
Qingsong Xu

Department of Electromechanical Engineering, Faculty of Science and Technology, University of Macau, Macao, China

Correspondence should be addressed to Qingsong Xu; qsxu@umac.mo

Received 28 April 2014; Accepted 1 July 2014; Published 14 July 2014

Academic Editor: Chengjin Zhang

Extreme learning machine (ELM) is a learning algorithm for single-hidden layer feedforward neural network dedicated to an extremely fast learning. However, the performance of ELM in structural impact localization is unknown yet. In this paper, a comparison study of ELM with least squares support vector machine (LSSVM) is presented for the application on impact localization of a plate structure with surface-mounted piezoelectric sensors. Both basic and kernel-based ELM regression models have been developed for the location prediction. Comparative studies of the basic ELM, kernel-based ELM, and LSSVM models are carried out. Results show that the kernel-based ELM requires the shortest learning time and it is capable of producing suboptimal localization accuracy among the three models. Hence, ELM paves a promising way in structural impact detection.

1. Introduction

Plate structure is able to provide some merits in terms of lightweight, high strength, and design flexibility. Hence, plate structures have been widely employed in aerospace, civil, and mechanical engineering. Even though the structures are designed to tolerate some damage, maintenance is needed to be performed to guarantee the structure integrity for safety reasons. One usual origin of structural damage arises from the impact exerted on the structures. Hence, it is important to detect the location and magnitude of the impact force [1, 2]. Such a detection process is beneficial to the subsequent maintenance work on the plate structures.

In the literature, various techniques have been developed to facilitate the impact localization of plate structures. Generally, these methods can be classified into active and passive approaches [3, 4]. This paper is focused on the passive approach owing to its nice efficiency. The passive method is implemented using the sensors attached to structures [5]. System modeling and data-driven techniques are two popular means in passive detection of structural damage. The former approach suffers from a major shortcoming; that is, an accurate system model is difficult to be established for complex structures, and the model error will be transferred as explicit localization errors. In contrast, the data-driven technique relies on the information of input-output data only, while it does not require the knowledge of system model. Hence, it is easier to realize and exhibits some advantages over the system modeling approach.

Data-driven techniques based on artificial intelligence (AI) are prevalent in the past two decades [6]. In particular, neural networks (NNs) have been popularly adopted in previous works to establish a regression model [7, 8]. However, it has been recognized that there is no universal method to determine an optimal NN structure in terms of the number of hidden layers and number of neurons in each layer [9]. Moreover, NN exhibits the risks of overfitting and sinking into local optima. Alternatively, support vector machine (SVM) is a promising tool to localize the impact on structures accurately [10, 11]. As compared with NN, SVM owns the major advantages of global optimization and higher generalization capability. Least squares support vector machine (LSSVM) simplifies traditional SVM by introducing equality constraints instead of inequality constraints [12]. The
potential of LSSVM in impact localization of plate structures has been demonstrated in the previous work [5].

For any AI-based data-driven algorithms, a practical concern is the learning time. Generally, an NN requires considerable learning time to generate an accurate regression model, which blocks its applications in rapid implementation scenarios. To overcome this issue, extreme learning machine (ELM) has been recently proposed to realize an extremely fast learning [13, 14]. ELM is a learning algorithm for single-hidden layer feedforward neural network. In theory, this algorithm is able to provide good generalization capability at extremely fast learning speed. Comparative studies of benchmark function approximation problems reveal that ELM can learn thousands of times faster than conventional NN and can produce good generalization performance in most cases.

ELM has been widely applied to solve different engineering problem, such as nonlinear hysteresis modeling [15]. However, the research on impact localization using ELM is rare in the literature. In this paper, the ELM is introduced into the domain of structural impact localization. Experimental study is carried out using LSSVM and ELM for the impact location detection of a plate structure. Comparative investigations in terms of learning time and localization accuracy are carried out in detail. It is shown that ELM paves a promising way in the domain of structural impact localization, because it provides the fastest learning time along with suboptimal localization accuracy.

The remainder of the paper is organized as follows. Section 2 presents the data processing technique that is employed in this work. Section 3 gives a brief review of the LSSVM and ELM algorithms. The experimental setup and data acquisition are described in Section 4. Section 5 conducts a collection of experimental studies using LSSVM and ELM approaches along with a detailed comparative study. Section 6 summarizes this paper.

2. Data Processing

In this research, the impacts are exerted on an aluminum structural plate. A set of four piezoelectric (PZT) sensors are mounted on the surface of the plate to acquire the induced acoustic waves. The sensor outputs are acquired with an NI PCI-4472 card using a sampling rate of 100 kHz. In order to reduce the data dimension, useful features are extracted from the time-history signal. Specifically, the magnitude of the maximum response and its corresponding time are employed because these two features are the best among the known features [8].

To extract the two features from each sensor output, the Hilbert transform is employed to solve the envelope of each sensor signal. Then, the two features of each sensor output are extracted. More details can be found in the literature [16].

The four sensors provide a total of eight features for the modeling of impact location. To generate a better modeling accuracy, it is suggested to normalize the input data into the range between −1 and 1 [13]. However, in this research, preliminary testing reveals that the normalization to zero mean and unit variance produces a better prediction result for the impact location problem. Hence, each of the eight signals $s$ is normalized through the transformation:

$$N(s) = \frac{s - \mu}{\sigma},$$  \hspace{1cm} (1)

where $\mu$ and $\sigma$ represent the mean and variance of the signal, respectively.

3. LSSVM and ELM Modeling

In this section, the employed LSSVM and ELM algorithms are briefly described.

3.1. LSSVM. Generally, given an input data, LSSVM maps it into a high-dimensional feature space and constructs a linear regression function therein. In this research, the LSSVM as described in [5] is employed. Moreover, the Bayesian inference is used to tune the hyperparameter in an optimal manner. More details about its implementation can be referred to in the literature [5, 17].

3.2. Basic ELM. Given $N$ arbitrary samples $(x_1, t_1)$ with $x_i \in R^n$ and $t_i \in R^m$, a standard single-layer feedforward neural network with $L$ hidden neurons and activation function $g(x)$ can be mathematically described as

$$\sum_{i=1}^{N} \beta_i g (w_i x_j + b_i) = d_j, \quad j = 1, \ldots, N,$$

where $w_i$ is the weight vector connecting inputs and the $i$th hidden neurons, $\beta_i$ is the weight vector connecting the $i$th hidden neurons and output neurons, $b_i$ is the threshold of the $i$th hidden node, and $d_j$ is the ELM output for the $j$th data point.

This network can approximate these $N$ samples with zero error $\sum_{j=1}^{N} \|d_j - t_j\| = 0$. That is, there exist $\beta_i$, $w_i$, and $b_i$ to meet

$$\sum_{i=1}^{N} \beta_i g (w_i x_j + b_i) = t_j, \quad j = 1, \ldots, N.$$  \hspace{1cm} (3)

The above $N$ equations can be expressed into a compact form:

$$H\beta = T,$$  \hspace{1cm} (4)

where

$$H = \begin{bmatrix} h (x_1) \\ \vdots \\ h (x_N) \end{bmatrix} = \begin{bmatrix} g(w_1 x_1 + b_1) & \cdots & g(w_L x_1 + b_L) \\ \vdots & \cdots & \vdots \\ g(w_1 x_N + b_1) & \cdots & g(w_L x_N + b_L) \end{bmatrix}_{N \times L}$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_L^T \end{bmatrix}_{L \times m}, \quad T = \begin{bmatrix} t_1^T \\ \vdots \\ t_L^T \end{bmatrix}_{N \times m}.$$
In ELM, the hidden-node parameters \((w_i, b_i)\) remain fixed once randomly generated. To train the network is simply equivalent to find a least square solution \(\hat{\beta}\) of the linear system (4); that is,

\[
\|H\hat{\beta} - T\| = \min_\beta \|H\beta - T\|.
\] (6)

The smallest norm least square solution to (4) is

\[
\hat{\beta} = H^\dagger T,
\] (7)

where \(H^\dagger\) is the Moore-Penrose generalized inverse of the matrix \(H\).

Therefore, given a training set \(\{(x_i, t_i) \mid x_i \in R^n, t_i \in R^m, i = 1, \ldots, N\}\), the basic ELM algorithm is very straightforward as stated below [13].

**Step 1.** Randomly generate hidden-node parameters \((w_i, b_i), i = 1, \ldots, L\).

**Step 2.** Calculate the hidden layer output matrix \(H\).

**Step 3.** Calculate the output weight vector \(\beta\) as \(\beta = H^\dagger T\).

3.3. Kernel-Based ELM. In the case that the hidden layer feature mapping \(h(x)\) is unknown to the user, a kernel matrix of ELM can be defined as follows:

\[
\Omega_{ELM} = HH^T : \Omega_{ELM_{i,j}} = h(x_i)h(x_j) = K(x_i, x_j).
\] (8)

The output function of ELM becomes

\[
f(x) = h(x)H^T \left( \frac{1}{\lambda} + HH^T \right)^{-1} T
\]

\[= \begin{bmatrix}
K(x, x_1) \\
\vdots \\
K(x, x_N)
\end{bmatrix}^T \left( \frac{1}{\lambda} + \Omega_{ELM} \right)^{-1} T.
\] (9)

In this kernel-based ELM, the hidden layer feature mapping \(h(x)\) needs not to be known to the user, while the kernel \(K(u, v) = \exp(-\gamma\|u - v\|^2)\) is given to the user. In addition, the number of hidden nodes \(L\) needs not to be specified.

Hence, given a training set \(\{(x_i, t_i) \mid x_i \in R^n, t_i \in R^m, i = 1, \ldots, N\}\), the kernel ELM algorithm is further simplified as a single learning step implementation as follows [14].

**Step 1.** Calculate the output function:

\[
f(x) = \begin{bmatrix}
K(x, x_1) \\
\vdots \\
K(x, x_N)
\end{bmatrix}^T \left( \frac{1}{\lambda} + \Omega_{ELM} \right)^{-1} T.
\] (10)

4. Experimental Setup and Data Acquisition

4.1. Experimental Setup. The experimental setup consists of a clamped aluminum plate of size 490 mm \(\times\) 390 mm, four piezoelectric sensors glued on the plate surface, a data acquisition card plugged in a personal computer, and an instrumented hammer. The experimental setup is shown in Figure 1.

The four sensors are mounted on the plate surface by conductive glue. The distances between sensors and the nearest edges are 170 mm (horizontally) and 100 mm (vertically), respectively. The dimension of the plate and the location of the glued sensors are shown in Figure 2.

For data acquisition, an NI PCI-4472 data acquisition card was plugged in a PCI slot of a personal computer. The
time-varying data were gathered by a developed LabVIEW program. Preliminary experiment shows that when the hammer knocks on the plate, the voltages collected by sensors increase rapidly initially and then decay to zero rapidly. During this transient response period, the sensor output voltage signals were recorded and saved in data files. The data acquisition rate is set as 100 kHz.

4.2. Data Acquisition. Through experimental impact testing, two sets of impact data were acquired with the aforementioned experimental setup. The first set includes a regular grid of 63 impacts, as shown in Figure 3. It is used for the training of the LSSVM and ELM models. Another set of 90 random impacts, as shown in Figure 4, is employed for the model testing.

The data acquisition and impact localization were executed on a personal computer (PC) equipped with 2.83 GHz Quad CPU and 8 GB RAM running Windows 7 (64-bit) operating system.

4.3. Performance Index. To quantify the localization accuracy, an error function is employed to provide a measure of the accuracy. Specifically, to visualize the results, the averaged errors of $x$ and $y$ coordinates are calculated, and the ratio between the product of them and the plate area is considered as the evaluation index:

$$\eta = \frac{\sum_{i=1}^{N} (x - x_i) \sum_{i=1}^{N} (y - y_i)}{N^2 A},$$

(11)
where $A$ denotes the area of the plate structure, and $\eta$ is the ratio between the error area and the plate area. It is found that the lower the index is, the better the detection accuracy for the impact location will be.

5. Results and Discussion

5.1. LSSVM Results. Using the approach as described previously, the LSSVM model is trained with the training set as shown in Figure 3 and then tested with the testing set as illustrated in Figure 4. Moreover, two LSSVM models are generated to predict the $x$ and $y$ coordinates, respectively. To get a reliable result, the LSSVM model is trained by 30 times. The training error of a typical result is shown in Figure 5, which leads to a low training error index $\eta = 0.0278\%$.

Using the two trained LSSVM models, the testing results for $x$ and $y$ coordinates are shown in Figure 6(a). In addition, the discrepancies between the predicted and actual $x$ and $y$ coordinates are illustrated in Figure 6(b). It is calculated that a testing error index of 0.7358% is resulted for the 90 sets of testing data samples.

5.2. Basic ELM Results. Concerning the basic ELM algorithm, the activation function is selected as the sigmoid function. Two ELM models are established for the prediction of the $x$ and $y$ coordinates, respectively.

As the number of hidden nodes increases, the training and testing accuracy of the basic ELM models for the $x$ and $y$ coordinate predictions are depicted in Figures 7(a) and 7(b), respectively. It is found that when the number $L = 65$, which is close to the number (63) of training samples, the training error becomes zero, whereas the testing error is greatly increased. To generate stable results for the two basic ELM models, the number of hidden nodes is selected as $L = 100$.

In addition, the training of each model is conducted by 30 times. The training errors of a typical result are illustrated in Figure 8. It is found that the ELM models produce a zero training error index approximately. This indicates that the basic ELM exhibits a very nice function approximation capability.

Using the trained basic ELM models, the testing results are generated as shown in Figure 9. For a clear visualization, the testing errors of the two coordinates are depicted in Figures 9(a) and 9(b), respectively. It is calculated that the ELM models create a large testing error index of 3.8980%.

5.3. Kernel ELM Results. Regarding the kernel ELM model, the radial basis function (RBF) kernel is selected owing to its popularity. Two kernel ELM models are developed for the detection of $x$ and $y$ coordinates of the impact. By selecting
the regularization coefficient $\lambda = 1$, preliminary results indicate that the training and testing performances of ELM models are dependent on the RBF parameters.

As the RBF parameter increases, the training and testing accuracies for the $x$ and $y$ coordinates are shown in Figures 10(a) and 10(b), respectively. It is seen that the training errors increase with the rising of the RBF parameter. Whereas the testing errors arrive at the minimum values when the RBF parameters are 65 and 20 for $x$ and $y$ coordinates, respectively. Hence, to produce better testing results, the RBF parameters of the two ELM models are selected as 65 and 20 for the prediction of $x$ and $y$ coordinates, respectively.

By running each model 30 times, it is found that each time leads to the same result. The training errors of the two kernel ELM models are illustrated in Figure 11, which results in a training error index of 2.0154%. Using the two trained kernel ELM models, the testing results for the two coordinates are obtained, as shown in Figure 12(a). The corresponding testing errors of the $x$ and $y$ coordinate predictions are shown in Figure 12(b). The kernel ELM models produce the testing error index of 0.7746%.

5.4. Discussion on Localization Results. For a clear comparison of the detection performance of the LSSVM, basic ELM (BELM), and kernel ELM (KELM) models, the results are tabulated in Table 1. The training time means the time required by each model for the model training for a single coordinate prediction. The testing time is the execution time needed by each model to predict a single coordinate of the impact.
Table 1: Comparison among LSSVM, basic ELM (BELM), and kernel ELM (KELM) results.

<table>
<thead>
<tr>
<th>Model</th>
<th>Training time (s)</th>
<th>Training error index (%)</th>
<th>Testing time (s)</th>
<th>Testing error index (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSSVM</td>
<td>0.1438</td>
<td>0.0278</td>
<td>0.0098</td>
<td>0.7358</td>
</tr>
<tr>
<td>BELM</td>
<td>0.1248</td>
<td>0.0000</td>
<td>0.0468</td>
<td>3.8980</td>
</tr>
<tr>
<td>KELM</td>
<td><strong>0.0134</strong></td>
<td>2.0154</td>
<td><strong>0.0007</strong></td>
<td>0.7746</td>
</tr>
</tbody>
</table>

By comparing the training and testing time, it is evident that the KELM requires the minimum time for both training and testing processes. Whereas the LSSVM needs the maximum time for the model training and BELM calls for the longest execution time for the impact localization.

In addition, given the training samples, it is deduced from Table 1 that the BELM is able to produce the minimum training error, which is equal to zero approximately. In contrast, the KELM creates the maximum training error. Therefore, the BELM provides an excellent function approximation capability.

Moreover, from the testing accuracy of the three models, it is found that the LSSVM is able to deliver the most accurate localization result. It is further observed that the BELM gives the lowest accuracy of the impact localization although it produces the best training error. By comparing the testing accuracy of KELM and LSSVM, it is seen that the KELM localization error is only 5.3% higher than that of LSSVM.

In summary, the KELM produces the suboptimal localization accuracy among the three models, while it requires the shortest training and testing execution time. Therefore, the KELM is more suitable for the scenarios where the execution time is the most critical requirement, whereas the LSSVM is more appropriate to the situations when the localization accuracy is most desirable.

6. Conclusions

This paper presents a comparison study of LSSVM and ELM dedicated to impact localization of a plate structure with surface-mounted PZT sensors. By extracting useful features through Hilbert transform of each sensor output, regression
models are established to predict the location of impact exposed by a hammer. Both basic ELM and kernel-based ELM algorithms are employed for comparison investigation with respect to LSSVM approach. Experimental results reveal that the basic ELM is able to provide a nice function approximation capability. The kernel ELM requires the minimum execution time for training and testing processes, while LSSVM is able to produce the most accurate localization result. In addition, kernel ELM gives the suboptimal localization accuracy. Therefore, the kernel ELM is more suitable for the scenarios where the execution time is the most critical requirement, whereas the LSSVM is more appropriate to the situations when the localization accuracy is most desirable.

In the future, more advanced ELM algorithm will be attempted to further improve its localization accuracy while maintaining the smallest execution time. Moreover, its application in real impact localization of aircraft structures will be carried out in the future work.

Conflict of Interests

The author declares that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported by the Macao Science and Technology Development Fund under Grant 070/2012/A3 and the Research Committee of the University of Macau under Grants MYRG083(Y1-L2)-FST12-XQS and MYRG078(Y1-L2)-FST13-XQS.

References
