A Novel Approach to Peak Detection Using Sequential Learning Algorithm

B. N. Sumukha[1], R. Chandan Kumar[1], Skanda S. Bharadwaj[1], and Koshy George[1,2]

[1]PES Centre for Intelligent Systems

[2]Department of Telecommunication Engineering

PES Institute of Technology (now, PES University)

100 Feet Ring Road, BSK 3rd Stage, Bangalore 560085, India

***Abstract*—Peak detection is a facile wing of signal processing. Conventional peak detection algorithms detect peaks when the entire signal is made available to them. In contrast, we propose a method that is based on recognizing the fundamental shapes of a signal, and the overall method intuitive in nature. Towards this, we use a feedforward neural network that is trained using the online sequential learning algorithm which provides better convergence performance relative to the back propagation algorithm. Moreover, the training avoids complex pre-processing tasks and feature extraction. Most importantly, the entire signal is not required for the algorithm to detect the peaks.**

Keywords—Peak detection; artificial neural networks; online sequential learning

I. Introduction

Peak detection is an antiquated problem, where ample research has been carried out and innumerable techniques have been developed. It is a quotidian requirement in scientific data processing. Peak detection finds itself applicable in a wide variety of domains. For example, in [1], it has been shown that peak detection is integral in determination of diseases and malfunctioning of the heart. Peak detection is applied to image data reduction (quantization) [2]. There is a profusion of algorithms that has been developed for peak detection.

Peak detection can be carried out by conventional window threshold technique [3-4]. Other interesting approaches include the use of wavelet transforms [5], Hilbert transforms [6], nonlinear filtering [7], Kalman filtering [8], and Hidden Markov Models (HMM) [9]. Machine learning techniques like K-Means clustering [10], fuzzy C-Means clustering [11], higher-order statistics [12], automatic multiscale-based peak detection (AMPD) algorithm [13], and Artificial Neural networks [14-15] have also been used with some remarkable results. The above-mentioned techniques are intuitively cumbersome as it involves a lot of pre-processing and feature extraction. The major prerequisite for the techniques presented above is the presence of the entire signal at the time of detection. We present an algorithm, which avoids these requirements.

In the context of peak detection using artificial neural networks (ANN), there is very little work reported using semi-supervised and unsupervised learning (e.g., [14] and [15]). Supervised learning is predominant in this literature. Since it involves a lot pre-processing and feature extraction, it makes the algorithm dependent on the availability of the entire signal and is not best suited for application in which the signals are acquired sequentially.

All the techniques mentioned above consider offline training of the acquired data. To the best knowledge of the authors, there appears to be no work reported that explores online sequential learning. From a practical viewpoint, online learning is better suited for real-time applications. Moreover, offline techniques rely on the features that are extracted from the signals in the dataset. Intuitively, this process is computationally expensive and time consuming.

The back propagation algorithm (BPA) is an algorithm that can be considered for sequential learning. However, it is well known that BPA requires a rather long time to train the network. Further, it is quite likely that learning stops as it gets trapped in a local minimum. Huang and his co-workers proposed the Extreme Learning Machines (ELMs) that overcome these drawbacks [16]. Such machines are feed-forward neural networks (FFNNs) with a single hidden layer. The synaptic weights corresponding to the input layer is made independent of the environment by keeping it fixed, and the synaptic weights of the output layer is determined as a solution to a least-squares problem. Evidently, the training time is much faster than BPA as the requirement for iterations is completely eliminated.

The authors in [17] proposed a sequential version of ELM and called it the online sequential extreme learning machine (OSELM). This algorithm requires the inversion of a matrix that is related to the data. However, it has been the experience in [18] that OSELM can perform rather poorly as this matrix is often singular or nearly singular. Another sequential variant called the online sequential learning algorithm (OSLA) was used in [19], and observed to perform quite satisfactorily in the

978-1-5090-5256-1/16/$31.00

context of system identification and control.

Sinusoids are the most commonly occurring signal in nature. Performing peak detection on such a signal helps setting the stage for peak detection in other complex signals. To begin with, on-line peak detection is first tried on sinusoidal signals. A sum of sinusoids signal composed of several frequency components is depicted in Fig. 1. As evident from the figure, such signals have many peaks and detection of such peaks has been a problem of interest. In this paper, we present an algorithm to find peaks in signals that are sinusoidal in nature using an artificial neural network trained sequentially using OSLA.

The goal of this paper is two-fold: First, to propose an algorithm that would provide comparable detection accuracy in the context of peak detection. The second objective is to use OSLA in the context of pattern classification in order to substantiate the claim that the entire signal is not really necessary to detect peaks in the given signal. The algorithm does not require feature extraction and allows sequential learning. The methodology is explained in Section II and the learning algorithm in Section III. The results using an available dataset are presented in section IV.

II. METHODOLGY

The database is synthesized by generating different shapes from sinusoidal signals that contain several frequency components. The peak detection algorithm initially addresses a basic shape detection or shape classification problem. Once the required shape is obtained, the algorithm proceeds to detect the peaks. Consider a sinusoidal signal that is varying from 0 to 2π, the algorithm distinguishes the signal into 4 fundamental shapes. These 4 basic shapes are depicted in Fig. 2. A sliding window of length L is incorporated in order to split the signal into the above-mentioned shapes. This window length L depends on the system under consideration. The window is again divided into two equal parts of length equal to L/2 and the effective slopes of the two windows are analyzed to differentiate the shapes. If both halves of the window are found to have an effective rising gradient, the shape is said to be under class 1. If the effective slope of the first half of the window has a rising gradient and the other has a falling gradient i.e. the window, as a whole resembles a hat function, the shape is said to be under class 2. If the effective slopes of both halves are found to have a decreasing gradient, the shape is said to be under class 3. Finally, if the effective slope of the first half of the window has a decreasing gradient and the second half has a rising gradient i.e. the window as a whole resembles an inverted hat function, then the shape is said to be under class 4. To generate the above-mentioned database, we extract the shapes from different signals that are of the form,

$$y(t) = \sin(2\pi f_1 t) + \sin(2\pi f_2 t) + \cdots + \sin(2\pi f_n t)$$

Signals of the form as represented in the above equation consist of sum of sinusoids with different frequency components, varied over a certain range are considered for creating the database. Such signals are passed through the sliding window and different shapes are extracted and bifurcated into their respective classes as explained before. The length of the sliding window depends on the sampling frequency. Thus, the length of the sliding window fixes the number of samples making up an element in the database.

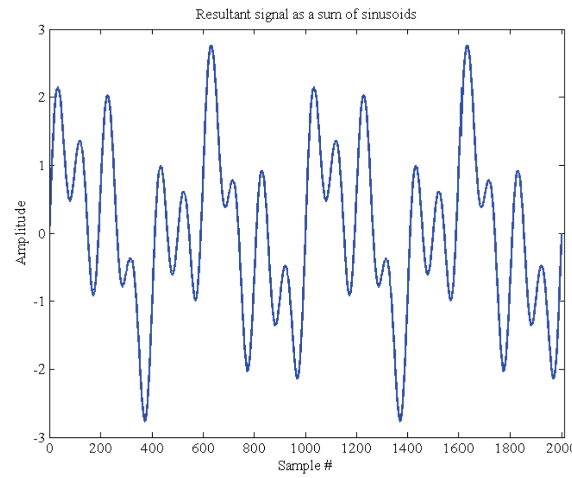

Fig. 1. A signal as a sum of sinusoids with different frequency components

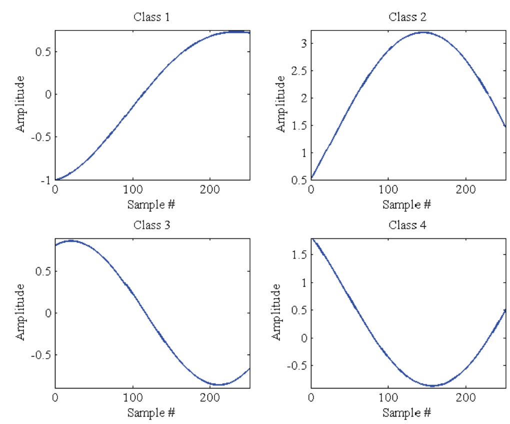

Fig. 2. Basic shapes considered for training the ANN for shape detection.

A database of different shapes and their corresponding class labels are created using 10 signals represented by (1), where each signal is made up of 7 frequency components (n = 7). The frequency is varied from 1 through 31 Hz. Such various datasets can be created for peak detection where the number of samples in each is varied.

The objective of this paper is to propose an algorithm, which detects peaks in the signal without the need of having the entire signal beforehand. The algorithm detects the peak as and when the signal is acquired sequentially. In order to give a more pragmatic approach, the training signals are sent into neural network in a random fashion. Once the network is entirely trained and the weights are updated, the network is tested in a fashion that is similar to the signal received by a system in real time. The entire signal is sent into the network sequentially through the sliding window that is now capable of understanding the shape of the chunk of the signal contained in it.

The raw signal is directly sent into the ANN sequentially through a sliding window of length L. For instance, in our algorithm we considered a window of length 251, which is dependent on the number of samples in the signal; i.e., it depends on the sampling frequency of the signal. The network contains an input layer, an output layer and only one hidden layer. The network is trained using OSLA where the weights corresponding to the first layer (i.e., W_1) are randomized, and only the output weights (i.e., W_2) are determined from a set of simultaneous linear equations.

Any example belonging to class 2 is of importance due to the existence of a peak. Thus, when the ANN classifies a window as class 2, we proceed to find the peak in that chunk of the signal. To find the peak in the window, a very fundamental approach is followed. The window is said to contain a peak when its centre has the maximum amplitude. This ensures that the peak is detected only once and thus eliminating the detection of the same peak as the window slides sequentially one-by-one.

III. Online Sequential Learning Algorithm

Consider an FFNN with a single hidden layer. The input layer has m_0 nodes, the hidden layer has m_1 neurons and the output layer has m_2 neurons. Such a network is denoted $N_{m_0:m_1:m_2}$. Whilst the hidden layer has a nonlinear activation function, the output layer is linear. For an FFNN $N_{m_0:m_1:m_2}$, the matrix of synaptic weights that connects the inputs nodes to the hidden layer is denoted W_1, and the matrix of synaptic weights that connects the hidden layer to the output layer is denoted W_2. Clearly, $W_1 \epsilon R^{m_1\times(m_0+1)}$ and $W_2 \epsilon R^{m_2\times(m_1+1)}$, where the bias is also taken into account. The activation function of the hidden layer is $\varphi(v) = \text{atanh}\,(bv)$ and the output layer is a linear layer with unity gain. We denote the input to the network at instant k as x_k and the outputs of the hidden and output layer respectively denoted $y_{1,k}$ and $y_{2,k}$. The desired output at instant k is denoted $y_{d,k}$. Clearly, $x_k \in R^{m_0}$, $y_{1,k} \in R^1$ and $y_{2,k}, y_{d,k} \in R^{m_2}$.

In online sequential learning algorithm, the weight matrix W_1 is randomly initialized and the weight matrix W_2 is initialized as a zero matrix. Subsequently, only the weight matrix W_2 is updated in accordance with the following equations:

$$P_{k+1} = P_k - \frac{P_k y_{1,k+1} y_{1,k+1}^T P_k}{1 + y_{1,k+1}^T P_k y_{1,k+1}} \tag{1}$$

$$W_{2,k+1} = W_{2,k} + (y_{d,k+1} - W_{2,k} y_{1,k+1}) y_{1,k+1}^T P_{k+1} \tag{2}$$

These recursions are initialized as follows:

$$P_0 = \frac{1}{\lambda} I \tag{3}$$

$$W_{2,0} = 0 \tag{4}$$

for some sufficiently small λ. When used for system identification and control, SELA improves the transient performance when compared to the conventional back propagation algorithm [18], [19].

It may be noted that the recursions of OSLA arises when the Tikhonov-Phillips functional

$$J = \|W_2 Y_1 - Y_d\| + \lambda \|W_2\| \tag{5}$$

is minimized. Here,

$$Y_1 = (\bar{y}_{1,1} \quad \bar{y}_{1,2} \quad \cdots \quad \bar{y}_{1,N}) \tag{6}$$

$$Y_2 = W_2 Y_1 \tag{7}$$

$$Y_d = (y_{d,1} \quad y_{d,2} \quad \cdots \quad y_{d,N}) \tag{8}$$

where, the training data is denoted $\{x_k\,, y_{d,k}\}_{k=1}^N$, and

$$y_{1,k} = \varphi(W_{1,k}\, \bar{y}_{0,k})$$

$$\bar{y}_{0,k} = \begin{pmatrix} 1 \\ x_k \end{pmatrix} \qquad \bar{y}_{1,k} = \begin{pmatrix} 1 \\ y_{1,k} \end{pmatrix}$$

The functional has two terms, one connected with the approximation error and the second connected with the weight matrix. The parameter λ allows trade-off between these two objectives. This parameter is called the regularization parameter. It may be noted that a least-squares solution is sought and hence is the global minimum. Accordingly, OSLA converges to the global minimum. Further, since only the output weight matrix is adapted, the convergence is faster. It appears that the algorithm update equations (1) and (2) are similar to OSELM which is the sequential version of the extreme learning machine. We emphasise that the initializations in OSELM is quite different from (3) and (4). This difference leads to marked improvement in the performance of the algorithm.

IV. Results

The shape defining signals used in this paper are from the database mentioned earlier. We first described the training of the network using these components. It may be recalled that the signals used to synthesize the components of the database are made up of 7 sinusoidal frequency components in which the frequency ranges between 1 through 31 Hz. We use a network with an input layer with 251 nodes, a hidden layer with 400 nodes and an output layer with 4 nodes represented by $N_{251:400:4}$. The accuracy achieved for shape detection by OSLA is 96.7%.

The confusion matrix of shape detection is shown in Fig. 3. The database consists of 3990 elements and belong to the 4 classes (C_1 C_2 C_3 and C_4) mentioned earlier. Of these, 877 (i.e., 22%) are correctly classified as C_1, 1071 (i.e., 26.8%) are correctly classified as C_2, and so on. From the off-diagonal components it can be seen that 10 components are wrongly classified as C_2 and 8 as C_4, which belongs to C_1. Similarly, 27 elements are wrongly as C_1 and 20 elements are classified as C_3, which actually belong to C_2. Also, 11 elements are classified as C_2 and 8 as C_4, which belongs to C_3. Lastly, 14 are wrongly classified as C_1 and 22 as C_3, which belong to C_4. Each column therefore provides this information about a particular class. Accordingly, the last row gives the percentage of classification – correct and incorrect – for each class.

Finally, out of 918 elements classified as C_1, 877 (i.e., 95.5% of 918) are correct and 41 (i.e., 4.5% of 918) are incorrect. Similarly, out of 1092 elements classified as C_2 1071 (i.e., 98.1% of 1092) are correctly classified and 21 (i.e., 1.9% of 1092) are incorrectly classified. The last column of the confusion matrix provides this information. Finally it can be seen that the overall correctness of the shape detection is 97% and incorrect classification is 3%.

Using the trained network with a configuration that gave the above-mentioned results, peaks were found for test signals that are sum of sinusoids and also for ECG signals of two different subjects. The test was run for 100 different sinusoid signals whose frequency components were randomly generated in each run. The accuracy is calculated as the number of peaks found by the proposed algorithm compared with that found by the MATLAB built-in function *findpeaks*. The network achieved an accuracy of 98.32%, accuracy that is averaged over 100 trial runs.

A comparison of peak detection performed by the proposed algorithm and that performed by the conventional algorithm used in *findpeaks* is depicted in Fig. 4. Fig. 4a depicts the peaks found by this function and Fig. 4b depicts the peaks detected by the proposed peak detection algorithm.

To substantiate the potential of the algorithm, it was also experimented on the ECG signals of human beings. ECG signals are quite complicated when compared to a sinusoidal signal. The variations in the former signal can be quite drastic when compared to the latter signal. The proposed algorithm was able to detect peaks in the ECG signals as well though it was not trained to do so. Similar comparison studies of peak detection for ECG signals are depicted in Fig. 5 and Fig. 6. In these figures the results of the function *findpeaks* as well as the proposed technique are shown. ECG signals from two different subjects are considered in these figures. It can be observed from Fig. 6 that our method is able to pick the peaks despite the presence of a very high frequency component.

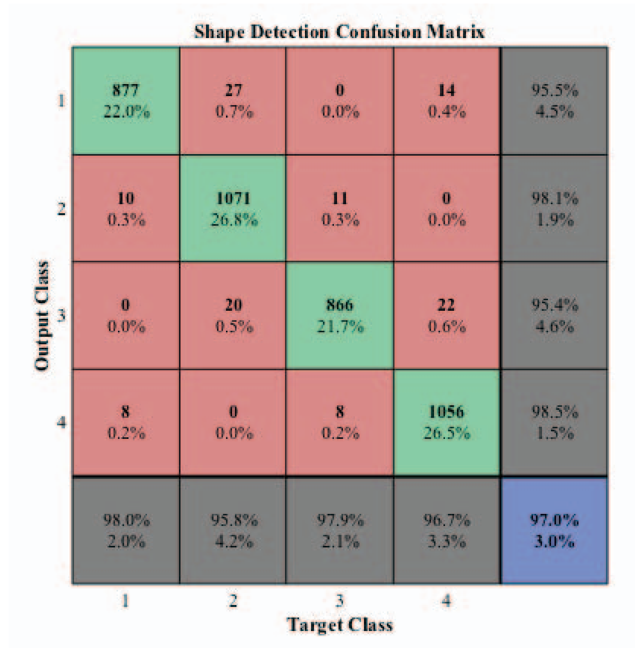

Fig.3. Confusion matrix for shape detection

Thus, in general, the algorithm proves that the peaks of any signal of sinusoidal nature can be detected without having to perform any sort of pre-processing or feature extraction. Since the algorithm detects the shape of the signal to find peaks only the sampling frequency of the signal has to be taken into consideration.

Comments: (a) The two important factors that affect the performance of the algorithm are as follows: (i) the frequency range of the system under consideration and (ii) the highest frequency component in the signal (i.e., the sampling frequency of the system). The length of the chosen window is affected by these factors and hence has to be chosen carefully. (b) It has also been observed that increasing the number of frequency components within the given frequency range does not affect the algorithm.

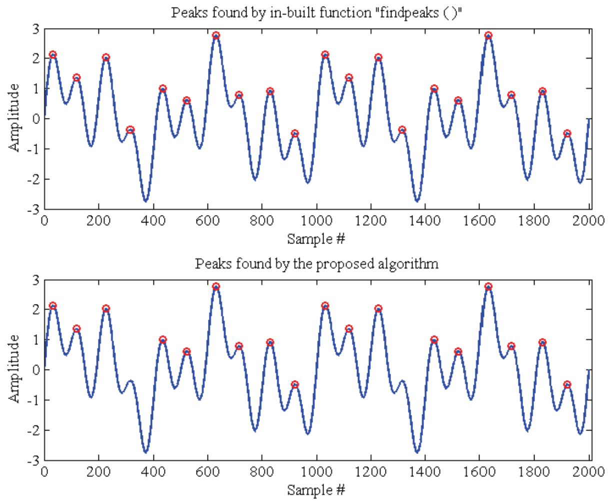

Fig. 4. Comparison study of peak detection methods for sum of sinusoids: (a) Peaks found by the function *findpeaks*. (b) Peaks found by the proposed technique.

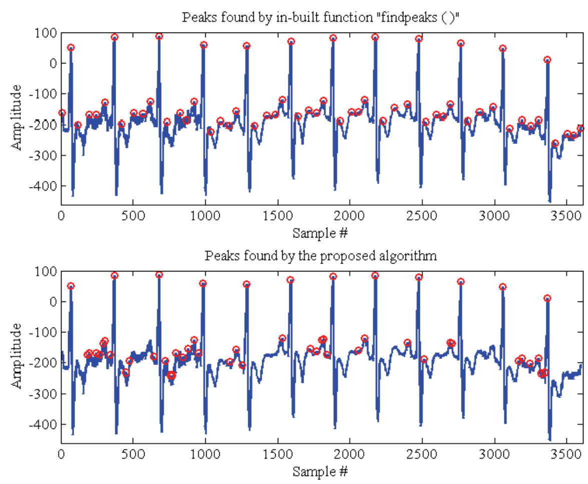

Fig. 5. Comparison study of peak detection methods for a ECG signal form Subject 1: (a) Peaks found by the function *findpeaks*. (b) Peaks found by the proposed technique.

V. CONCLUSIONS

A method for detecting peaks is proposed on this paper. This requires a feedforward neural network to be trained to classify segments of the signal into four fundamental shapes. The network is trained with the online sequential algorithm that avoids complex pre-processing and feature extraction. Accordingly, the whole process is intuitively appealing. The novelty is in the capability to detect peaks using merely sequential processing of data, and hence amenable for real-time implementations. Moreover, a network that has been trained on the four fundamental shapes of a sinusoidal is able to detect the peaks in more complex signals such as the electrocardiogram.

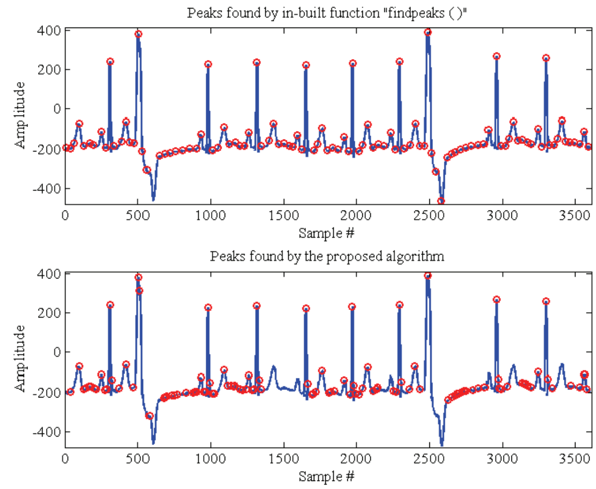

Fig. 6. Comparison study of peak detection methods for a ECG signal form Subject 2: (a) Peaks found by the function *findpeaks*. (b) Peaks found by the proposed technique.

REFERENCES

[1] S. Chattopadhyay, S. Das, A. Nag, J. K. Ray, S. Bhattacharjee, and B. Neogi, "Design and Simulation Approach Introduced to ECG Peak Detection with Study on Different Cardiovascular Diseases," *International Journal of Scientific and Research Publications (ISSN)*, vol. 2, no. 12, pp. 2250-3153, December 2012.

[2] M. I. Sezan, "A peak detection algorithm and its application to histogram-based image data reduction," *Computer Vision, Graphics, and Image Processing,* vol. 49, no. 1, pp. 36-51, January 1990.

[3] M. L. Jacobson, "Auto-Threshold peak detection in physiological signals," in *Proceedings of the 23rd Annual International Conference of the IEEE*, Al-Ain, UAE, 2001, pp. 2194–2195.

[4] J. Pan and W. J. Tompkins, "A real-time QRS detection algorithm," *IEEE Trans. Biomed. Eng.*, vol. 32, pp. 230–236, 1985.

[5] P. Du, W. A. Kibbe, and S. M. Lin, "Improved peak detection in mass spectrum by incorporating continuous wavelet transform-based pattern matching," *Bioinformatics,* vol. 22, pp. 2059–2065, 2006.

[6] D. Benitez , P. A. Gaydecki, A. Zaidi, and A. P. Fitzpatrick, "The use of the Hilbert transform in ECG signal analysis," *Comput. Biol. Med.*, vol. 31, pp. 399–406, 2001.

[7] M. Aboy, J. McNames, T. Thong, D. Tsunami, M. S. Ellenby, and B. Goldstein, "An automatic beat detection algorithm for pressure signals," *IEEE Trans. Biomed. Eng.*, vol. 52, pp. 1662–1670, 2005.

[8] A.T. Tzallas, V. P. Oikonomou, and D. I. Fotiadis, "Epileptic spike detection using a Kalman filter based approach," in *Proceedings of the 28th IEEE EMBS Annual International Conference*, New York, NY, USA, 2006, pp. 501–504.

[9] D. A. Coast, R. M. Stern, G. G. Cano, and S. A. Briller, "An approach to cardiac arrhythmia analysis using hidden Markov models," *IEEE Trans. Biomed. Eng*., vol. 37, pp. 826–836, 1990.

[10] S. S. Mehta, D. A. Sheta, N. S. Lingayat, and V.S. Chouhan, "K-Means algorithm for the detection and delineation of QRS-complexes in electrocardiogram," *IRBM* 2010, vol. 31, pp. 48–54, 2010.

[11] S. Sharma, S. S. Mehta, and H. Mehta, "Development of derivative based algorithm for the detection of QRS-complexes in single lead electrocardiogram using FCM," *IJCA* 2011, vol. 4, pp. 19–23, 2011.

[12] K. I. Panoulas, L. J. Hadjileontiadis, and S. M. Pasa, "Enhancement of R-wave detection in ECG data analysis using higher-order statistics," in *Proceedings of the 23rd Annual International Conference of the IEEE*, Thessaloniki, Greece, 2001, pp. 344–347.

[13] F. Scholkmann, J. Boss, and M. Wolf, "Efficient Algorithm for Automatic Peak Detection in Noisy Periodic and Quasi-Periodic Signals," *Algorithms,* vol. 5, pp. 588-603, 2012.

[14] Q. Xue, Y. H. Hu, and W. J. Tompkins, "Neural-Network-Based adaptive matched filtering for QRS detection," *IEEE Trans. Biomed. Eng.*, vol. 39, pp. 317–329, 1992.

[15] G. Vijaya, V. Kumar, H. K. Verma, "ANN-Based QRS-complex analysis of ECG," *J. Med. Eng. & Technol.*, vol. 22, pp. 160–167, 1998.

[16] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: A new learning scheme of feed-forward neural networks," in *Proceedings of IEEE International Joint Conference on Neural Networks (IJCNN2004),* Budapest, Hungary, July 2004, pp. 985–990.

[17] N.-Y. Liang, G.-B. Huang, P. Saratchandran, and N. Sundararajan, "A fast and accurate online sequential learning algorithm for feed-forward networks," *IEEE Transactions on Neural Networks,* vol. 17, pp. 1411–1423, 2006.

[18] K. Subramanian, S. G. Krishnappa, and K. George, "Performance comparison of learning algorithms for system identification and control," in *Proceedings of the 12th IEEE India International Conference (INDICON 2015),* New Delhi, India, December 2015.

[19] K.George, K.Subramanian, and N.Seshadhri, "Improving transient performance in adaptive control of nonlinear systems," in *Proceedings of the 4th International Conference on Advances in Control and Optimization of Dynamical Systems (ACODS16),* Tiruchirappali, India, February 2016, pp. 658–663.