Neural Comput & Applic
DOI 10.1007/s00521-014-1568-2

Fast detection of impact location using kernel extreme learning machine

Heming Fu · Chi-Man Vong · Pak-Kin Wong · Zhixin Yang

Received: 10 September 2013 / Accepted: 11 March 2014

Abstract Damage location detection has direct relationship with the field of aerospace structure as the detection system can inspect any exterior damage that may affect the operations of the equipment. In the literature, several kinds of learning algorithms have been applied in this field to construct the detection system and some of them gave good results. However, most learning algorithms are time-consuming due to their computational complexity so that the real-time requirement in many practical applications cannot be fulfilled. Kernel extreme learning machine (kernel ELM) is a learning algorithm, which has good prediction performance while maintaining extremely fast learning speed. Kernel ELM is originally applied to this research to predict the location of impact event on a clamped aluminum plate that simulates the shell of aerospace structures. The results were compared with several previous work, including support vector machine (SVM), and conventional back-propagation neural networks (BPNN). The comparison result reveals the effectiveness of kernel ELM for impact detection, showing that kernel ELM has comparable accuracy to SVM but much faster speed on current application than SVM and BPNN.

Keywords Damage location detection · Kernel ELM · Plate structure

H. Fu · C.-M. Vong (✉)
Department of Computer and Information Science, Faculty of Science and Technology, University of Macau, Taipa, Macau
e-mail: cmvong@umac.mo

P.-K. Wong · Z. Yang
Department of Electromechanical Engineering, Faculty of Science and Technology, University of Macau, Taipa, Macau

1 Introduction

Damage detection, which contains *location detection* and *severity of damage*, has extensive application prospect in many practical applications such as aerospace structure because it is very important in clarifying and diagnosing the location and severity of damage for reducing the accidents and keeping the health of structure [1, 2]. Recently, a research [3] verified that the impact events on composite plates can simulate the real impact situations of the aircraft panels made of composite materials. It was also indicated [3] that neural network-based location detection strategies, which were developed for simple structures, were applicable to more complex structures such as aircraft component with a high degree of material and geometrical complexity. However, due to the expense of the experimental structure, only nondestructive impact events for a neural network were considered in the work [3]. Meanwhile, except for nondestructive impact events, destructive impact events were also considered. Watkins et al. [4] used the *strain signatures* caused by impact-induced damage on fiber-reinforced composite plate to classify the type and extent of damage. In this paper, the authors used a large amount of plates to eliminate the effect of previous impact event because kinetic energy may cause damage to the structure. However, it is very expensive to prepare so many plates for research on that kind of destructive impact tests. In order to guarantee the consistency of experimental conditions and parsimony of funding, we just concentrated on the nondestructive impact events in this paper.

In the literature, various learning algorithms have also been applied to predict the impact location [3–12]. For example, Jones et al. [5] used back-propagation neural network (BPNN) with experimental data acquired from fully clamped plate with sensors to explore the location and

Published online: 23 March 2014

magnitude estimation. In order to improve the accuracy of results, they explored several methods such as *removing DC (direct current) shift and high-frequency noise*, and *only using impacts within the range of training set*. It was mentioned previously [3] that a neural network-based strategy that integrates classification and regression produced a better performance than methods that only used classification or regression. However, all of these methods require the determination of optimal numbers of neurons in hidden layer, and also the time of adjusting network weights is very long. In addition, the execution time of the learnt network is usually too long for a real-time application. Hence, it is much desired to explore a prediction model with less computation while maintaining the accuracy.

Except artificial neural networks, other learning algorithms have also been applied to location detection. For instance, Worden et al. [12] explored the optimal sensor placement by using genetic algorithms before they built their impact detection system. Besides, the least squares support vector machines (LS-SVM) were extensively used in some engineering applications [13, 14]. Meanwhile, an enhanced LS-SVM [15] approach using Hilbert transform was proposed to reinforce the convexity and computational speed due to the problem that the conjugate gradient method cannot solve the Karush–Kuhn–Tucker system of LS-SVM directly. However, about current application, most of the existing LS-SVM-based studies only focused on the classification problems but not regression like the current one. In our previous work [16], we integrated the principal components analysis (PCA) and SVM to predict the impact location. The comparison between the results of SVM and that of BPNN with the same experimental data revealed that SVM has better generalization performance than BPNN on impact detection. Nevertheless, the execution time of the prediction model (PCA + SVM) does not perfectly fulfill the real-time requirement for aerospace structures.

Extreme learning machines (ELM) [17–20] as generalized single-hidden layer feedforward networks (SLFNs) are proposed to overcome these problems. The basic concept of ELM is that the parameters of its hidden layer, including the input weights and biases of the hidden nodes, are unnecessary to be tuned [17, 19]. Moreover, these parameters are assigned randomly. Compared with other learning algorithms, ELM has drawn growing concern from researchers as it provides better generalization performance at a much faster speed caused by less computational complexity. As a result, many variants have been proposed to solve different problems in the literature, such as *kernel-based ELM* [21], *fully complex ELM* [22, 23], *online sequential ELM* (OS-ELM) [24, 25], and *incremental ELM* (I-ELM) [19, 26, 27]. Among them, kernel ELM [21], where the hidden layer feature mapping is determined by a kernel matrix and the number of hidden nodes (the dimensionality of the hidden layer feature space) are also not needed to be specified, has been verified to achieve a better and more stable performance than basic ELM. Moreover, the kernel ELM is proved to have similar generalization performance to SVM while maintaining a much faster learning (and even execution) speed. Therefore, kernel ELM is employed in the current research. The nondestructive impact data acquired from clamped plate are used for training and test. To reduce the computational cost, PCA is used to extract principal components with minimal information loss from the acquired impact data, which is different from the method applied in Jones's paper [5]. Jones et al. [5] adopted the real and imaginary parts of the FFT results as input to the back-propagation neural networks for the same reason. Then, those authors also chose removing the direct current (DC) shift and high-frequency noise to improve the prediction accuracy. However, in this research, kernel ELM is trained with the extracted principal components by PCA, which show a better result than that in [5]. To evaluate the performance on accuracy and time, the predicted results are also compared with the results of BPNN and SVM, respectively. This paper is arranged as follows. Section 2 gives an overview of the physical facilities, details about data acquisition, and the procedure of data processing. Section 3 presents the criteria for comparison, the results, and evaluations of all the algorithms applied in the research. Finally, conclusion is given in Sect. 4.

2 Experimental setup

2.1 Physical facilities

In this research, the panel on real aerospace structure is simulated by a composite plate. The experimental equipment consists of a rectangular 491 mm × 392 mm clamped plate, four sensors pasted on the plate by conductive glue, a data acquisition card assembled in computer, and a hammer with rubber head for knocking. The methods of connection are shown in Fig. 1.

The plate is drilled in four corners and fastened on a heavy rectangular metal base that is larger than it. The distances between every two neighbored sensors are 170 mm horizontally and 100 mm vertically. The positions of sensors are based on the researches about optimal sensor placement in the literature [5, 12]. The dimension of the plate and the locations of sensor are provided in Fig. 2.

Data acquisition card (NI PCI-4472) manufactured by National Instruments is integrated in the computer and the time-varying data (i.e., vibration signal after knocking/

impact) can be collected by simple program written with corresponding software LabVIEW. As the sensors are sensitive to plate vibration, to avoid apparent noise, the experiment environment must be kept quiet in a sense. In this situation, when the hammer knocks on the plate, the voltages collected by sensors would increase rapidly. During the period of one knocking, the variety of voltage values is recorded in an independent file. Each file represents the information about one impact, i.e., a training data point, whose format is presented in Fig. 3. The acquisition rate is 100,000 Hz, and there are $6 \times 10{,}000$ sampled values for one impact.

2.2 Data acquisition

All experimental data applied in this research are acquired from this set of experimental facilities while the clamped plate is knocked (or impacted). The vibration signals were collected through the four sensors while an impact event occurred. The experimental data consist of the time-dependent vibration signals after impact event, together with the impact location (i.e., *x*- and *y*-coordinates).

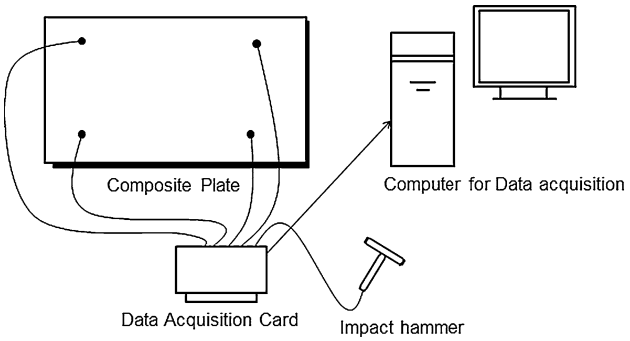

Fig. 1 Experimental setup for impact events

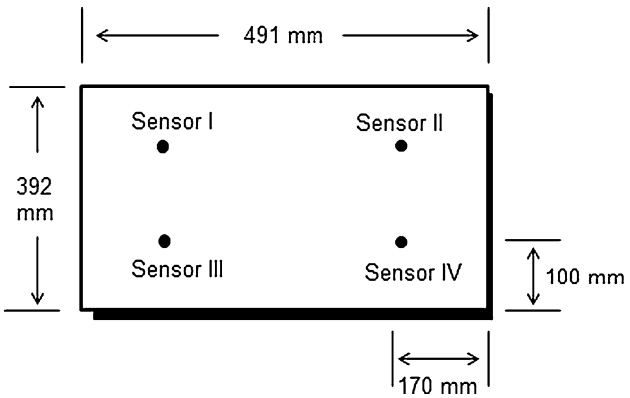

Fig. 2 Dimensions of plate and location of sensors

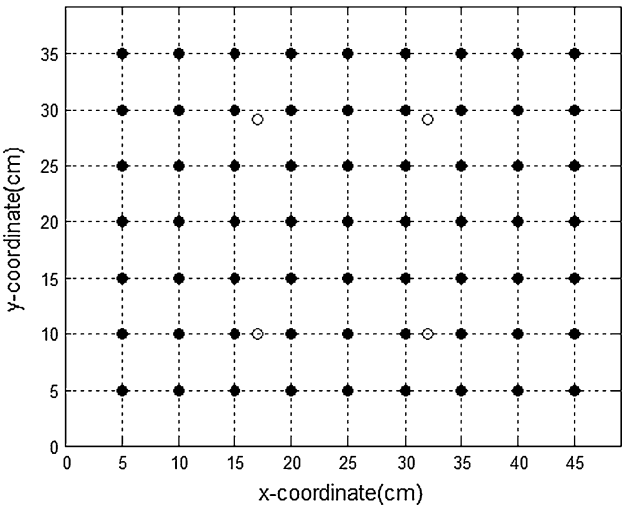

Fig. 4 Distribution of training impact locations. *Filled circle* training impact locations and *open circle* sensor locations

In this research, only two sets of experimental data are required. The first one is the training dataset (Fig. 4), which consists of 63 impact regions on the crossing of grids with interval of 5 cm (nine equally distributed horizontally and seven equally distributed vertically). Since the plate in the current study is a mock of the components on real aerospace structure that are usually very large, it is practically precise enough to employ 5-cm interval between training data. Another set of 100 impact events is the test dataset (Fig. 5), in which impact locations are acquired randomly.

2.3 Format of experimental data

As shown in Fig. 3, a data file contains the information about one impact event. There are totally six variables and 10,000 instances in one data file. The first and second columns represent time stamps and voltage values sent back from hammer, respectively, that can be neglected because of no contribution to location detection. The remaining four variables are the sensor readings in voltage. By combining 10,000 instances of the four sensor readings,

Fig. 3 Acquired senor values for one impact (one training data)

Time	hammer	sensor1	sensor2	sensor3	sensor4
0.000000	0.001527	0.002631	0.002451	-0.002360	-0.000214
0.000001	0.001395	0.004278	0.001466	0.003006	-0.030697
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
……	……	……	……	……	……

10000 instances

there are totally 40,000 values for one impact event. With all these values and the corresponding impact location as target, the format of experimental datasets in current research is determined. However, the huge number of attributes for an impact event will affect the prediction time and accuracy. Hence, PCA was applied to extract the principal components from the 40,000 attributes. After PCA, 88 principal components retaining 95 % of information (details can be referred to [16]) were extracted as the features of impact event, followed by training and testing the prediction model using kernel ELM. We have tried to use a single model to predict both *x*- and *y*-coordinates but experimental results (not listed in this paper) verified that an independent model for a coordinate has higher prediction accuracy. In order to more accurately predict the impact location, the *x*- and *y*-coordinates are, respectively, predicted using two independently trained models, and hence two corresponding training datasets were constructed. The workflow of preparing experimental dataset is illustrated in Fig. 6.

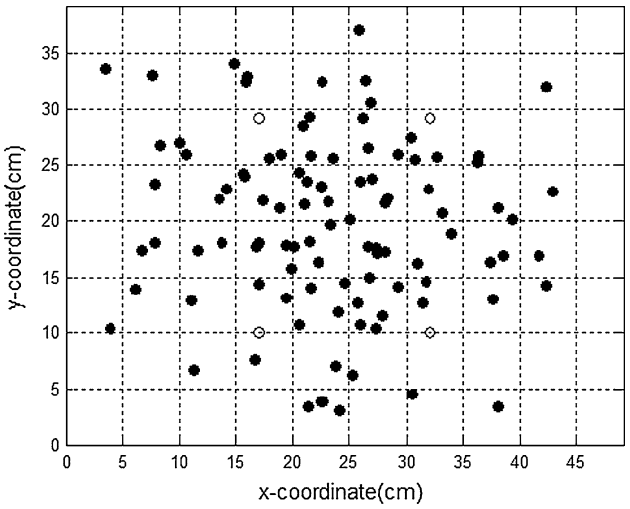

Fig. 5 Distribution of test impact locations. *Filled circle* test impact locations and *open circle* sensor locations

As shown in Fig. 6, the extracted features of all impact events were combined with the corresponding measured *x*-coordinates to construct the training dataset for the prediction of *x*-coordinate. The training dataset for the prediction of *y*-coordinate is constructed in a similar way.

2.4 Model training

As discussed previously, 88 principal components were chosen as the features of an impact event. Before model training, the principal components and coordinates (*x*- or *y*-coordinate) were, respectively, normalized to the range of $[-1, 1]$ by using the corresponding maximum and minimum values (in this research, they are 40 and -40 for the PCA results, 45 and 5 for coordinates separately, which can transform the PCA results and coordinates into the fixed range to easily show their relationship) from all the components and coordinates in the training dataset. The

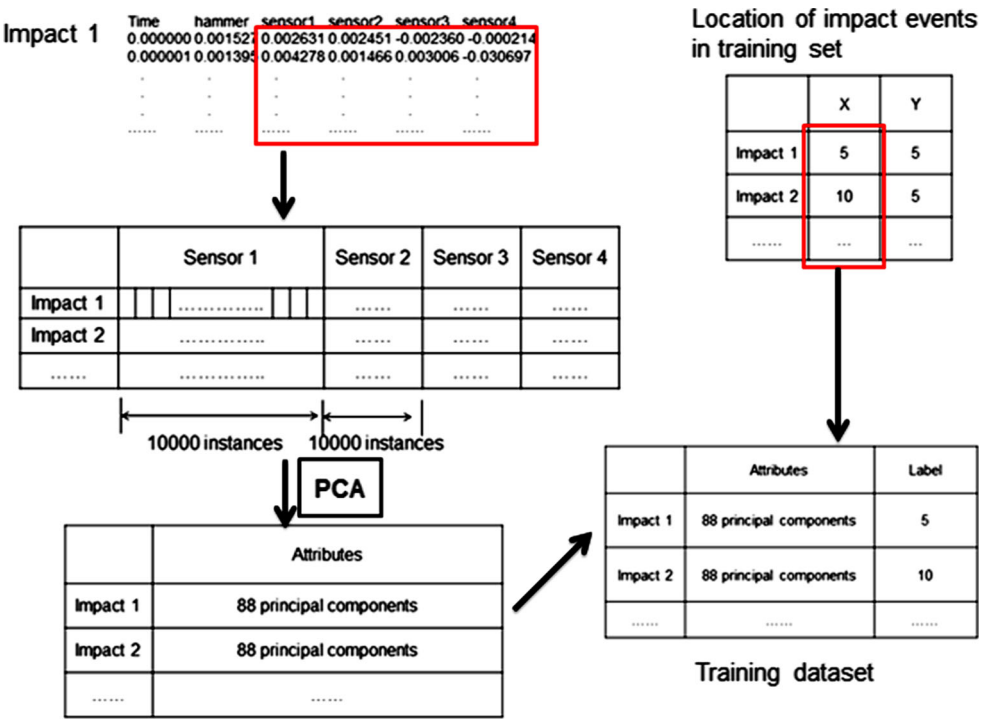

Fig. 6 Workflow of converting raw data into training dataset

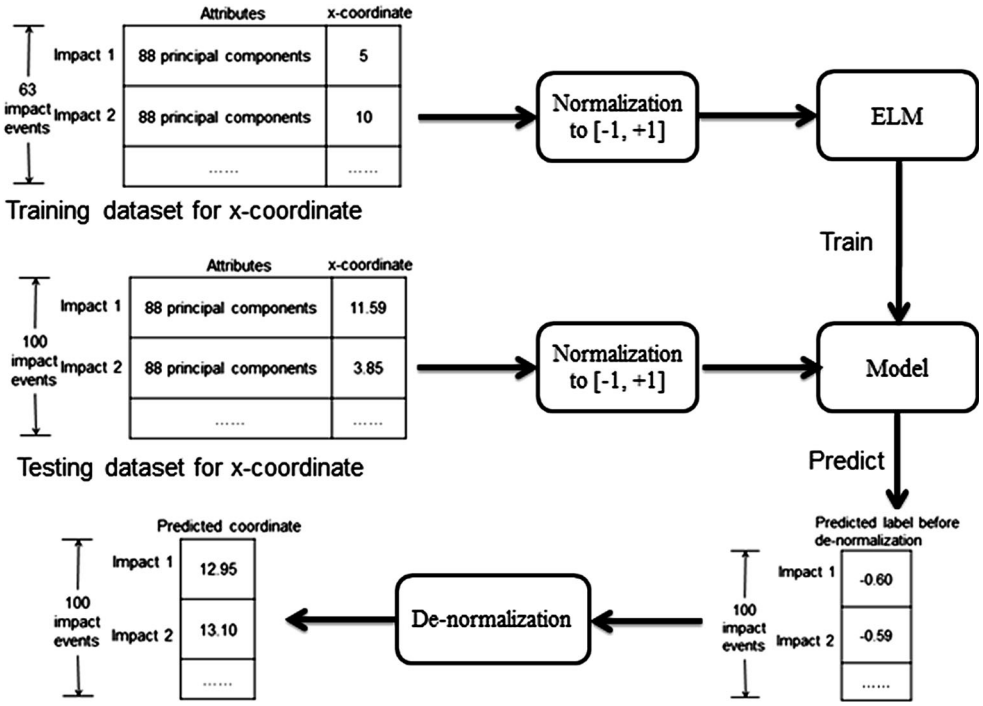

Fig. 7 Workflow of training and testing procedure for x- or y-coordinate

training and test procedure was implemented with MATLAB toolboxes of PCA and kernel ELM [28]. In this study, RBF kernel was chosen. In order to explore the optimal parameters for kernel ELM model, grid search and ninefold cross-validation methods were employed. The range of grid search was set as 2^{-10}–2^{10} with a small exponent interval of 0.1. The workflow of prediction for x-coordinate is presented in Fig. 7.

The impact events in test set were processed in the similar way. The values of four sensor readings were processed with the same PCA transformation matrix of training set and normalized within the same range of $[-1, 1]$. After prediction, the results were de-normalized to original range.

3 Experimental results

In the previous work [5], several evaluation criteria to quantify the detection performance in current application were employed such as MAE and RMSE functions, expressed as

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n} |v - v_i| \tag{1}$$

$$\mathrm{RMSE} = \frac{1}{n}\sqrt{\sum_{i=1,t=1}^{n}\left((x_t - x_i)^2 + (y_t - y_i)^2\right)} \tag{2}$$

where n is the number of test impact events, v and v_i are the measured coordinate and predicted coordinate, respectively, (x_t, y_t) are the measured coordinates of the impact locations within the test set, and (x_i, y_i) are the predicted results. The RMSE function gives the radial distance between the real and the predicted impact location in centimeters. An alternative approach for evaluation in Worden's paper [12] employs *error area ratio* for better visualizing the results. This evaluation is expressed as

$$P = \frac{\frac{1}{n^2}\sum_{i=1,t=1}^{n}(x_t - x_i)\sum_{i=1}^{n}(y_t - y_i)}{S} \tag{3}$$

where S denotes the plate area, P is the ratio of error area to the whole plate. A smaller ratio P indicates better prediction accuracy.

3.1 Results of kernel ELM

By employing the collected and preprocessed datasets and training techniques described in Sect. 2, a kernel ELM model was constructed. The general experimental results of kernel ELM are shown in Table 1. After analyzing the prediction results over the test set of 100 impact events, the MAE of x- and y-coordinates are 3.5433 and 2.8073 cm, respectively, according to Eq. (1), with RMSE in Eq. (2) equals to 0.7317 cm. Meanwhile, an area error using Eq. (3) is 0.52 % of the plate area. The standard deviation for these error measures is 0.0 due to property of kernel ELM. The total time consumption is 19.8 ms including the training and testing time for x- and y-coordinates except the time cost of PCA.

The difference and error between every pair of the measured and predicted x- and y-coordinates are shown in Figs. 8 and 9, respectively, where the x-axis stands for the

Table 1 Results of kernel ELM over original test set

Model	MAE (cm)		Error area (%)	RMSE (cm)	Training time (ms)		Test time (ms)		Total time (ms)
	x	*y*			*x*	*y*	*x*	*y*	
K-ELM	3.54 ± 0.0	2.81 ± 0.0	0.52 ± 0.0	0.7317 ± 0.0	6.0 ± 0.2	8.4 ± 0.3	1.9 ± 0.2	3.5 ± 0.2	19.8 ± 0.6

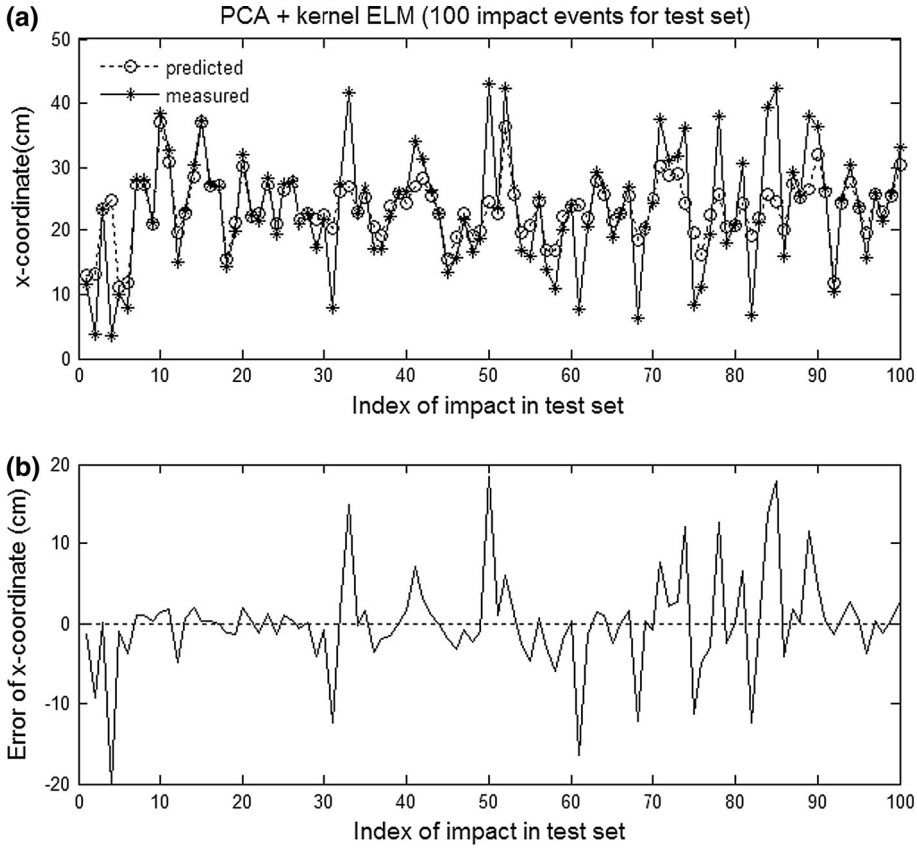

Fig. 8 **a** Differences between measured *x*-coordinates and predicted results (100 impact events for test). **b** Errors of predicted and measured *x*-coordinates (100 impact events for test)

indices of the 100 impact events in test set. From Figs. 8b and 9b, it can be observed that some specific impact locations (e.g., 4th, 50th in Fig. 8b; 70th, 89th in Fig. 9b) exhibit large errors. According to the previous work [5], this phenomenon is caused due to the fact that the coordinates of these test impact locations are outside the boundaries of the training set. Generally, when a prediction model is trained with data drawn from a specific space, it is less likely to accurately predict the unknown data out of that space.

An alternative evaluation proposed from [5] is to prune the impact events outside of the training boundaries from the test set. This evaluation is reasonable because on practical situation, the surface of the aerospace structure consists of numerous plates, on each of which several sensors are mounted, so that no out-of-boundary impact event may occur. Therefore, the prediction accuracy should be evaluated again over a test set without those out-of-boundary impact events as shown in Fig. 10.

With this new test set, the experimental results are shown in Table 2. The kernel ELM model gives the MAE of *x*- and *y*-coordinates as 2.3069 and 1.9748 cm, respectively, while results in RMSE of 0.5025 cm, and the area of error occupies 0.24 % of the plate. Besides, the total time consumption is 14.9 ms, which includes the training and testing time for *x*- and *y*-coordinates except the time taken by PCA. Table 2 also presents the results of SVM and BPNN, and the details of these results are described in Sect. 3.2. As shown in Figs. 11 and 12, the differences and errors between the measured and predicted coordinates were improved.

3.2 Comparison with other methods

To further evaluate the performance of kernel ELM on impact location detection, SVM and BPNN were adopted for comparison. The same training dataset and pruned test set (with 91 impact events) were employed. The same

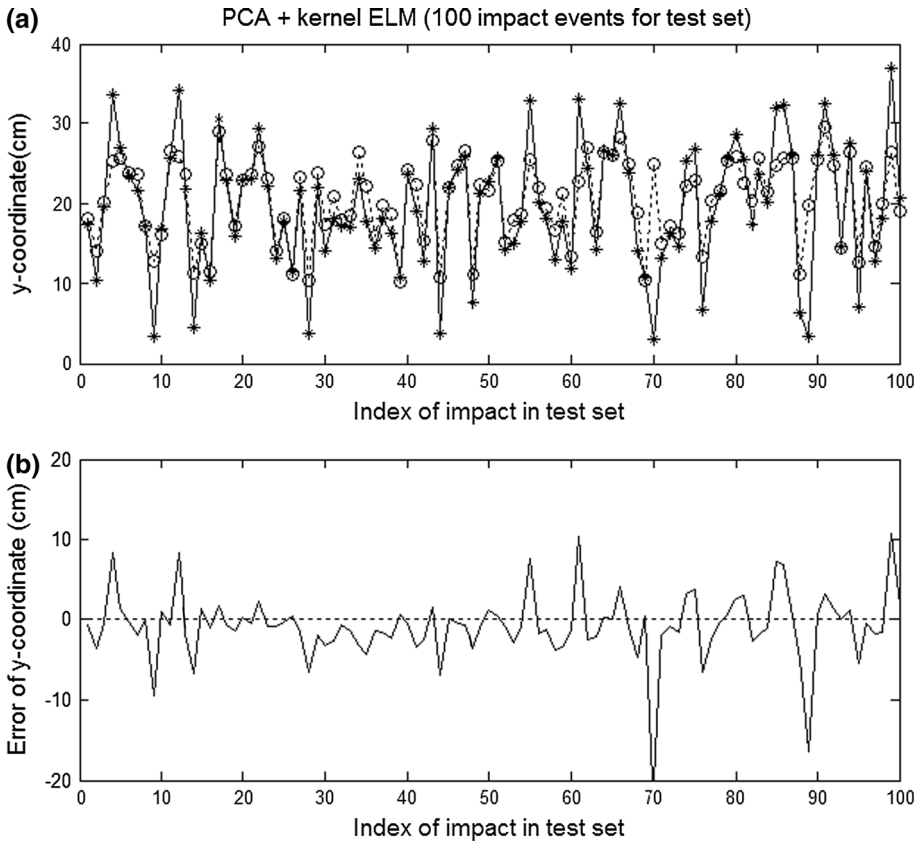

Fig. 9 **a** Differences between measured *y*-coordinates and predicted results (100 impact events for test). **b** Errors of predicted and measured *y*-coordinates (100 impact events for test)

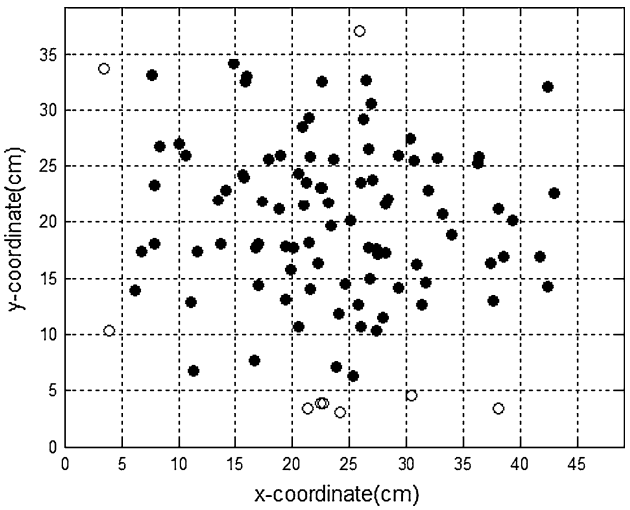

Fig. 10 Locations of test impact events after deletion. *Filled circle* test impact locations and *open circle* removed impact locations

normalization and preprocessing (i.e., PCA) techniques were applied. The development of SVM models can be referred to author's previous study [16], and grid search method coupled with cross-validation algorithm is chosen for exploring the parameters. Within a chosen range of 2^{-10} to 2^{10} with the exponent interval of 0.1, the optimal parameters were selected through ninefold cross validation again. For BPNN, there are 88 inputs, 11 hidden neurons (determined through an exhaustive search), and 2 outputs corresponding to *x*- and *y*-coordinates. The predicted results of SVM and BPNN are listed in Table 2 with comparison to the results of ELM. In Table 2, the training time just counts the time of building model by using the normalized training dataset for *x*- and *y*-coordinates, respectively. The test time counts the time taken to predict the *x*- and *y*-coordinates of the corresponding normalized test set.

Table 2 Comparison among kernel ELM, SVM, BPNN over pruned test set

Model	MAE (cm)		Error area (%)	RMSE (cm)	Training time (ms)		Test time (ms)		Total time (ms)
	x	*y*			*x*	*y*	*x*	*y*	
K-ELM	2.31 ± 0.0	1.97 ± 0.0	0.24 ± 0.0	0.5025 ± 0.0	5.8 ± 0.3	5.4 ± 0.2	2.7 ± 0.2	1 ± 0.6	14.9 ± 3.3
SVM	1.83 ± 0.0	1.35 ± 0.0	0.13 ± 0.0	0.343 ± 0.0	15 ± 1.3	13.6 ± 1.4	3.9 ± 0.4	3.8 ± 0.5	36.2 ± 3.2
BPNN	4.63 ± 0.83	3.94 ± 0.6	0.95 ± 0.22	0.87 ± 0.11	453 ± 36.4	448.1 ± 33.5	7 ± 0.9	7.1 ± 1	915.2 ± 45.5

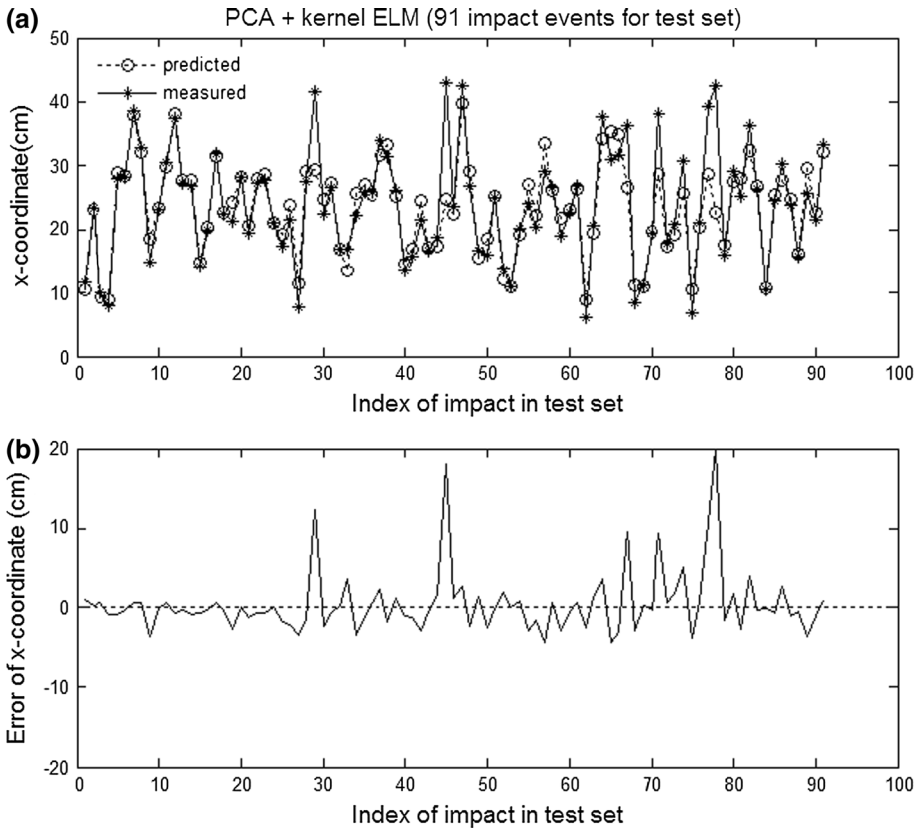

Fig. 11 **a** Differences between measured *x*-coordinates and predicted results (91 impact locations for test). **b** Errors of predicted and measured *x*-coordinates (91 impact locations for test)

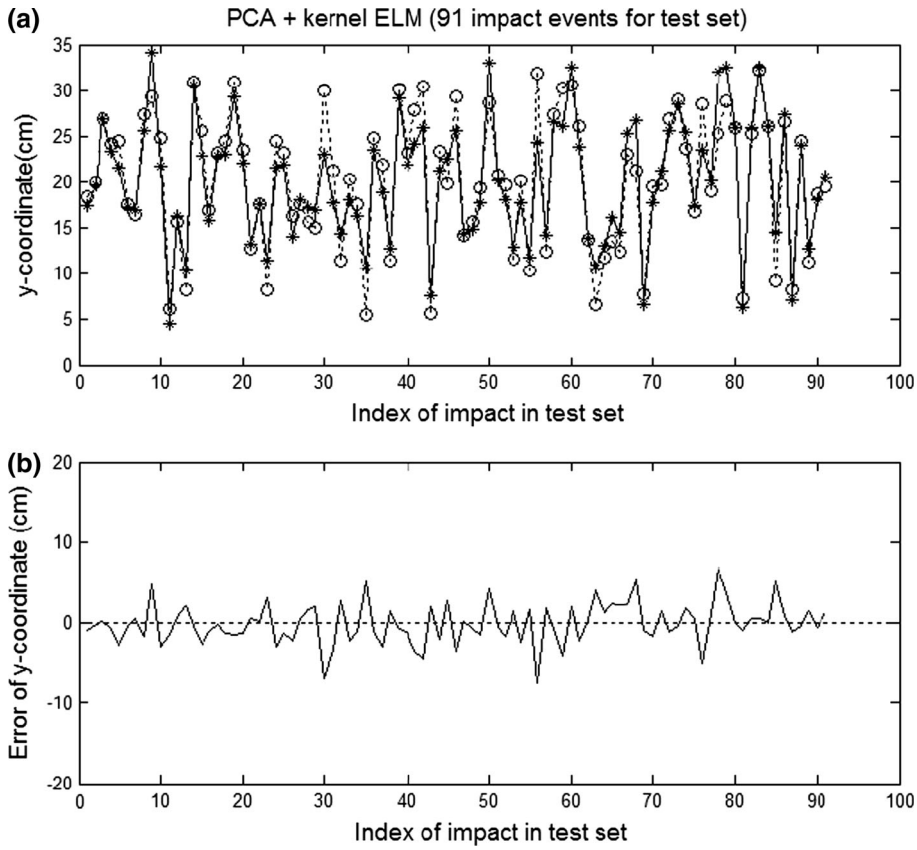

Fig. 12 **a** Differences between measured *y*-coordinates and predicted results (91 impact locations for test). **b** Errors of predicted and measured *y*-coordinates (91 impact locations for test)

Table 2 illustrates that kernel ELM has significantly higher accuracy than BPNN on location detection. Even though kernel ELM cannot catch up with SVM on the accuracy, the difference is insignificant for the current application. In addition, kernel ELM has apparent advantage on computational time over the other two methods. From Table 2, the total time of kernel ELM is about 1.43 times faster than that of SVM, which shows the importance in real-time requirement of the impact location detection system. This improvement of time implies that the location detection system can simultaneously monitor more panels on the aerospace structure, or can spend less time on estimating impact location, so that a cheaper system can be achieved. Both issues achieve economic benefits.

In summary, the predicted results of kernel ELM reveal that this algorithm is very suitable for current real-time application. Furthermore, kernel ELM has similar prediction accuracy to SVM but much better accuracy than BPNN.

4 Conclusion

This paper originally applies kernel ELM to predict the impact location occurred on a plate, which is illustrative important example for the topics of aircraft health monitoring and maintenance. Kernel ELM is employed to predict the x- and y-coordinates of impact locations. The speed of kernel ELM is 1.43 times and 35.2 times faster than SVM and BPNN, respectively, while kernel ELM can still attain a comparable prediction accuracy of SVM but far better than that of BPNN. From these results, it can be concluded that kernel ELM is very suitable to the application of location detection, which has a tough real-time requirement.

Acknowledgments This work is co-supported by FDCT Macau SAR, Grant No. FDCT/075/2013/A, and the research grants of University of Macau, Grant Nos. MYRG075(Y1-L2)-FST13-VCM and MYRG075(Y2-L2)-FST12-VCM. The authors would also like to express their sincere gratitude to Dr. Q. S. Xu and Mr. Kehn Wong for offering the experiment devices and technical support.

References

1. Sohn H, Farrar CR, Hunter NF, Worden K (2001) Structural health monitoring using statistical pattern recognition techniques. Trans Am Soc Mech Eng J Dyn Syst Meas Control 123:706–711
2. Mujica LE, Vehi J, Staszewski W, Worden K (2008) Impact damage detection in aircraft composites using knowledge-based reasoning. Struct Health Monit 7:215–230
3. LeClerc J, Worden K, Staszewski W, Haywood J (2007) Impact detection in an aircraft composite panel: a neural-network approach. J Sound Vib 299:672–682
4. Watkins SE, Akhavan F, Dua R, Chandrashekhara K, Wunsch DC (2007) Impact-induced damage characterization of composite plates using neural networks. Smart Mater Struct 16:515
5. Jones RT, Sirkis JS, Friebele E (1997) Detection of impact location and magnitude for isotropic plates using neural networks. J Intell Mater Syst Struct 8:90–99
6. Jones RT, Sirkis JS, Friebele EJ, Kersey AD (1995) Location and magnitude of impact detection in composite plates using neural networks. In: Smart structures and materials' 95, pp 469–480
7. Sung D-U, Oh J-H, Kim C-G, Hong C-S (2000) Impact monitoring of smart composite laminates using neural network and wavelet analysis. J Intell Mater Syst Struct 11:180–190
8. Friswell MI, Penny JET, Garvey SD (1998) A combined genetic and eigensensitivity algorithm for the location of damage in structures. Comput and Struct 69:547–556
9. Okafor AC, Otieno AW, Dutta A, Rao VS (2001) Detection and characterization of high-velocity impact damage in advanced composite plates using multi-sensing techniques. Compos Struct 54:289–297
10. Maseras-Gutierrez MA, Staszewski WJ, Found MS, Worden K (1998) Detection of impacts in composite materials using piezoceramic sensors and neural networks. pp 491–497
11. Yan G, Zhou L (2009) Impact load identification of composite structure using genetic algorithms. J Sound Vib 319:869–884
12. Worden K, Staszewski W (2000) Impact location and quantification on a composite panel using neural networks and a genetic algorithm. Strain 36:61–68
13. Wong P-K, Xu Q, Vong C-M, Wong H-C (2012) Rate-dependent hysteresis modeling and control of a piezostage using online support vector machine and relevance vector machine. IEEE Trans Ind Electron 59:1988–2001
14. Xu Q, Wong P-K (2011) Hysteresis modeling and compensation of a piezostage using least squares support vector machines. Mechatronics 21:1239–1251
15. Xie J (2010) Improved least square support vector machine for structural damage detection. In: 2010 2nd international conference on computer engineering and technology (ICCET), pp V6-237–V6-240
16. Fu H, Xu Q (2013) Locating impact on structural plate using principal component analysis and support vector machines. Math Probl Eng 2013. http://dx.doi.org/10.1155/2013/352149
17. Huang GB, Zhu Q-Y, Siew C-K (2004) Extreme learning machine: a new learning scheme of feedforward neural networks. In: Proceedings of 2004 IEEE international joint conference on neural networks, pp 985–990
18. Huang G-B, Zhu Q-Y, Siew C-K (2006) Extreme learning machine: theory and applications. Neurocomputing 70:489–501
19. Huang G-B, Chen L, Siew C-K (2006) Universal approximation using incremental constructive feedforward networks with random hidden nodes. IEEE Trans Neural Netw 17:879–892
20. Huang G-B, Wang DH, Lan Y (2011) Extreme learning machines: a survey. Int J Mach Learn Cybernet 2:107–122
21. Huang G-B, Zhou H, Ding X, Zhang R (2012) Extreme learning machine for regression and multiclass classification. IEEE Trans Syst Man Cybern B Cybern 42:513–529
22. Huang G-B, Li M-B, Chen L, Siew C-K (2008) Incremental extreme learning machine with fully complex hidden nodes. Neurocomputing 71:576–583
23. Li M-B, Huang G-B, Saratchandran P, Sundararajan N (2005) Fully complex extreme learning machine. Neurocomputing 68:306–314
24. Liang N-Y, Huang G-B, Saratchandran P, Sundararajan N (2006) A fast and accurate online sequential learning algorithm for feedforward networks. IEEE Trans Neural Netw 17:1411–1423
25. Huang G-B, Liang N-Y, Rong H-J, Saratchandran P, Sundararajan N (2005) On-line sequential extreme learning machine. Comput Intell 2005:232–237

26. Huang G-B, Chen L (2007) Convex incremental extreme learning machine. Neurocomputing 70:3056–3062
27. Huang G-B, Chen L (2008) Enhanced random search based incremental extreme learning machine. Neurocomputing 71:3460–3468
28. (2013) *kernel ELM toolbox, Matlab toolbox for kernel ELM*. Available: http://www.ntu.edu.sg/home/egbhuang/elm_kernel.html