

## Research Article

# Determination of Optimal Drop Height in Free-Fall Shock Test Using Regression Analysis and Back-Propagation Neural Network

Chao-Rong Chen,<sup>1</sup> Chia-Hung Wu,<sup>1,2</sup> and Hsin-Tsong Lee<sup>2</sup>

<sup>1</sup> Department of Electrical Engineering, National Taipei University of Technology, Taipei, Taiwan

<sup>2</sup> Environmental Engineering and Testing Section, System Development Center, National Chung-Shan Institute of Science and Technology, Tao-Yuan 325, Taiwan

Correspondence should be addressed to Chia-Hung Wu; [fydreamwu@gmail.com](mailto:fydreamwu@gmail.com)

Received 29 January 2014; Revised 25 June 2014; Accepted 29 June 2014; Published 17 August 2014

Academic Editor: Gyuhae Park

Copyright © 2014 Chao-Rong Chen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The primary purpose of this study is to provide methods that can be used to determine the most suitable drop height for shock testing military equipment, in an efficient and cost ineffective manner. Shock testing is widely employed to assess the performance of electronic systems, including military devices and civilian systems. Determining the height of the drop for a test item is an important step prior to performing the test. Dropping a test item from excessive height leads high G-peak values to damage the test equipment. On the other hand, dropping an item from a low height may not generate the required G-peak value and duration. Therefore, prior to performing shock tests, an optimal drop height must be established to ensure that the resulting G-peak value and duration time match the required test values. The traditional trial-and-error methods are time-consuming and cost-ineffective, often requiring many attempts. To improve the conventional approaches, this study proposes the application of regression analysis and back-propagation neural network for determining the most suitable drop height for free-fall shock tests. A new method is suggested for determining the drop test height. The results of the model are verified, using the results of a series of experiments that are conducted to verify the accuracy of the suggested approaches. The results of the study indicate that both approaches are equally effective in providing an effective guideline for performing drop tests from heights that would result in the peak Gs and duration needed for testing electronic devices.

## 1. Introduction

Shock testing has an increasingly important role in ensuring that military electronic devices meet the requirements for effective and reliable operation in harsh environments. Shock specifications of military electronic devices are generally expressed in terms of a simple acceleration pulse, such as a half-sine wave or a sawtooth wave in milliseconds, to simulate the shocks that the devices may experience in military environments. A shock test may include an assessment of the overall system integrity for safety purposes during handling, transportation, or use. The provision and application of shock simulation methods present many problems for laboratory-based teams [1]. Shock loading problems cover a very wide

range for physical parameters and are associated with many complex types of damage and malfunction. Shock tests will also be conducted to assess the life time from physical and functional performance of a device. Guo and Zhang [2] utilized the vibration transmissibility of a honeycomb structure which is applied regression analysis to design a shock absorber. Various methods have also designed shock waveforms and the conical lead pellet; however, these studies made the determination of the drop height in a shock machine disappear [3–5]. In the conventional trial-and-error method, a test item is placed on the shock machine table and then dropped from different heights to obtain various G-peak values and durations until the required shock specifications are met. The required shock specifications are obtained from

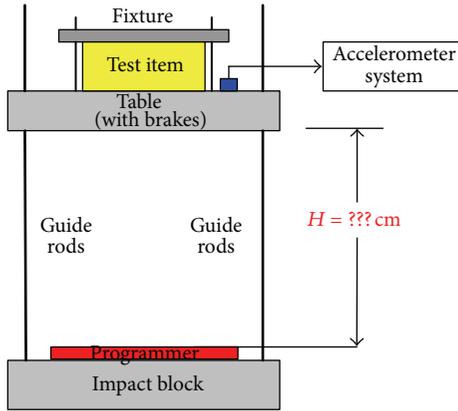


FIGURE 1: A free-fall shock test machine.

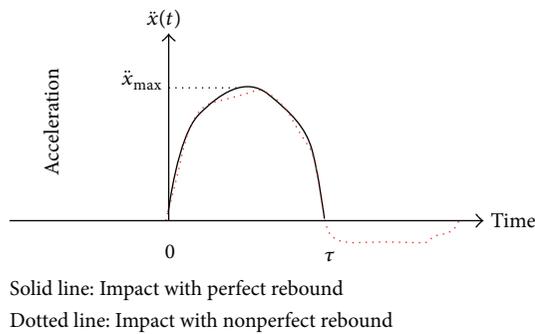


FIGURE 2: The waveform of a half-sine shock.

measuring the shock to a device. A device must be tested to verify its environmental adaptability to serve its owner. The test item was placed on the impact table of a free-fall shock test machine, as shown in Figure 1 and Figure 2 shows a typical shock waveform in the time domain. The dotted line represents the actual waveform resulting from the free-fall shock test item impacting the elastomers with an imperfect rebound [6]; the solid line indicates the specifications defined based on the peak Gs and duration of a half-sine waveform. The duration defines that a test item makes contact with the top programmer until the time it breaks contact. The shock event is often simulated using a dummy test item to adjust the height of the shock machine and determine the type of programmers needed via repetitive trials before a formal shock test is performed with the free-fall shock machine in the laboratory. This method is time-consuming and cost-ineffective as many attempts are needed and elastomers are consumed before the optimal drop height is determined. Thus far, a simple, convenient method is not suggested for estimating the optimal drop height. Regression analysis and a back-propagation neural network (BPNN) are reliable estimating approaches for determining the optimal drop height to assure the shock test quality in this crucial problem.

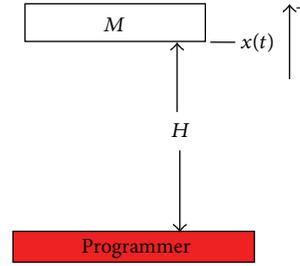


FIGURE 3: Model of shock motion.

## 2. Shock Motion

The purpose of the shock test is to obtain a waveform to match the solid line in Figure 2. The solid line in Figure 2 represents the acceleration history measured by an accelerometer fixed to the table. The relationship between the shock specifications (peak Gs and duration), drop height, and the programmer are elucidated in a shock test. Shock motion analysis is applied to use the dynamical model depicted in Figure 3.

In Figure 3,  $M$  is the total mass, including that of the impact table, test item, and fixture;  $x(t)$  is the impact table displacement; and  $H$  is the drop height from the top of the programmer to the bottom of the impact table.

The motion of the test item can be divided into two stages: before impact and after impact. According to Newton's second law the equation for motion in the preimpact stage can be expressed as

$$M\ddot{x}(t) = -Mg \quad \text{or} \quad \ddot{x}(t) = -g, \quad (1)$$

where  $g$  is gravity. The downward velocity is defined as negative and is derived as

$$V_f = -\sqrt{2gH}. \quad (2)$$

In the postimpact stage, as the table impacts the programmer, the programmer will rebound with a reactive force  $F$  to the impact table. Therefore, the equation for the motion of the test item during the impact can be written as

$$M\ddot{x}(t) = -Mg + F. \quad (3)$$

A detailed analysis of the reactive force is complex as the reactive force varies depending on the mechanical properties of the programmer material. In many cases, the reactive force is nonlinear result in computing the exact displacement,  $x(t)$ , which is difficult. A shock test is designed so as to verify the specifications, and an appropriate programmer is used to generate a waveform in which the peak Gs and duration mimic the solid line and match the shape in Figure 2. When the table impacts the programmer, its acceleration is given as (4) in which the initial condition of zero acceleration (gravity is neglected) is assumed. Consider the following:

$$\ddot{x}(t) = \ddot{x}_{\max} \sin\left(\frac{\pi}{\tau}t\right), \quad 0 \leq t \leq \tau. \quad (4)$$

The initial velocity of (4) is found based on (2); that is,

$$\dot{x}(0) = V_i = -\sqrt{2gH}, \quad (5)$$

where  $V_i$  is the impact velocity of the test item and programmer at the instant of impact. Equation (4) can be integrated and the result is substituted into (5) to yield

$$\dot{x}(t) = V_i + \frac{\tau \cdot \ddot{x}_{\max}}{\pi} \left[ 1 - \cos\left(\frac{\pi}{\tau}t\right) \right], \quad 0 \leq t \leq \tau. \quad (6)$$

Let  $V_\tau$  be the velocity at time  $t = \tau$  as the test item breaks contact with the programmer. Let  $t = \tau$  be substituted into (6), and  $V_\tau$  can be derived as in

$$V_\tau = V_i + \frac{2\tau \cdot \ddot{x}_{\max}}{\pi}. \quad (7)$$

The area under the curve of acceleration is equal to the change in velocity between  $V_\tau$  and  $V_i$ , as in

$$V_\tau - V_i = \Delta V = \frac{2\ddot{x}_{\max}}{\pi}\tau. \quad (8)$$

The function of the programmer is to act like a spring. Let the elastic coefficient of the programmer be given by  $k$ . The formula for the conservation of energy is given below:

$$\frac{1}{2}MV^2 = -\frac{1}{2}kx^2 + Mgx, \quad (9)$$

where  $V$  is the velocity of the test item and  $x$  is the displacement of the test item;  $x = 0$  implies that the test item is in contact with the programmer. The force is defined such that the upward direction is positive.

Equation (9) can be differentiated with respect to time to obtain

$$MV \frac{dV}{dt} = -kx \frac{dx}{dt} + Mg \frac{dx}{dt}. \quad (10)$$

Finally, (10) can be solved to yield

$$\ddot{x} + \frac{k}{M}x = g, \quad x(0) = 0, \quad x'(0) = V_i. \quad (11)$$

Let  $w = \sqrt{k/M}$ ; thus, the solution to (11) is

$$x = \frac{-g}{w^2} \cos(wt) + \frac{V_i}{w} \sin(wt) + \frac{g}{w^2}. \quad (12)$$

From (12) and (5), the acceleration can be rewritten as

$$\begin{aligned} \ddot{x} &= g \cos(wt) - V_i w \sin(wt) \\ &= g \cos(wt) + \sqrt{2gHw} \sin(wt) \\ &= A \sin(wt + \phi), \end{aligned} \quad (13)$$

where  $A$  and  $\phi$  are the magnitude and phase of the acceleration, respectively. Consider

$$\begin{aligned} A &= \sqrt{(g)^2 + (\sqrt{2gHw})^2} \\ \phi &= \tan^{-1} \left( \frac{g}{\sqrt{2gHw}} \right). \end{aligned} \quad (14)$$

When  $\sqrt{2gHw}$  is much larger than  $g$ ,  $A$  and  $\phi$  can be simplified as follows:

$$A \approx \sqrt{2gHw} = \sqrt{\frac{2gkH}{M}} \quad (15)$$

$$\phi \approx 0.$$

From (13)–(15), the acceleration can be represented as in

$$\ddot{x} = \sqrt{\frac{2gkH}{M}} \sin(wt) = \ddot{x}_{\max} \sin\left(\frac{\pi}{\tau}t\right), \quad (16)$$

where

$$\ddot{x}_{\max} = \sqrt{\frac{2gkH}{M}} \quad (17)$$

$$\tau = \frac{\pi}{w} = \sqrt{\frac{M}{k}}\pi.$$

The peak Gs is determined from  $k$  and  $H$ , and duration is determined from  $k$ .

### 3. Regression Analysis and Back-Propagation Neural Network (BPNN)

**3.1. Regression Analysis.** Regression analysis is used to fit the curve of the relationship between the input and output database. In this study, polynomial analysis and the ration linear polynomial regression analysis were performed on the data sets. The polynomial is denoted as follows:

$$H = f(y; \theta_0, \theta_1, \theta_2, \dots, \theta_{n-1}, \theta_n), \quad (18)$$

where  $y$  is the input data;  $H$  is the output data; and  $\theta_0, \theta_1, \theta_2, \dots, \theta_{n-1}, \theta_n$  are the parameters to be determined. All data can be reformulated into an equation as follows:

$$\begin{aligned} f(y_i; \theta_0, \theta_1, \theta_2, \dots, \theta_{n-1}, \theta_n) \\ = \theta_0 + \theta_1 y_i + \theta_2 y_i^2 + \dots + \theta_{n-1} y_i^{n-1} + \theta_n y_i^n, \end{aligned} \quad (19)$$

where  $i = 1, \dots, k$  for different cases.

The square error can be written as  $[H_i - f(y_i)]^2$ . The total square error  $Q$  is

$$\begin{aligned} Q &= \sum_{i=1}^k [H_i - f(y_i)]^2 \\ &= \sum_{i=1}^k \left[ H_i - (\theta_0 + \theta_1 y_i + \theta_2 y_i^2 + \dots + \theta_{n-1} y_i^{n-1} + \theta_n y_i^n) \right]^2, \end{aligned} \quad (20)$$

where  $i = 1, \dots, k$  for different cases and  $n$  is the polynomial order.

To minimize the total square error  $Q$ , one must set the partial differential of  $Q$  to  $\theta_0, \theta_1, \theta_2, \dots, \theta_n$  at zero:

$$\frac{\partial Q}{\partial \theta_0} = \frac{\partial Q}{\partial \theta_1} = \frac{\partial Q}{\partial \theta_2} = \dots = \frac{\partial Q}{\partial \theta_n} = 0. \quad (21)$$

The optimal solutions of  $\theta_0, \theta_1, \theta_2, \dots, \theta_{n-1}, \theta_n$  can be derived when the data sets are individually input into (20) to obtain the following matrix equation for the different cases:

$$\begin{bmatrix} 1 & y_1 & y_1^2 & \dots & y_1^{n-1} & y_1^n \\ 1 & y_2 & y_2^2 & \dots & y_2^{n-1} & y_2^n \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & y_k & y_k^2 & \dots & y_k^{n-1} & y_k^n \end{bmatrix} \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_n \end{bmatrix} = \begin{bmatrix} H_1 \\ H_2 \\ \vdots \\ H_k \end{bmatrix} \quad (22)$$

or

$$\mathbf{P}\boldsymbol{\theta} = \mathbf{H}, \quad (23)$$

where  $n$  is the polynomial order.

It is clear that (23) has  $n$  unknown variables over all cases. Substituting (21) into (20) yields the square error as follows:

$$Q = \|\mathbf{H} - \mathbf{P}\boldsymbol{\theta}\|^2 = (\mathbf{H} - \mathbf{P}\boldsymbol{\theta})^T (\mathbf{H} - \mathbf{P}\boldsymbol{\theta}). \quad (24)$$

Another type is the rational linear polynomial. The rational linear polynomial is often used to obtain minimum errors in many cases. This mathematical expression can be written as

$$D_x = \frac{a_x \cdot G_x + b_x}{G_x + T_x}, \quad (25)$$

where  $x$  is the case number;  $a_x, b_x$ , and  $T_x$  are the parameters;  $D_x$  is the output value; and  $G_x$  is the input value of the case  $x$ .

**3.2. BPNN.** The back-propagation neural network (BPNN) was introduced by Rumelhart and McClelland in 1985. It is a multilayer and forward-feedback perceptron with learning capability [7, 8]. The use of the BPNN is a computational technique for realizing that a performance similar to that of a human solves problems. A BPNN can be considered as a web of simple processing units, based on neurons that are linked to each other through connections that operate similar to synapses. These connections contain the “knowledge” of the network, and the patterns of connectivity indicate the objects in the network. The knowledge of the network is acquired through a learning process in which the connectivity between processing units is varied (through weight changes). BPNN is an efficient alternative for obtaining solutions where it is possible to obtain data regarding problem behavior. BPNN has several attractive characteristics. The abilities which adapt to system data and perform new tasks are important advantages of this technique. BPNNs are parallel structures that usually require small amounts of memory and short processing times. BPNNs can store knowledge in a distributed type and have a high fault tolerance. In this paper, a multilayer perceptron (MLP) was used; it was trained via a supervised learning algorithm. The learning algorithm updates the weight and bias values according to Levenberg-Marquardt optimization. The Levenberg-Marquardt algorithm provides a numerical solution to the problem of minimizing a function. It is a popular alternative to the Gauss-Newton method to find a minimum value. A BPNN can handle nonlinear relations between the variables. The weight adjustments are realized to

minimize the error function for a certain pattern. Equation (26) defines the error function as

$$E_p = \frac{1}{2} \sum_j (d_{pj} - y_{pj})^2, \quad (26)$$

where  $d_{pj}$  is the desired output and  $y_{pj}$  is the actual output.

The partial derivative  $(\partial E_p)/(\partial w_{ji})$  determines the weight adjustment characterizing the gradient descent of the algorithm. This partial derivation measures the weight  $w_{ji}$  contribution to the BPNN error function for pattern  $p$ . If this derivation is positive, the error amount is considered to be increasing and the weight  $w_{ji}$  should be reduced in order to decrease the difference between the actual output and the desired output. In the case of the partial derivation being negative, the weight  $w_{ji}$  contributes to the generation of an output smaller than the desired output (and, therefore, to reduce this difference, the weight should be increased). Equation (27) defines the way in which the connection weights in the network are adjusted:

$$\Delta w_{ji} = w_{ji}(t+1) - w_{ji}(t) = \eta \left( -\frac{\partial E_p}{\partial w_{ji}}(t) \right), \quad (27)$$

where  $w_{ji}$  is the weight of the connection between neurons  $i$  and  $j$ , and  $\eta$  is the learning rate.

The derivation was shown in (27) to provide a generic rule used to adjust the weight of the connections. Equation (28) illustrates this generic rule of weight adjustment below:

$$w_{ij}(t+1) = w_{ij}(t) + \eta \delta_j y_j, \quad (28)$$

where  $\delta_j = \sum_k \delta_k w_{jk}$  if neuron  $j$  is a hidden layer unit.

The choice of an appropriate learning parameter  $\eta$  will significantly influence the convergence rate of the algorithm. If this parameter is too small, it may take several steps to reach an acceptable convergence. On the other hand, a high learning rate can lead to oscillations that could impede convergence. A possible solution is to use a momentum term  $\alpha$  that would quantify the importance of the weight variation at a step. Equation (29) illustrates the weight adjustment rule with a momentum term:

$$w_{ij}(t+1) = w_{ij}(t) + \eta \delta_j y_j + \alpha [w_{ij}(t) - w_{ij}(t-1)]. \quad (29)$$

The momentum term can make the learning algorithm more stable and accelerate the convergence in flat regions of the error functions. However, determining the optimal value of the momentum parameter  $\alpha$  introduces the same difficulties as encountered in the case of the learning rate parameter.

This study applies BPNNs to determine the drop height and duration for the shock test with programmers.

The structure of the BPNN (Figure 4) is as follows.

- (1) The BPNN has an input layer expressing the input parameters in the network. The number of neurons depends on the problem's complexity. The input layer is the first (bottom) layer in the structure.

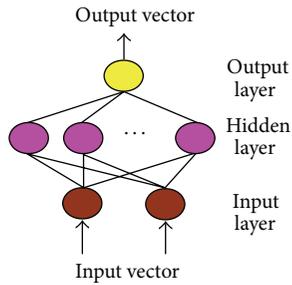


FIGURE 4: Structure of BPNN.

- (2) Above the bottom layer is a middle hidden layer expressing the interaction between input parameters and processed neurons. The number of neurons cannot be precisely determined. The number of neurons is typically determined based on when the optimal result is obtained.
- (3) The third (top) layer of the BPNN is an output layer which denotes the network output. The number of neurons is also determined based on the problem's complexity.
- (4) The BPNN includes a transfer function. The sigmoid function is chosen as the nonlinear transfer function; it is expressed as follows:

$$f(x) = \frac{1}{1 + e^{-x}}. \quad (30)$$

The following procedure is used to analyze the BPNN:

- (1) determine the number of neurons at each layer;
- (2) set the initial weights and bias values in the network randomly;
- (3) insert input and output vectors into the network for training the weights;
- (4) estimate the output values of the hidden and output layers;
- (5) calculate the difference in output values between the hidden layer and output layer;
- (6) establish the adjustment coefficients for weights and bias values;
- (7) update bias and weight values;
- (8) repeat steps (3)–(7) until the network converges.

#### 4. Proposed Scheme

The proposed prediction scheme is composed of data collected under different conditions. The peak Gs and duration of the shock test are determined for different programmers. From previous (and new) experiments, the data sets are arranged so as to fit the optimal curve by regression analysis; the BPNN is also trained to develop a knowledge database until the learning structure is robust. The data sets for four cases of different programmers are gathered to train and recall the target using either a BPNN or regression analysis.



FIGURE 5: The setup of the free-fall shock machine.

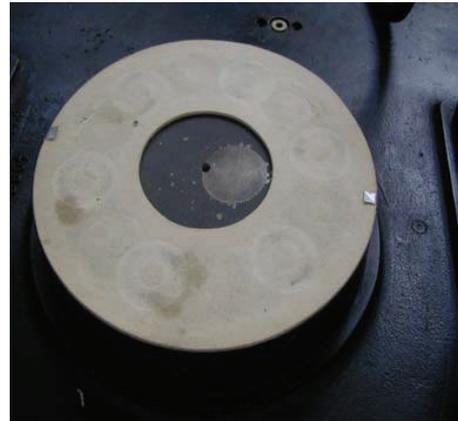


FIGURE 6: The elastomer of felt on the machine.



FIGURE 7: The plastic elastomer on the machine.

Finally, the degree of accuracy of targets is compared for either BPNN or regression analysis in various situations.

*4.1. Data Collected and Error Evaluating Processes.* Before collecting data sets, all programmers and free-fall shock machines are calibrated for testing. Figure 5 shows the complete experimental setup of the free-fall shock machine. Figures 6 and 7 show photographs of different programmers.

TABLE 1: Training data sets by free-fall shock machine.

Set	$H$ (cm)						
	10	20	40	60	80	100	120
Case 1 plastic programmer							
Gpk (g)	54.08	95.51	165.71	233.61	291.395	334.19	383.3
$k$	0.1	0.1	0.1	0.1	0.1	0.1	0.1
Case 2 three pieces of elastomer							
Gpk (g)	—	82.41	190.03	324.22	463.55	613.73	794.82
$k$	—	0.3	0.3	0.3	0.3	0.3	0.3
Case 3 two pieces of elastomer							
Gpk (g)	59.21	134.2	306.63	449.72	632.13	811.05	—
$k$	0.45	0.45	0.45	0.45	0.45	0.45	—
Case 4 one piece of elastomer							
Gpk (g)	82.82	199.28	443.36	647.99	862.61	—	—
$k$	0.9	0.9	0.9	0.9	0.9	—	—

TABLE 2: Test data sets by free-fall shock machine.

Set	$H$ (cm)			
	90	60	60	40
Gpk (g)	314.04	326.54	436.34	445.68
$k$	0.1	0.3	0.45	0.9
$D$ (ms)	2.98	2.92	1.66	1.30

In the experiment, the same test item, type, and number of programmers were used to obtain data sets at seven heights (10, 20, 40, 60, 80, 100, and 120 cm); each test was run twice at each height to obtain peak Gs and durations that were close to the specifications. The experiment was run 28 times to ensure that the 28 data sets collected were within tolerances. Bad data sets were eliminated as necessary during data processing. Tables 1 and 2 present all 28 data sets which indicate the peak Gs and durations  $\tau$ . Figure 8 shows the four representative shock-test results. The data sets were used to verify the effectiveness of the proposed method for determining the optimal drop height and duration.

Tables 3 and 4 also showed the resultant mean square errors (MSEs). The MSE is defined in (31) and the maximum error (ME) is also defined in (32):

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (H_{ti} - H_{ei})^2 \quad (31)$$

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (\tau_{ti} - \tau_{ei})^2$$

$$\begin{aligned} \text{Max } E_H &= \text{MAX} |H_{tj} - H_{ej}| \\ \text{Max } \tau_H &= \text{MAX} |\tau_{tj} - \tau_{ej}|, \end{aligned} \quad (32)$$

where  $H_t$  and  $\tau_t$  are the target height and duration, respectively;  $H_e$  and  $\tau_e$  are, respectively, the height and duration estimated by either regression analysis or BPNN;  $j$  is the test number; and  $n$  is the total number of data sets in each case.

**4.2. Regression Analysis Application.** The 24 data sets were selected for drop heights of 10, 20, 40, 60, 80, 100, and 120 cm (Table 1), and the remaining four data sets were used for verification of the results via regression analysis (Table 2). The relationship between  $D$  and  $G$  in cases 1 and 3 used a linear polynomial for estimating duration as (23); however, the relationship between  $D$  and  $G$  in cases 2 and 4 used rational linear polynomial for estimating duration as (25). The optimal parameters for (25) are  $a_2 = 0.0895$ ,  $b_2 = 1095$ , and  $T_2 = 102.7$  when  $x = 2$  and  $a_4 = 0.925$ ,  $b_4 = 90.82$ , and  $T_4 = -43.33$  when  $x = 4$ . Cases 1 and 3 and cases 2 and 4 are coupled when drawing the fitting curves. Figures 9 and 10 depict the relation between drop height  $H$  and  $G$ . Figures 11 and 12 depict the relation between  $D$  and  $G$ . The solid line and black dotted line are the fitting curves for cases 1 and 3 in Figures 9 and 10, respectively, as determined by regression analysis. In Figures 11 and 12, the solid line and black dotted line are the fitting curves for cases 2 and 4, respectively, as also determined by regression analysis. The circles in these figures are the fitting data sets; the crosses represent the test data sets. Tables 3 and 4 show the MSEs of height and duration values. According to the estimated results, the average MSE heights of the regressive patterns and test patterns are 0.02218 cm and 0.9368 cm, respectively. Meanwhile, the average MSE duration for the regressive patterns and test patterns is 0.00656 ms and 0.024 ms, respectively. The maximum average error of height and average error of duration in the regressive and test patterns is individually less than 1 cm and 1 ms, respectively. Regression analysis can be used to determine the equations for drop height and duration; however, only three variables can be used to fit the curves.

**4.3. BPNN Application.** In the same arrangement used for the training network in Table 1, the remaining four data sets are used for verification of the results obtained via the BPNN test (Table 2). Each selected data set includes two elements, the peak Gs and the coefficient of elasticity,  $k$ . The coefficient of elasticity,  $k$ , and the weight of a test item for the four cases are set as constants. The data sets are normalized with respect to

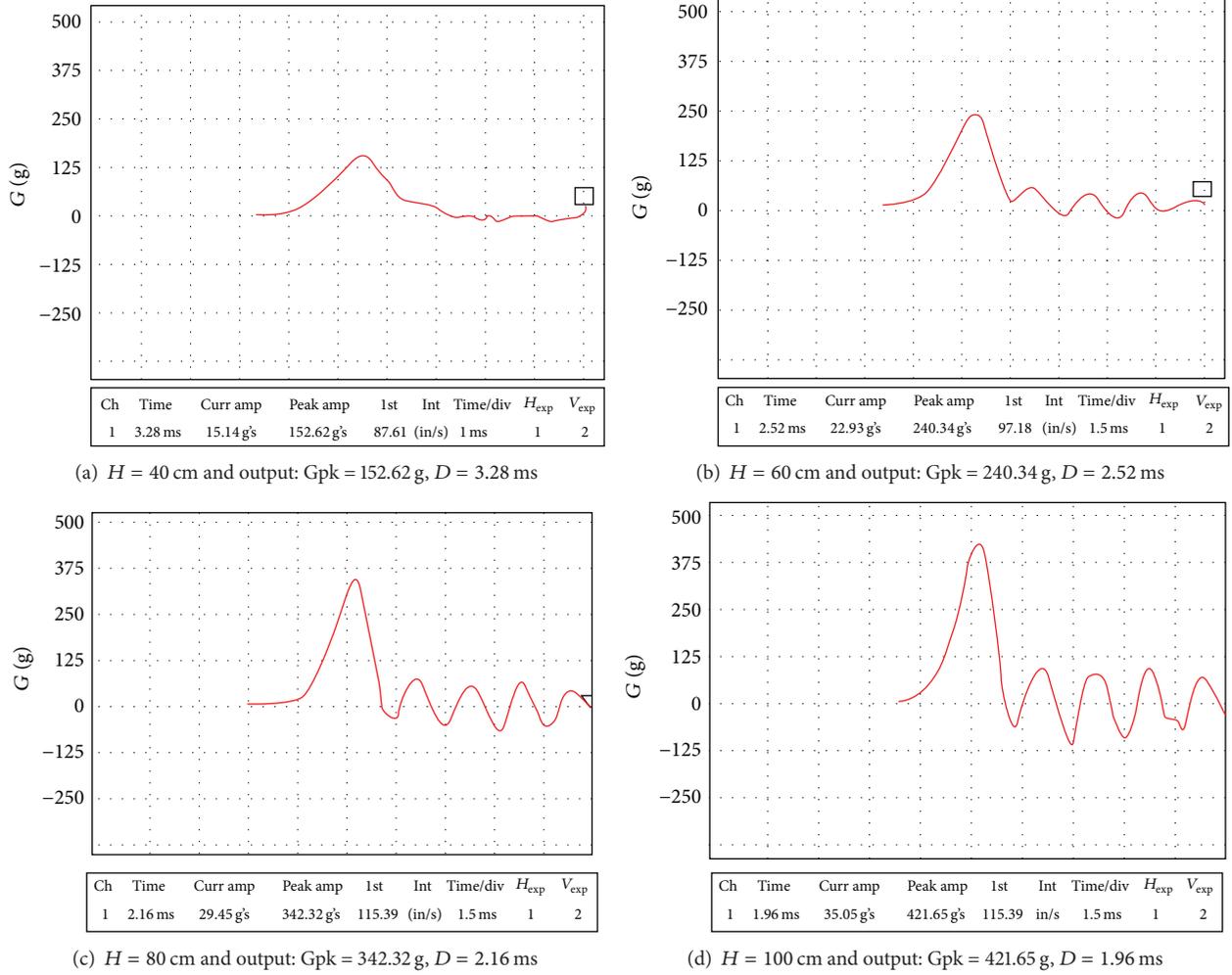


FIGURE 8: Representative shock test results.

TABLE 3: MSE of heights by BPNN and regression analysis.

Item	Case				Average MSE (cm)
	Case 1 (cm)	Case 2 (cm)	Case 3 (cm)	Case 4 (cm)	
Training patterns					
BPNN	$2.026 \times 10^{-18}$	$5.845 \times 10^{-21}$	$8.067 \times 10^{-21}$	$1.338 \times 10^{-21}$	$5.103 \times 10^{-19}$
Regression	0.0887	$4.14 \times 10^{-27}$	$8.84 \times 10^{-26}$	$1.73 \times 10^{-27}$	0.02218
Test patterns					
BPNN	0.0462	0.1406	3.227	0.013	0.8567
Regression	0.178	0.112	3.411	0.046	0.9368

TABLE 4: MSE of time duration by BPNN and regression analysis.

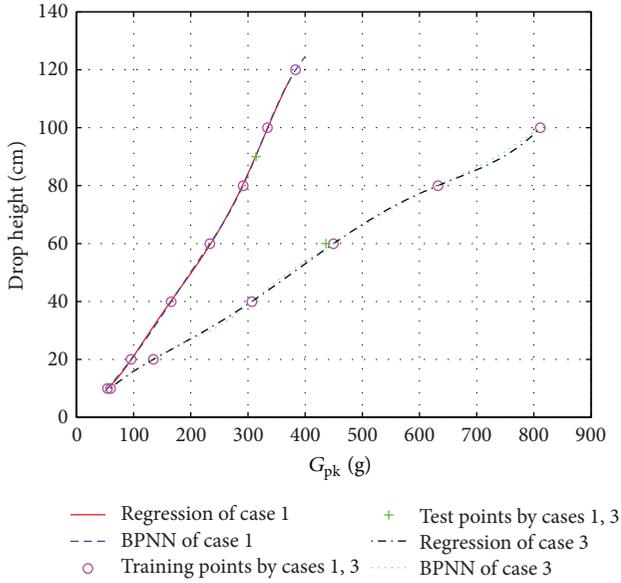
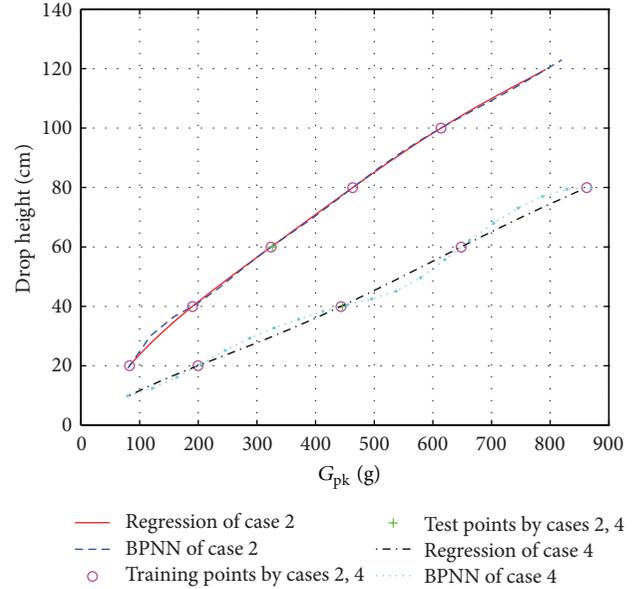
Item	Case				Average MSE (ms)
	Case 1 (ms)	Case 2 (ms)	Case 3 (ms)	Case 4 (ms)	
Training patterns					
BPNN	$5.210 \times 10^{-25}$	$3.324 \times 10^{-27}$	$5.941 \times 10^{-24}$	$7.60 \times 10^{-28}$	$1.617 \times 10^{-24}$
Regression	$2.051 \times 10^{-5}$	0.0248	$2.77 \times 10^{-28}$	0.0014	0.00656
Test patterns					
BPNN	$7.67 \times 10^{-4}$	$2.551 \times 10^{-6}$	$9.20 \times 10^{-4}$	$5.87 \times 10^{-5}$	$4.36 \times 10^{-4}$
Regression	$5.39 \times 10^{-4}$	0.0905	0.002	0.003	0.024

TABLE 5: Maximum errors (ME) of height by BPNN and regression analysis.

Item	Case				Average ME (cm)
	Case 1 (cm)	Case 2 (cm)	Case 3 (cm)	Case 4 (cm)	
Training patterns					
BPNN	$2.404 \times 10^{-9}$	$1.532 \times 10^{-10}$	$1.608 \times 10^{-10}$	$6.280 \times 10^{-11}$	$6.952 \times 10^{-10}$
Regression	0.498	$1.421 \times 10^{-13}$	$5.4 \times 10^{-13}$	$8.530 \times 10^{-14}$	0.1245
Test patterns					
BPNN	0.215	0.375	1.796	0.1147	0.6252
Regression	0.178	0.112	3.411	0.046	0.937

TABLE 6: Maximum errors (ME) of time duration by BPNN and regression analysis.

Item	Case				Average ME (ms)
	Case 1 (ms)	Case 2 (ms)	Case 3 (ms)	Case 4 (ms)	
Training patterns					
BPNN	$1.172 \times 10^{-12}$	$9.415 \times 10^{-14}$	$1.377 \times 10^{-13}$	$5.596 \times 10^{-14}$	$3.65 \times 10^{-13}$
Regression	0.0076	0.3071	$3.575 \times 10^{-14}$	0.0583	0.0932
Test patterns					
BPNN	0.0277	0.0016	0.0303	0.0077	0.0168
Regression	0.0232	0.301	0.0428	0.0502	0.1043

FIGURE 9: Fitting curves of  $H$  versus  $G$  in cases 1 and 3.FIGURE 10: Fitting curves of  $H$  versus  $G$  in cases 2 and 4.

peak  $G_s$ , height, and duration to 0.2–0.8. This study uses two neurons in the input layer, 12 neurons in the hidden layer, and one neuron in the output layer. After the BPNN was trained 163 times, the network converged (Figure 13). The sudden drop means a local minimum error. Using early stopping can overcome this phenomenon and the results were not affected through real experiments verified. Tables 3 and 4 show the MSE values of height and duration obtained by regression analysis and by BPNN. Therefore, the fitting curve by BPNN comprises both drop height  $H$  versus  $G$  and duration  $D$  versus  $G$  (Figures 9–12). Equations (31) and (32) are then applied to

calculate the height and duration of the training pattern for an average MSE of  $5.103 \times 10^{-29}$  cm and  $1.617 \times 10^{-24}$  ms, respectively; both values show a satisfactory training result. The averages of maximum errors of height and duration via training by BPNN are  $6.952 \times 10^{-10}$  cm and  $3.65 \times 10^{-13}$  ms, respectively (Tables 5 and 6). The four data sets were further tested to estimate the height and duration by BPNN (Figures 9–12), and Tables 3 and 4 show the values of training and test results. The average MSEs of the height and duration for the test pattern are 0.8567 cm and  $4.36 \times 10^{-4}$  ms, respectively. The averages of the maximum errors of height and duration are

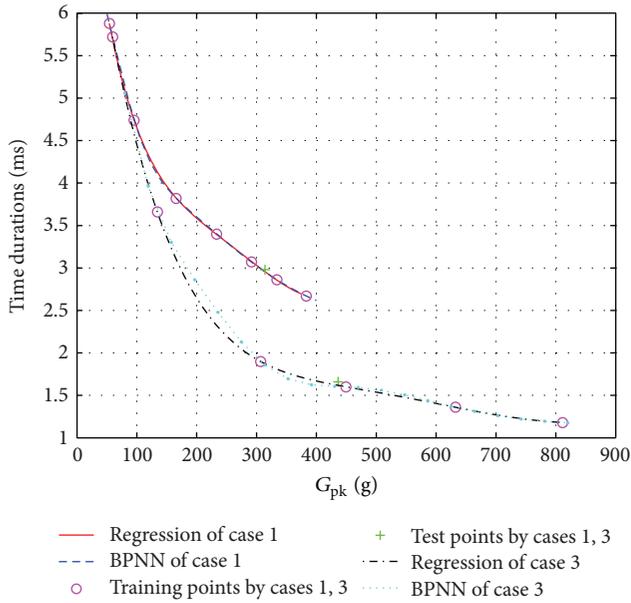


FIGURE 11: Fitting curves of  $D$  versus  $G$  in cases 1 and 3.

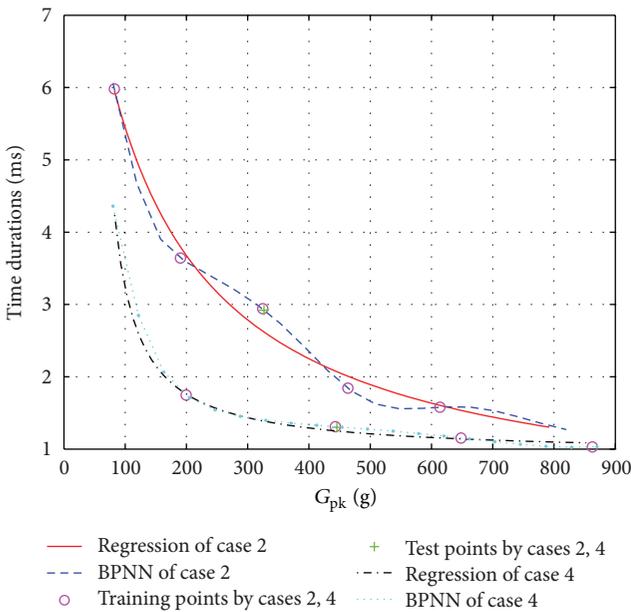


FIGURE 12: Fitting curve of  $D$  versus  $G$  in cases 2 and 4.

0.6252 cm and 0.0168 ms, respectively. Thus, the BPNN can be applied successfully to determine the optimal drop height and duration for the free-fall shock test. The BPNN values have small errors as they obtained a good fitting curve in this study. Tables 3–6 summarize the conditions in the different cases and computational results.

**4.4. Comparison of the Two Approaches.** The fitting curves obtained by regression analysis do not completely match the fitting curves obtained by the BPNN. The curves of the BPNN are spiral on regressive curves (Figures 10–12);

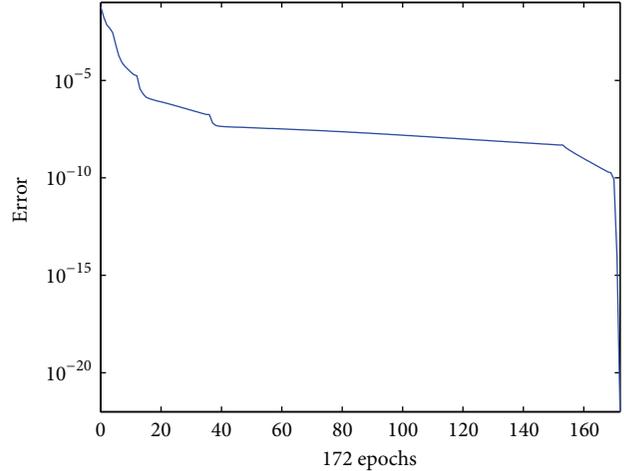


FIGURE 13: Training of epochs by ANN.

TABLE 7: Performance comparison: regression analysis versus BPNN.

Condition	Manner	
	Regression analysis	BPNN
Ability to handle more than 2 variables	NO	YES
Degree of accuracy	GOOD	EXCELLENT
Ability to develop approximate equations	YES	NO
Need to collect cases	YES	YES
Time efficiency	FAST	FAST
Need for recalling	YES	YES
Ability to solve complicated nonlinear problems	NO	YES

these curves indicate that some difference exists between the two methods. The computational results obtained by regression analysis and BPNN are very close to the actual drop heights and durations. A further comparison of the two approaches (Tables 3–6) indicates that the BPNN performs better than regression analysis in terms of average MSE and average maximum error. Moreover, the BPNN outperforms regression analysis based on its ability to accommodate more than three variables. Additionally, the order of precision obtained when using regression analysis is not particularly high. It is noteworthy that the test value for duration in case 2 obtained by the BPNN and regression analysis is 0.0016 ms and 0.301 ms, respectively (Table 6). Case 2 demonstrates that the test point on the curve obtained by BPNN estimated with a higher degree of accuracy than obtained by regression analysis (Figure 12) (the application of regression analysis is limited to only three variables). A summary of this study’s conclusions and a performance comparison are given in Table 7.

## 5. Conclusion

Regression analysis and a BPNN are applied to analyze the nonlinear relationships and estimate the optimal height and duration for free-fall shock tests in this study. The conventional “trial-and-error” approach for determining drop height is time-consuming as it requires repeated trials. The goal of this study is to improve the conventional method (in terms of time and cost) of conducting free-fall shock tests. The BPNN accurately estimates the values of height and duration during shock testing without limiting the number of variables. The results of this study indicate that both approaches are equally effective in providing an effective guideline for performing drop tests.

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

## Acknowledgments

The authors would like to thank the Environmental Engineering and Testing Section, System Development Center of National Chung-Shan Institute of Science & Technology, for the data sets that were used in this research. This work was supported in part by the National Science Council in Taiwan, under the Project title: Caltech-Taiwan collaboration on energy research-uncertainty mitigation for renewable energy integration, Project no. NSC 101-3113-P-008-001.

## References

- [1] M. C. Jones, “Shock simulation and testing in weapons development,” in *Proceedings of the 32nd Annual Technical Meeting of the Institute of Environmental Sciences*, pp. 17–21, 1986.
- [2] Y. Guo and J. Zhang, “Shock absorbing characteristics and vibration transmissibility of honeycomb paperboard,” *Shock and Vibration*, vol. 11, no. 5-6, pp. 521–531, 2004.
- [3] P. F. Cunniff and G. J. O’Hara, “A procedure for generating shock design values,” *Journal of Sound and Vibration*, vol. 134, no. 1, pp. 155–164, 1989.
- [4] C. T. Morrow and H. I. Sargeant, “Sawtooth shock as a component test,” *The Journal of the Acoustical Society of America*, vol. 28, no. 5, pp. 959–965, 1956.
- [5] R. L. Felker, *Empirical Rules for Shock Spectrum, Pulse, and Lead Pellet Interrelationship and Implementation*, vol. 30, Autonetics Division, North American Aviation, 1967.
- [6] L. Lalanne, *Mechanical Vibration & Shock Volume II*, pp. 30–50, New York Publishing, 1st edition, 2003.
- [7] R. Hecht-Nielsen, “Theory of the backpropagation neural network,” in *Proceedings of the International Joint Conference on Neural Networks (IJCNN '89)*, vol. 1, pp. 593–605, Washington, DC, USA, June 1989.
- [8] P. Venkataraman, *Applied Optimization with MATLAB Programming*, John Wiley & Sons, New York, NY, USA, 1st edition, 2002.



**Hindawi**

Submit your manuscripts at  
<http://www.hindawi.com>

