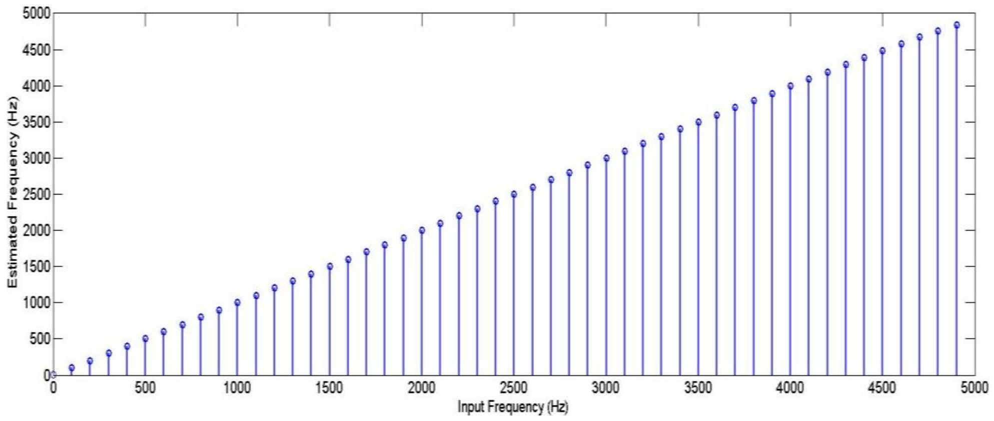

Fig. 4: Proposed Estimator Response for Full Estimation Range.

4.2. Estimation error

We also investigate the performance of the estimator for the low frequencies as this range is important for applications such as seismic applications. A comparison had been done between the response of the derivative estimator and the proposed estimator by calculating the estimation error $(f_{in} - \hat{f}_{low})$. Fig. 5 shows a relation between the estimation error $(f_{in} - \hat{f}_{low})$ and the input frequency (f_{in}) for both the conventional method and the proposed one.

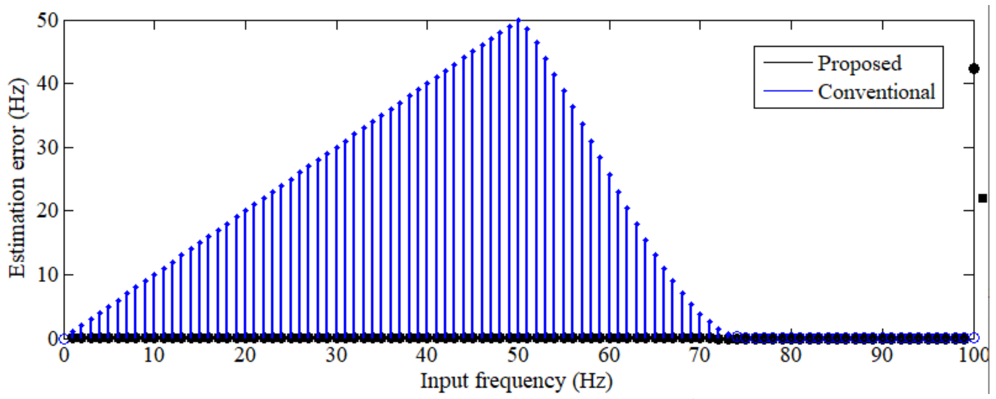

Fig. 5: Comparison between Derivative and Proposed Estimators for Low Frequencies.

Fig. 5 is shown that no estimation errors are founded for the proposed estimator for the input frequency range. While there are estimation errors for the traditional derivative estimator, for input frequencies $0 < f_{in} < 50$ Hz the derivative estimator has 0 Hz output. The output begins to appear for frequencies greater than 50 Hz. In the range $50 < f_{in} < 70$ Hz there is an error in estimation from the frequencies greater than 75 Hz the estimator begins to generate correct output.
Another important range of frequency which is near the Nyquist rate, another comparison is done between the derivative and proposed estimator to investigate the difference between both of them. Fig.6 shows the estimation error, which is the difference between $(f_{in} - \hat{f})$. For the proposed estimator the error is zero, which indicates an accurate estimation for the high frequencies. While the output of the derivative estimator has an estimation, error increased when the input frequency approaches the half sampling frequency $(\frac{f_s}{2})$.

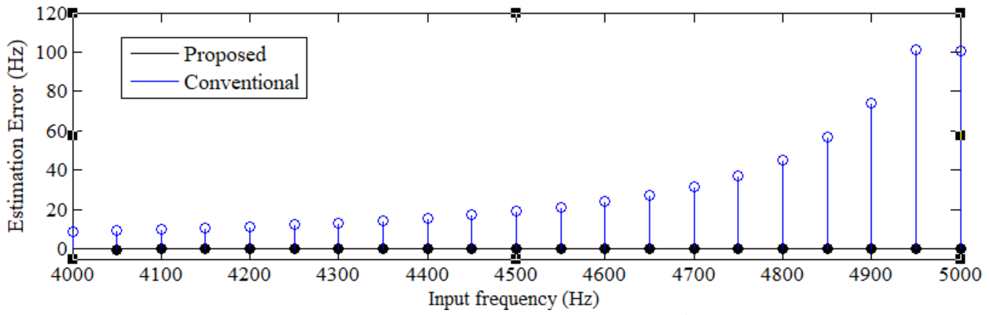

Fig. 6: Comparison between Derivative and Proposed Estimators for High Frequencies.

4.3. Frequency estimation with the effect of AWGN

Fig. 7. is shown the estimation of $1kHz$ with the change in AWGN from $-20{:}\,30\ dB$. As proved in Eqn. (32) & Eqn. (33) the estimated frequency with low AWGN attract to $\frac{f_s}{4}$ which is 2.5 kHz. And an acceptable estimation appear at 8 dB, while the accurate result appear at 16 dB which is a good.

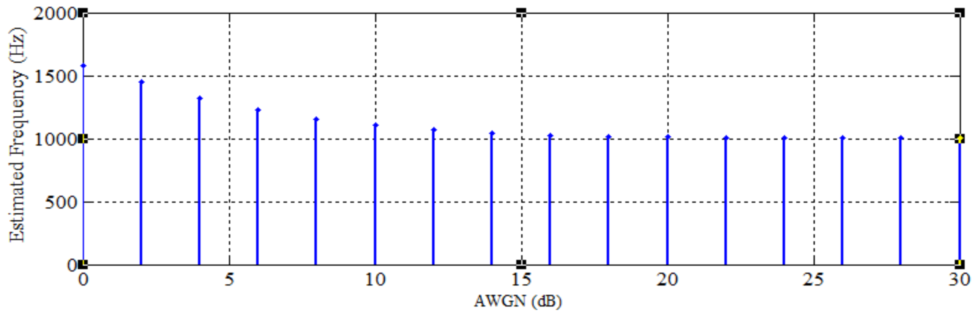

Fig. 7: Effect of AWGN on the Proposed Estimator.

4.4. Tracking input frequency

The estimator needs to track the change in input frequency quickly with no long delay. Fig. 8 shows the tracking for the change of the input frequency with time. The input frequency begins with 1 kHz at 1 ms and increased by 0.5 kHz for each 1 ms. When it reaches 5 kHz, it fixed for 3ms and decreases by 0.5 Khz for each 1 ms. The proposed estimator tracks the input frequency accurately without discontinues at any changing period.

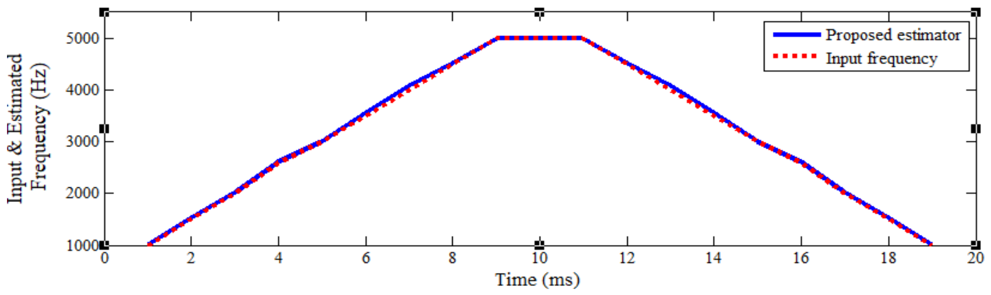

Fig. 8: Tracking Input Frequency.

5. FPGA implementation

The proposed estimator is modeled using the Xilinx system generator tool [22]. VHDL code is used to describe the proposed estimator. All signals of the estimator are fixed-point signals with 16 bits. Fig. 9 shows the proposed estimator model in a Xilinx system generator. The model receives the input signal x_i with AWGN and passes through one delay unit. The output delayed signal from the delay unit passes through adder and subtractor. Peak detector circuit receives the output of subtractor and the input signal to find the peak of each sinusoidal signal and calculate the ratio between the peak of each signal. Another peak detector circuit receives the output of the adder and the input signal to find the peak of each sinusoidal signal and calculate the ratio between the peak of each signal. The two generated signals from the first and second peak detector circuits are directed to arcsine, and arcos circuits respectively. The format block is the circuit which performs the Eqn. (29), receives the output of the arcsine and arcos, compare the value of arcsine input with f_s , determine the boundaries of Eqn. (29), and finally, calculate the output estimated frequency. The main blocks in the proposed estimator model are:-

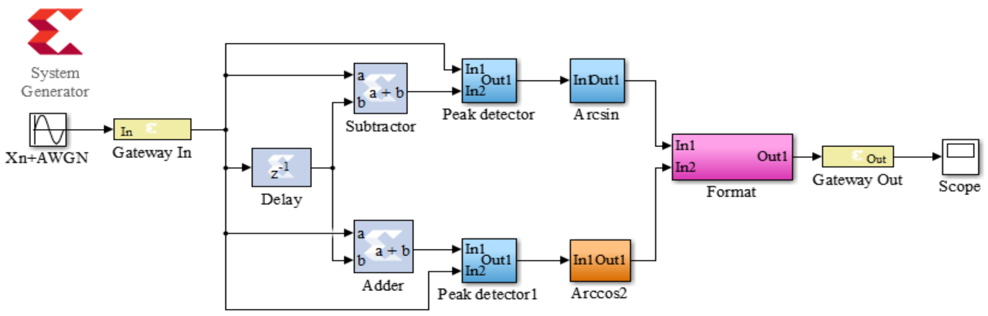

Fig. 9: Hardware Model of Proposed Frequency Estimator.

5.1. Peak detector

The peak detector block determines the maximum value of the amplitude of the sinusoidal wave; it consists of one delay unit and a comparator. The input is first rectified so that no negative values still existed. The input to the delay $|u(n)|$ unit and the output of the delay $|u(n-1)|$ are compared using a comparator when $|u(n)| < |u(n-1)|$ is founded; it means the maximum point is reached and it holds until the next maximum is determined. Fig. 10 shows the input to the peak detector circuit, which is sinusoidal with varying amplitude waveform, and the final output of the block. The final output is changing according to the changing of the amplitude of the input signal.

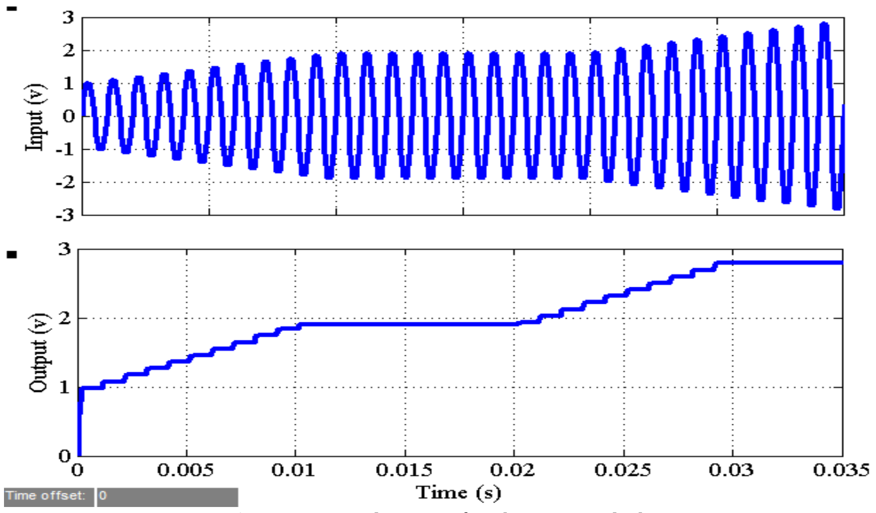

Fig. 10: Input and Output of Peak Detector Block.

5.2. Arcsin block

The arcsine and arcos blocks are designed using cordic algorithm, which is a common algorithm to calculate the different trigonometric functions. the idea of cordic algorithm is "rotating" the phase of a complex number, by multiplying it by a succession of constant values. however, then multiplies can all be powers of 2, so in binary arithmetic, they can be done using just shifts and adds; no actual multiplier is needed. compared to other approaches, cordic is a clear winner when a hardware multiplier is unavailable, e.g. in a microcontroller, or when you want to save the gates required to implement one, e.g. in an fpga. the input and output of the arcsine block are shown in Fig. 11.

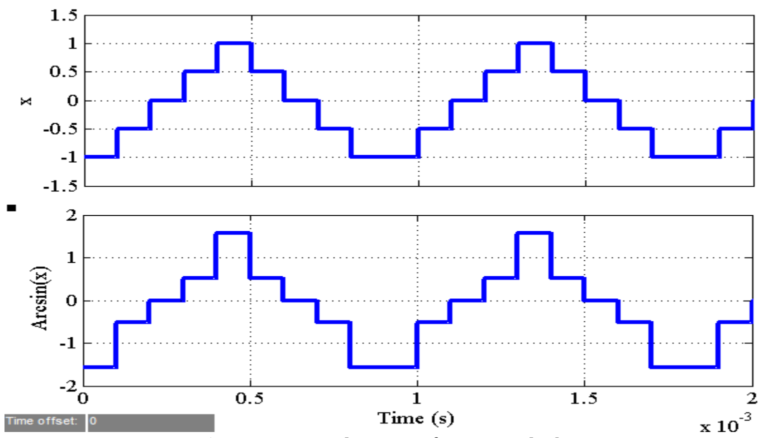

Fig. 11: Input and Output of Arcsine Block.

5.3. Format block

This block receives the arcsine and arcos signals and generates the final estimated frequency. The task of this block is to apply Eqn. (9) by choosing the boundaries of the equation. The first step is comparing the input arcsine value with the sampling frequency (f_s), if the input is less than $(0.1f_s)$ then the output frequency will be the arcos input divided by 2.45, while if the input arcsine is greater than $(0.4f_s)$, then the output frequency is arcos input. When both previous conditions did not realize the output will be the arcsine input. Fig. 12 shows the estimated frequency for an input frequency of $1kHz$, and it is clear that the estimator takes 0.5ms (half a full period of the input signal) to generate the accurate estimated frequency. This delay because of different calculations through the estimator. The estimator implemented using Spartan-6 X6SLX45 board. FPGA consumed resources are shown in Table 1.

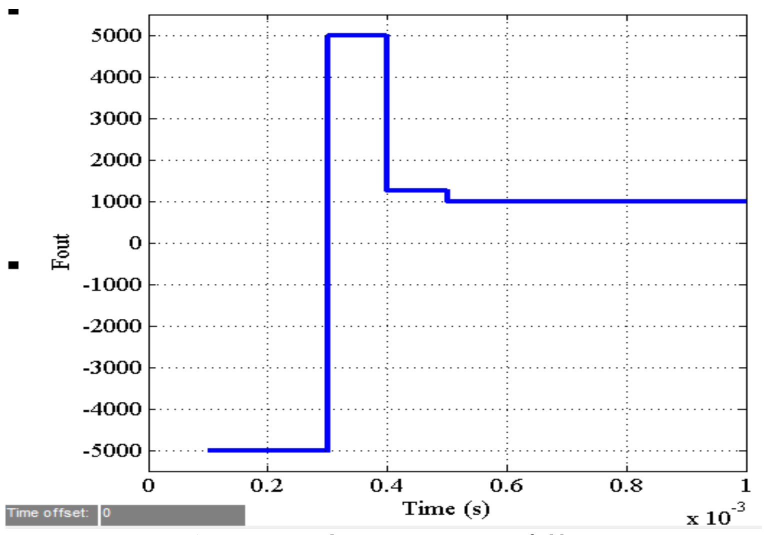

Fig. 12: Proposed Estimator Response of 1khz.

Table 1: FPGA Consumed Recourses

Component	Used	Available
Slice flip flops	500	33280
4 Input LUTs	980	33280
Occupied Slices	1020	16640
Bounded IOBs	32	84
Maximum Frequency	375 MHz	
Power Consumption	265 mW	

6. Conclusion

An accurate and stable proposed frequency estimator, which solved the limitations of the derivative algorithm has been presented. It solves the problem of traditional derivative estimators, which is the inability to estimate low frequencies (near zero), and high frequencies (near half the sampling frequency). The proposed estimator divides the dynamic range into three ranges; low-frequency range, middle frequency range, and high-frequency range. Each range has its correct formula. The boundaries of each region have been calculated by using the grey wolf optimization technique, which showing the effectiveness and superiority of deep learning over the traditional methods. Simulation results investigate the performance of the proposed estimator in case of a single frequency and multiple frequencies with AWGN. All simulation results indicate the ability of the proposed estimator to estimate a correct frequency accurately in all the dynamic range. Comparisons between proposed and traditional ones have been made and simulation results showed the enhancement in performance provided by the proposed one. The proposed frequency estimator has been implemented with low complexity using FPGA, consumed less power about 265 mW, and worked at 375 MHz.

References

[1] Ch. Namitha, V. Uma Mahesh, M. Anusha, S. K. Rao, V. Chandra,"Frequency Estimation Using Minimum Algorithm on Seismic Data", Microelectronic, Electromagnetics and Telecommunications, Vol. 471 , pp. 153-163, 2018. https://doi.org/10.1007/978-981-10-7329-8_16.

[2] Y. Sun, T. Fei and N. Pohl, "A High-Resolution Framework for Range-Doppler Frequency Estimation in Automotive Radar Systems," in IEEE Sensors Journal, vol. 19, no. 23, pp. 11346-11358, 1 Dec.1, 2019. https://doi.org/10.1109/JSEN.2019.2933776.

[3] Kim, J., Kim, J., Nguyen, L. et al.," Tonal signal detection in passive sonar systems using atomic norm minimization", EURASIP Journal in Advanced Signal Processing, vol. 43, 2019. https://doi.org/10.1186/s13634-019-0641-5.

[4] T. Mohsen, Z. J. Mehdi, J. Mojtaba," A Novel Method for Frequency Estimation Considering Instrument Transient Effect" International Journal of Electrical and Computer Engineering, vol. 5, pp. 177-188, 2015. https://doi.org/10.11591/ijece.v5i2.pp177-188.

[5] J.-R. Liao and S. Lo, "Analytical solutions for frequency estimators by interpolation of DFT coefficients," Signal Processing, vol. 100, pp. 93–100, July 2014. https://doi.org/10.1016/j.sigpro.2014.01.012.

[6] B. G. Quinn, "Estimating frequency by interpolation using Fourier coefficients," IEEE Trans. Signal Processing, vol. 42, no. 5, pp. 1264–1268, May 1994. https://doi.org/10.1109/78.295186.

[7] N. Thong-un, W. Wongsaroj, W. Treenuson, J. Chanwutitum, H. Kikura, "Doppler frequency estimation using makimum likelihood function for low ultrasonic velocity profile", Acoustical Letter, vol. 38 (5), pp. 268:271, 2017. https://doi.org/10.1250/ast.38.268.

[8] S. Zhao, X. Guang, L. Zhang, "Efficient Iterative Frequency Estinator of Sinusoidal Signal in Noise", Circuits, Systems, and signal Processing, vol. 36(8), pp. 3265:3288, 2017. https://doi.org/10.1007/s00034-016-0453-x.

[9] W. Liu et al. "A Novel Carrier Loop Algorithm Based on Maximum Likelihood Estimation (MLE) and Kalman Filter (KF) for Weak TC-OFDM Signals." Sensors (Basel, Switzerland) vol. 18 (7), 2018. https://doi.org/10.3390/s18072256.

[10] K. Charles, R. Carrasco, R. Parra, "Complexity Reduction of MLSE and MAP Equalizers Using Modified Prolate Basis Expansion", Electronics, vol. 8 (11), 2019. https://doi.org/10.3390/electronics8111333.

[11] Myriam Desainte-Catherine and Sylvain Marchand, "High Precision Fourier Analysis of Sounds Using Signal Derivatives," Journal of the Audio Engineering Society, vol. 48, no. 7/8, pp. 654–667, July/August 2000.

[12] Sylvain Marchand, “Improving Spectral Analysis Precision with an Enhanced Phase Vocoder using Signal Derivatives,” in Proc. DAFx, Barcelona, November 1998, pp. 114–118.
[13] El-kenawy, E. S. M. T. (2019). A Machine Learning Model for Hemoglobin Estimation and Anemia Classification. International Journal of Computer Science and Information Security (IJCSIS), 17(2).
[14] H. Hassan, A. I. El-Desouky, A. Ibrahim, E. M. El-kenawy and R. Arnous, (2020) "Enhanced QoS-based Model for Trust Assessment in Cloud Computing Environment," in IEEE Access. https://doi.org/10.1109/ACCESS.2020.2978452.
[15] El-kenawy, E. S. M., El-Desoky, A. I., & Al-rahamawy, M. F. (2012). Distributing Graphic Rendering using Grid Computing with Load Balancing. International Journal of Computer Applications, 975, 888.
[16] El-Kenawy, E. S. M. T., El-Desoky, A. I., & Sarhan, A. M. (2014). A bidder strategy system for online auctions trust measurement. International Journal of Strategic Information Technology and Applications (IJSITA), 5(3), 37-47. https://doi.org/10.4018/ijsita.2014070103.
[17] El-Knawy, E. S. M. T., & El-Desoky, A. I. (2016). TRUST MEASUREMENT FOR ONLINE AUCTIONS: PROPOSAL OF NEW MODEL. INTERNATIONAL JOURNAL OF INNOVATIVE COMPUTING INFORMATION AND CONTROL, 12(2), 385-394.
[18] W. Xiaohong, R. Tian, “Classification of CT brain images based on deep learning networks”, Computer Methods and Programs in Biomedicine, Vol. 138,pp. 49-56, 2017. https://doi.org/10.1016/j.cmpb.2016.10.007.
[19] X. Li, F. Dong, S. Zhang, W. Guo, “A Survey on Deep Learning Techniques in Wireless Signal Recognition”, Wireless Communications and Mobile Computing, Vol. 2019. https://doi.org/10.1155/2019/5629572.
[20] Hassib, E. M., El-Desouky, A. I., Labib, L. M., & El-kenawy, E. S. M. WOA+ BRNN: An imbalanced big data classification framework using Whale optimization and deep neural network. Soft Computing, 1-20.
[21] Hassib, E. M., El-Desouky, A. I., El-kenawy, E. S. M., & Elghamrawy, S. (2019). An Imbalanced Big Data Mining Framework for Improving Optimization Algorithms Performance. IEEE Access. https://doi.org/10.1109/ACCESS.2019.2955983.
[22] Xilinx, Vivado, Design Suite User Guide: Model-Based DSP Design using System Generator, UG897, v2016.1 ed., Xilinx, Apr. 2018.