ARTICLE IN PRESS

Available online at www.sciencedirect.com
SCIENCE DIRECT®

ELSEVIER

International Journal of Plasticity xxx (2004) xxx–xxx

INTERNATIONAL JOURNAL OF Plasticity

www.elsevier.com/locate/ijplas

Explicit and implicit viscoplastic models for polymeric composite

M.S. Al-Haik [a], H. Garmestani [b,*], A. Savran [c]

[a] *Department of Mechanical Engineering, FAMU-FSU College of Engineering Florida A&M University-Florida State University, Tallahassee, FL 32310, USA*
[b] *Department of Materials Science and Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA*
[c] *Department of Electrical and Electronics Engineering, Faculty of Engineering, Ege University, 35100 Bornova, Izmir, Turkey*

Received in final revised form18 November 2003

Abstract

The viscoplastic behavior of a carbon-fiber/polymer matrix composite was investigated through two different modeling efforts. The first model is phenomenological in nature and utilizes the tensile and stress relaxation experiments to predict the creep strain. The phenomenological model was constructed based on the overstress viscoplastic model. In the second model, the composite viscoplastic behavior is captured via neural networks formulation. The neural networks model was constructed directly from the experimental results obtained via creep tests performed at various stress–temperature conditions. The neural network was trained to predict the creep strain based on the stress–temperature–time values. The performance of the neural model is evaluated through the mean squared error between the neural network prediction and the experimental creep strain results. To minimize this error, several optimization techniques were examined. The minimization of the error when carried out by the Truncated Newton method outperforms the standard back-propagation and the conjugate gradient method in terms of convergence rate and accuracy. Using neural network with truncated Newton training algorithm, the prediction of the creep strain was very satisfactory compared to the phenomenological model.

Keywords: Material testing; Polymeric composite; Constitutive behavior; Viscoplastic material; Neural networks; Optimization

* Corresponding author. Tel.: +1-404-385-4495; fax: +1-404-894-9140.
E-mail address: hamid.garmestani@mse.gatech.edu (H. Garmestani).

0749-6419/$ - see front matter © 2003 Published by Elsevier Ltd.
doi:10.1016/j.ijplas.2003.11.017

1. Introduction

Because of their outstanding mechanical properties, polymer matrix composites (PMCs) are used extensively as commodity and specialty engineering materials. As PMCs continue to find new applications in aerospace, automotive industry, and civil engineering structures, there is an urgent need to assess the durability of these engineering materials. When polymeric composites are exposed to elevated temperature and excessive stress environments, the mechanical behavior of polymer matrix composites is significantly affected by these ambient conditions and the composites exhibit inelastic behavior. The reason of this phenomenon is micro-damage accumulated in laminates and inelastic stress–strain response of the matrix (Megnis and Varna, 2003). The inelasticity of polymeric composite structures is distinguished under conditions of creep (Kan et al., 1987).

The creep response of polymeric composites is related to the inherent time-dependent behavior of the polymeric phase and the temporal behavior of carbon fibers due to fiber–matrix debonding, interlayer delamination. Since all the later damage mechanisms are also time-dependent and occur co-jointly with polymeric creep, it is extremely difficult to separate out the individual contributions of the composite constituents. While there is enormous literature related to the modeling of the creep behavior of polymers (Krempl and Khan, 2003, Khan and Zhang, 2001, Khan and Pamies, 2002), there are fewer reviews on the modeling of creep in polymeric composites. Among the earlier investigations are the articles by Haplin (1968), Schapery (1968), Suvorova (1985), Chan (1988), Dillard (1991), and recently by Brinson and Gates, 1995), Govindarajan et al. (1997), and Brinson and Lin (1998). Most of this literature utilizes linear viscoelastic models to describe the creep behavior of the composite.

A thermodynamically consistent theory of nonlinear viscoelastic and viscoplastic materials was developed by Schapery (1997). This model considers the nonlinear viscoplastic response of materials as a particular case of nonlinear viscoelasticity corresponding to infinite retardation times. Experimental methodology for complete material characterization in the framework of this model was presented by Megnis and Varna (2003).

Another approach to the modeling of creep is formulated based on plasticity theory, which was applied initially to metal matrix composites by Perzyna (1966) and later by Liu and Krempl (1979). This approach can be expanded to include the time-dependent behavior for a laminated structure by including phenomenological macroscopic analysis for each particular laminate lay-up based on laminate theory type of analysis that describes the inelastic response of laminates based on inelastic stress–strain relationships of a layer.

For uniaxially reinforced polymeric composites, Sun and Chen (1991), Yoon and Sun (1991), Sun and Chang (1992) developed a simplified single parameter plasticity model for creep. This model was subsequently extended to laminated plates by Sun and Zhu (2000) and included thermal effects by Gates (1992, 1993a,b). Rate dependence was included by means of a viscoplastic model by Gates and Sun (1991, 1995) and later by Thiruppukuzhi and Sun (1998, 2001).

Viscoplastic constitutive equations are written explicitly and they involve many parameters, which significantly influence the behavior of the constitutive equations. Several viscoplastic models were presented by Lubarda et al. (2003), Ho and Krempl (2002), Li and Weng (1998), Rubin (1996), Freed and Walker (1993), Moreno and Jordan (1986), Fotiu and Nemat-Nasser (1996). Throughout these models, appropriate parameters must be determined accordingly, so the accurate behaviors of the material can be expressed. The parameter identification problem for viscoplastic model was investigated by Mahnken and Stein (1996). Once the model is set, the behavior of the material can only be expressed approximately by adjusting the parameters in the model. Under conditions of high stress, strain rate and high temperature, to secure more accurate results the model has to become more complicated in the mathematical formulation, but the problem of parameter identification of the material parameters will introduce more numerical errors and instabilities to the model as shown by Gates (1993a,b).

From our earlier investigation, Al-Haik et al. (2001), Al-Haik and Garmestani (2001), Al-Haik (2002), Garmestani et al. (2001), some of the problems involved with the phenomenological creep models, include:

(1) The models are simply based on the phenomenological investigation of material properties while the actual behavior of the material is very complex. Therefore, inevitably the model contains errors.
(2) All these models are limited in their mathematical form and because they are written explicitly, they require an appropriate set of data for parameter identification.
(3) Compared to metals, composite materials require extra parameters for their characterization like fiber orientation, volume fraction, fiber–matrix interface, etc. These parameters will raise the degree of complexity of the earlier constitutive models.

Bearing in mind the shortcomings of the phenomenological model, an alternative approach is to use a computation and knowledge representation paradigm i.e. *neural networks*. Recently, computational mechanics, which is quantitatively reliable by itself, has widened its feasibility to practical engineering problems by merging the *artificial neural networks* (ANN) flexibility.

An artificial neural network mimics the structure and functionality of a biological nerve system, and is referred to as a parallel distributed processing. An artificial neural network is capable of making decisions based on incomplete, noisy, and disordered information. It also can *generalize* rules from those cases on which it was trained and apply these rules to new stimuli. Neural networks architecture is a promising implicit-modeling scheme aiming to replace the traditional explicit constitutive equations used to describe material behavior.

Recently, several investigations suggested that neural network can directly map the behavior of a thermal viscoplastic material, and therefore it is not necessary to postulate a mathematical model and identify its parameters.

In the investigation carried out by Qingbin et al. (1996) a four-layer backpropagation neural network was built to acquire the constitutive relationship of 12Cr2Ni4A. Temperature, effective strain, and strain rate were used as the input

vectors to the neural network, the output of the neural network being the flow stress. The trained network was capable of not only reproducing the flow stress in the sample data (where the training occurred), but also predicted the flow stress when other datasets were examined.

Furukawa et al. (1995, 1996, 1998) proposed an implicit viscoplastic constitutive model using neural networks. In that model, inelastic material behavior was generalized in a state-space representation, and the state-space form was constructed by a neural network using input–output datasets. The input data consisted of inelastic strain, internal variables, and stress at every iterative time step. The output data consisted of inelastic strain rate. A technique to extract the input–output data from experimental data was also described. The proposed model was generated from pseudo-experimental data created by one of the widely used constitutive models (unified theory describing cyclic loading and viscous behavior) and was found to replace the model well. Then, having been tested with the actual experimental data, the proposed model resulted in a negligible amount of model errors indicating its superiority to all the existing explicit models in accuracy. Based on the variational principle of elastoplasticity, Daoheng et al. (2000) proposed a neurocomputational model to solve elastoplastic problems. In that model the finite element computation of elastoplastic mechanics was transformed to a quadratic programming problem with inequality constraints, that problem was solved using neural networks. It was shown that the neurocomputed results are well in agreement with those of theoretical analysis and the solution of the elastoplasticity can be obtained using real time analysis. In a different approach, Huber and Tsakmakis (2001) used neural networks to solve the inverse problem of parameter identification. Neural networks represented an explicit relation between the measured strain, stress, time and the material parameters to be identified. The constitutive model described finite deformation viscoplasticity and exhibited static recovery in both the isotropic and the kinematics hardening laws. To train the neural networks, a loading history was utilized, which consists of a homogeneous uniaxial deformation including cyclic loading and relaxation phases. It was shown that the neural networks were able to identify physically meaningful sets of material parameters so that the constitutive model may predict experimentally observed material behavior in a satisfactory manner.

In an earlier investigation, Al-Haik and Garmestani (2000) used simplistic backpropagation algorithm to capture the creep behavior of polymeric composite. The only input to the network was the loading level and time, and the network predicted the creep strain correctly. This model was expanded to include an extra input; the temperature and was trained via more convenient algorithms such as conjugate gradient and scaled conjugate gradient, Al-Haik (2002). Recently, the authors introduced the truncated Newton algorithm to train neural networks, Al-Haik et al. (2003).

This paper is dedicated to the design of an implicit creep model by means of artificial neural networks (ANN) to predict the creep behavior of a polymer matrix composite under different thermomechanical conditions. At this point we emphasize that the approach we present does not produce an "*explicit formula*" supplying the creep properties for each viscoplastic–temperature–stress–strain law, constructs an appropriate neural network, which, at a given time period "*produces*" a set of data

corresponding to different viscoplastic states. This neural network "*learns*", and if applied to another set of experimental data, may fulfill its task more accurately in shorter time periods. The present methodology may simulate to some extent the experiments after a period of "*learning*". Moreover, the method developed here will be compared to a classical explicit constitutive model for viscoplastic behavior.

2. Neural networks

Neural networks consist of processing elements and weighted connections; Fig. 1 illustrates a typical neural network. Each layer in a neural network consists of a collection of processing elements (PEs), also referred to as neurons, each processing element in a neural network collects the values from all of its input connections, performs a predefined mathematical operation, and produces a single output value. The neural network in Fig. 1 has three layers: F_X, which consists of the neurons F_X $\{x_1, x_2, \ldots, x_n\}$, F_Y $\{y_1, y_2, \ldots, y_p\}$, and F_Z $\{z_1, z_2, \ldots, z_q\}$, where n, p and q are the number of processing elements i.e., neurons at the corresponding layers as shown in Fig. 1. The neurons are connected with weighted connections represented by the arrows in the figure. Each weighted connection acts as both label and a value. For example, in Fig. 1 the connection from F_X neuron x_1 to the F_Y neuron y_1 is the connection weight w_{12}. The value of the connection weights is often determined by a neural network training procedure. It is through the adjustment of the connection weights that the neural network is able to learn. By performing the update operations for each of the neurons, the neural network is able to recall information. The input to a neuron from another neuron is obtained by multiplying the output of the connected neuron by the weight of the connections between them. The artificial neuron then sums up all the weighted inputs coming to it.

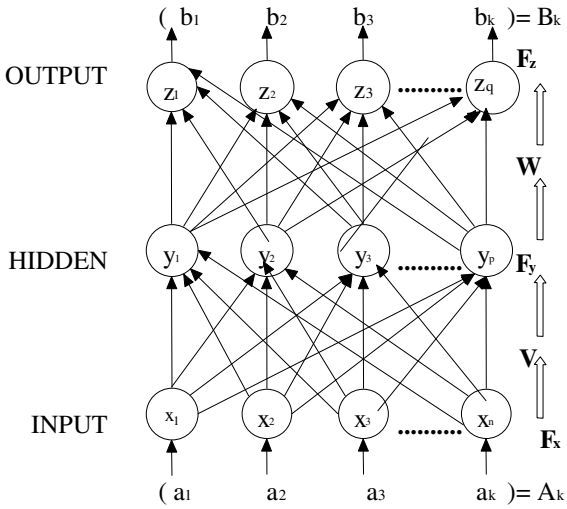

Fig. 1. Topology of the three layer feed-forward backpropagation.

The most common computation procedure performed by a PE is a linear combination of the input value x_i with the abutting connection weights w_{ij}

$$y_i = f\left(\sum_{j=1}^{n} x_i w_{ij}\right), \tag{1}$$

where f is the activation function used to compare the weighted sum of inputs and the threshold value of that neuron. The hyperbolic tangent is the most common activation function

$$f(x) = \tanh(\alpha x) = \frac{e^{\alpha x} - e^{-\alpha x}}{e^{\alpha x} + e^{-\alpha x}}, \tag{2}$$

where $\alpha > 0$.

A training cycle consists of the following steps: An input vector is presented at the inputs together with a set of desired responses, one for each node, at the output layer. A forward pass is done and the errors or discrepancies, between the desired and actual response for each node in the output layer, are found. These are then used to determine weight changes in the net according to the prevailing learning rule. The best-known learning algorithm is the backpropagation algorithm, as introduced by Werbos (1974).

A popular measure of the error E for a single pattern is the sum of the square differences

$$E = \frac{1}{2}\sum_i (t_i - y_i)^2, \tag{3}$$

where t_i is the desired or target response on the ith unit (neuron) and y_i is that actually produced on the same unit.

Learning routine of the backpropagation network assumes that each input pattern a_k is paired with a target pattern b_k representing the desired output (Fig. 1). These are called training pairs. Before training is started, all the weights are assigned to small random values usually in the range $(-1, +1)$. This initialization prevents the network from getting saturated by large values of the gain. Learning is done by iteratively adjusting the weight matrices of the connections in the network W and V, in order to minimize the cost function E. During the process, an input pattern is presented to the network and propagated forward to determine the actual output at the units in the output layer. An error signal for each output PEs is calculated and is *backpropagated* through the network in order to adjust the weights. The learning process continues until the network responds with the sum of the squared error of the output signals becomes less than a predetermined value.

3. Neurocomputing and optimization

3.1. Steepest descent

The standard backpropagation implements the steepest descent method (also called the gradient descent method). At each step of the steepest descent method the

weights are adjusted in the direction in which the error function decreases most rapidly. This direction is determined by the gradient of the error surface at the current point in the weight space. The error function used is the sum of square differences between the actual output and the desired output value for each unit in the output layer. Following the notation given by Sanchez-Sinencio (Sanchez-Sinencio and Lau, 1992), the output error across all the F_z PEs is found using the cost function

$$E = \frac{1}{2}\sum_{j=1}^{q}\left(b_j - z_j\right)^2, \tag{4}$$

where b_j is the desired output while z_j is the actual output. The output of an F_z processing element, z, is computed as

$$z_j = \sum_{i=1}^{p} y_i w_{ij}, \tag{5}$$

where w_{ij} is the weight for the connection between the ith and jth processing elements. And for each F_y (hidden-layer) neuron, the output is computed using

$$y_i = f\left(\sum_{h=1}^{n} a_h v_{hi}\right) = f(r_i), \tag{6}$$

where v_{hi} is the weight for the connection between the hth and ith processing elements. The hidden-layer activation function, f, is a hyperbolic tangent function. Implementing the steepest descent method, the weight adjustment is performed by moving along the cost function in the opposite direction of the gradient to a minimum of the input/output mapping producing the smallest amount of error. For example, in Fig. 1, the connection weights between the F_y and the F_z PEs are adjusted by using the gradient

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial}{\partial w_{ij}}\left[\frac{1}{2}\sum_{j=1}^{q}\left(b_j - z_j\right)^2\right] = \left(b_j - z_j\right)y_i = \delta_j y_i, \tag{7}$$

where δ_j is the error for each F_z PE. Next, the adjustment to the connection weights between the F_x and F_y PEs is found by utilizing the chain rule of partial differentiation, hence, we can calculate the weight changes for an arbitrary number of layers as

$$\frac{\partial E}{\partial v_{hi}} = \frac{\partial E}{\partial y_i}\frac{\partial y_i}{\partial r_i}\frac{\partial r_i}{\partial x_h}\frac{\partial x_h}{\partial v_{hi}} = \sum_{l=1}^{p}(b_l - y_l)y_l w_{hl} f'(r_i)a_h. \tag{8}$$

Having the gradient of the error, then the weight adjustments for the connections are updated in a negative direction to the gradient with a certain rate, as given by

$$\begin{aligned} w_{ij}^{\text{new}} &= w_{ij}^{\text{old}} - \alpha\frac{\partial E}{\partial w_{ij}}, \\ v_{hi}^{\text{new}} &= v_{hi}^{\text{old}} - \beta\frac{\partial E}{\partial v_{hi}}, \end{aligned} \tag{9}$$

where α and β are positive-valued constants that regulate the amount of adjustments made with each gradient move, they are called *learning rates*. The learning rates determine what amount of the calculated error sensitivity to weight change will be used for the weight correction. In standard backpropagation the minimization of the error is performed using a fixed learning rates α and β. The *momentum* term was introduced to allow for more prompt learning while minimizing unstable behavior. Momentum allows the network to ignore small features in the error surface, without which a network may get stuck in a shallow local minimum. On the other hand, when using momentum term, the network can escape from such a minimum. Introducing the momentum term to Eq. (9) we get

$$\begin{aligned} w_{ij}^{\text{new}} &= w_{ij}^{\text{old}} - \alpha \frac{\partial E}{\partial w_{ij}} + \lambda \Delta w_{ij}^{\text{old}}, \\ v_{hi}^{\text{new}} &= v_{hi}^{\text{old}} - \beta \frac{\partial E}{\partial v_{hi}} + \lambda \Delta v_{hi}^{\text{old}}, \end{aligned} \tag{10}$$

where λ is the momentum factor ($\lambda \in [0, 1]$).

3.2. Conjugate gradient

Within a neural network context, conjugate gradient methods can be thought of as extensions of the steepest descent to include variable learning rate.

Starting from some initial guess for the minimum $\mathbf{w}_0$, and initial search direction

$$\mathbf{s}_0 = -\frac{\partial E}{\partial \mathbf{w}_0} = -\mathbf{g}_0. \tag{11}$$

A series of approximations $\mathbf{w}_k$ to minimize the error function E are generated as follows: For $k = 0, 1$, start at point $\mathbf{w}_k$ perform a line minimization of E in the direction $\mathbf{s}_k$. Equivalently, find a scalar α_k such that the function

$$J(\alpha) = E(\mathbf{w}_k + \alpha \mathbf{s}_k) \tag{12}$$

is minimized. Here J is expressed as a function of the learning rate α for fixed values of $\mathbf{w}$ and $\mathbf{s}$. The new estimate of $\mathbf{w}$ is then given by

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \alpha_k \mathbf{s}_k. \tag{13}$$

To choose a new conjugate search direction several methods have been suggested (Fletcher and Reeves, 1964; Polak and Weiss, 1972). In this study we use the approach of Polak and Ribiere as it was proven by Haykin (1999) to be numerically more stable. In Polak and Ribiere algorithm the search direction is updated according to

$$\mathbf{s}_{k+1} = -\nabla E(\mathbf{w}_{k+1}) + \beta_{k+1} \mathbf{s}_k, \tag{14}$$

where

$$\beta_{k+1} = -\frac{\mathbf{s}_{k+1}^T (\mathbf{s}_{k+1} - \mathbf{s}_k)}{\mathbf{s}_k^T \mathbf{s}_k}. \tag{15}$$

Using Eqs. (13) and (14) leads to the expressions for the variable learning rate α_k.

3.3. The truncated Newton algorithm

The training of ANN is basically an unconstrained optimization problem

$$\text{minimize } E(\mathbf{w}) : \mathbf{w} \in R^n \tag{16}$$

where $E : R^n \to R$ is a nonlinear twice continuously differentiable function. We wish to find the local minimizer $\mathbf{w}^*$, for which there exist $\delta > 0$ so that

$$E(\mathbf{w}^*) \leqslant E(\mathbf{w}) \quad \text{for all } \|\mathbf{w} - \mathbf{w}^*\| \leqslant \delta. \tag{17}$$

A first-order necessary condition for a local minimum of E(**w**) is that, $\mathbf{g}(\mathbf{w}) = 0$, which is a system of nonlinear equations, where $\mathbf{g} : R^n \to R$ is the gradient vector of E. A well-known method for solving this system of nonlinear equations is Newton's method. In this method the error function $E(\mathbf{w})$ is approximated locally by a quadratic model and then minimizing this approximation will yield the minimum of the error function. The quadratic model of the error function E at $\mathbf{w}_k$ along $\mathbf{p}_k$ direction is given by the expansion

$$E(\mathbf{w}_k + \mathbf{p}_k) \approx E(\mathbf{w}_k) + \mathbf{g}_k^T \mathbf{p}_k + \frac{1}{2}\mathbf{p}_k^T \mathbf{H}_k \mathbf{p}_k. \tag{18}$$

The minimum of the right-hand side of Eq. (18) is achieved when $\mathbf{p}_k$ is the minimum of the quadratic function

$$q(\mathbf{p}_k) \approx \mathbf{g}_k^T \mathbf{p}_k + \frac{1}{2}\mathbf{p}_k^T \mathbf{H}_k \mathbf{p}_k. \tag{19}$$

Alternatively, such a Newton direction $\mathbf{p}_k$ satisfies the linear system of n simultaneous equations, known as the Newton equation

$$\mathbf{H}(\mathbf{w}_k)\mathbf{p}_k = -\mathbf{g}(\mathbf{w}_k). \tag{20}$$

Starting with an initial guess, $\mathbf{w}_0$, computes the sequence of Newton directions $\mathbf{p}_k$ to update each previous iterate by the formula

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \alpha_k p_k, \tag{21}$$

where $\mathbf{H}(\mathbf{w}_k)$ is the Hessian of the function $E(\mathbf{w}_k)$, α_k is the step size, and $\mathbf{p}_k$ is the descent direction. The problem can be viewed as the solution of a system of linear equations $\mathbf{A}\mathbf{w} = \mathbf{b}$, where $\mathbf{A} = \nabla^2 E(\mathbf{w}_k)$, i.e. the Hessian of $E(\mathbf{w}_k)$, and $\mathbf{b} = -\nabla E(\mathbf{w}_k)$. The iterative method is stopped, "*truncated*", before the exact solution to the Newton equation has been found, giving the method its name (Nash and Sofer, 1989, 1990, 1996). The truncated Newton method consists of nested iterations. There is an outer iteration that corresponds to computing a search direction $\mathbf{p}_k$ that performs a line search to achieve an optimal learning rate α_k. The computation of search direction uses an inner iteration corresponding to the iterative method used to solve the Newton equations, Eq. (20). The only possible obstacle using this method might be the computation of the matrix–vector products $\mathbf{H}(\mathbf{w}_k)\mathbf{p}_k$ for arbitrary vectors $\mathbf{p}_k$ since the Hessian matrix **H** might be singular at

random values of $\mathbf{w}_k$. To avoid a singular Hessian matrix, the concept of "*pre-conditioner*" was introduced by Nash and Sofer (1990), where the Hessian matrix need not be computed or stored. To avoid storing the Hessian matrix, one can consider the following discrete Hessian–vector product based on the following truncated Taylor expansion:

$$\nabla E(\mathbf{w}_k + h\mathbf{p}) = \nabla E(\mathbf{w}_k) + h\nabla^2 E(\mathbf{w}_k)\mathbf{p} + \mathrm{O}(h^2). \tag{22}$$

Hence, the Hessian–vector product $\mathbf{H}(\mathbf{w}_k)\mathbf{p}$ is approximated by

$$\nabla^2 E(\mathbf{w}_k)\mathbf{p} = \lim_{h\to 0} \frac{\nabla E(\mathbf{w}_k + h\mathbf{p}) - \nabla E(\mathbf{w}_k)}{h}, \tag{23}$$

where h is chosen as

$$h = \frac{\sqrt{\varepsilon_{\mathrm{m}}}(1 + \mathbf{w}_k)}{\mathbf{p}}, \tag{24}$$

ε_{m} is the machine accuracy constant and $\mathbf{p}$ is a descent direction.

The inner iteration of the algorithm is terminated using a quadratic truncation test, which monitors a sufficient decrease of the quadratic model

$$\left.\begin{aligned} &\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k \mathbf{H}\mathbf{s}_k, \\ &q_{k+1} = \tfrac{1}{2}(\mathbf{r}_{k+1} + \mathbf{g})^T \mathbf{p}_{k+1} \end{aligned}\right\} \quad \text{if}$$
$$(1 - q_k/q_{k+1}) \leqslant C_q/k \text{ exit the inner loop with search direction } \mathbf{p} = \mathbf{p}_{k+1}, \tag{25}$$

where k is the counter for the inner iteration and C_q is the tolerance, satisfying $0 < C_q < 1.0$.

If

$$C_q \leqslant \|\nabla E(\mathbf{w}_k)\|, \tag{26}$$

then the truncated Newton method will converge quadratically, see Nash and Sofer (1996, p.393) and Dembo and Steihaug (1983). This fact explains the faster rate of convergence and the better quality of results obtained with the truncated Newton method.

Both the conjugate gradient and truncated Newton methods avoid the need for using the Hessian matrix of the error function, i.e. $\mathbf{H}$, However, truncated Newton method goes one step further by generating an approximation to the Hessian matrix. Accordingly when the line search (to find the learning rate) is accurate and we are in close proximity to a local minimum, with positive definite Hessian, truncated Newton method tends to converge faster than the conjugate gradient. Moreover, truncated Newton is not sensitive to accuracy in the line search storage as conjugate gradient method. The implementation of a truncated-Newton method using the approximation to the matrix–vector product (Eq. (23)) requires only the function $E(\mathbf{w})$ and the gradient $\nabla E(\mathbf{w})$ be calculated. Vector storage is needed for the conjugate gradient method and for the line search, but no matrix storage.

4. Phenomenological (explicit) viscoplastic creep model

4.1. Analytical model description

In the following, the viscoplastic model proposed by Gates and Sun (1991, 1995) is presented with respect to the research effort described in this paper. Assuming uniaxial loading where the load is not parallel to the fiber direction, the total strain for elastoplastic (time-independent) constitutive relation may be written as a combination of elastic and plastic terms

$$\varepsilon^{\mathrm{t}} = \varepsilon^{\mathrm{e}} + \varepsilon^{\mathrm{p}}. \tag{27}$$

Hooke's law provides the relation between elastic strain and stress, $\varepsilon^{\mathrm{e}} = \sigma/E$, whereas, the plastic strain, ε^{p}, is expressed by a power law

$$\varepsilon^{\mathrm{p}} = A(\sigma)^n, \tag{28}$$

where A and n are material constants found from the experimental data.

For a rate-dependent constitutive relation, the total strain rate is divided into elastic and viscoplastic components

$$\dot{\varepsilon}^{\mathrm{t}} = \dot{\varepsilon}^{\mathrm{e}} + \dot{\varepsilon}^{\mathrm{vp}}, \tag{29}$$

where elastic strain rate is

$$\dot{\varepsilon}^{\mathrm{e}} = \frac{\dot{\sigma}}{E}, \tag{30}$$

while viscoplastic strain rate is decomposed into two terms

$$\dot{\varepsilon}^{\mathrm{vp}} = \dot{\varepsilon}^{\mathrm{vp}'} + \dot{\varepsilon}^{\mathrm{vp}''}. \tag{31}$$

Differentiating the plastic strain in elastoplastic constitutive relation, Eq. (28) gives the first part of the viscoplastic term, $\dot{\varepsilon}^{\mathrm{vp}'}$

$$\dot{\varepsilon}^{\mathrm{vp}'} = \begin{cases} \left\{ An(\sigma)^{n-1}\dot{\sigma} \right\} & \text{if } \dot{\sigma} > 0, \\ 0 & \text{if } \dot{\sigma} \leqslant 0. \end{cases} \tag{32}$$

Utilizing the "overstress" concept provides the second part of the viscoplastic term, $\dot{\varepsilon}^{\mathrm{vp}''}$, as

$$\dot{\varepsilon}^{\mathrm{vp}''} = \left[\frac{\langle H \rangle}{K} \right]^{1/m}, \tag{33}$$

where H is the overstress, $\langle\ \rangle$ are Macaulay brackets, and K and m are material constants found from the experimental data. The overstress, $H = (\sigma - \sigma^*)$, is considered as a scalar quantity that relates the quasistatic stress, σ^*, to the dynamic or instantaneous stress, σ, at the same strain level, Ho and Krempl (2002). Thus,

$$\dot{\varepsilon}^{\mathrm{vp}''} = \begin{cases} \left[\frac{\sigma - \sigma^*}{K} \right]^{1/m} & \text{if } \sigma > \sigma^*, \\ 0 & \text{if } \sigma \leqslant \sigma^*. \end{cases} \tag{34}$$

The quasistatic stress is found by using previously defined elastoplastic relation

$$\varepsilon = \frac{\sigma^*}{E} + A(\sigma^*)^n, \tag{35}$$

while the dynamic stress is the stress resulting from the time-dependent material behavior.

4.2. Material constants

The material constants K, m, A, and n are temperature dependent and are found from the experimental data. These constants are determined using load relaxation tests.

During the load relaxation, the quasistatic stress is constant, stress rate is negative and the total viscoplastic strain rate is zero, therefore from Eq. (32) $\dot{\varepsilon}^{\mathrm{vp}'} = 0$, and from Eqs. (27) and (29)

$$\dot{\varepsilon}^{\mathrm{vp}} = \dot{\varepsilon}^{\mathrm{vp}''} = -\dot{\varepsilon}^{\mathrm{e}}. \tag{36}$$

Combining Eqs. (29), (34) and (36) results in Eq. (37)

$$\left[\frac{\sigma - \sigma^*}{K}\right]^{1/m} = -\frac{\dot{\sigma}}{E} = \dot{\varepsilon}^{\mathrm{vp}}. \tag{37}$$

Applying polynomial regression to the stress–time data (from a load relaxation test) and differentiating the resulting stress–time curve with respect to time, stress rate, $\dot{\sigma}$, is determined. Applying Gauss–Newton and nonlinear regression methods to Eq. (35) yields σ^*. Once σ, σ^* and $\dot{\sigma}$ are known from a plot of the overstress $(\sigma - \sigma^*)$ against viscoplastic strain rate $(\dot{\varepsilon}^{\mathrm{vp}})$, m and K are computed. Note that m and K are independent of the initial applied load in a load relaxation test as they are computed after the load starts dropping under constant strain condition. Repeating load relaxation tests for different applied loads (strain levels) and calculating the quasistatic stress for each test as mentioned above, a quasistatic stress–strain curve is created. This curve is fit to Eq. (35) to yield the values of A and n.

During creep the stress is constant and the stress rate is zero, therefore, from Eqs. (27)–(30) and (32) the total strain rate may be written in the form of Eq. (38)

$$\dot{\varepsilon}^{\mathrm{vp}} = \dot{\varepsilon}^{\mathrm{vp}''} = \left[\frac{\sigma - \sigma^*}{K}\right]^{1/m}. \tag{38}$$

This relationship is a first-order nonlinear differential equation and is coupled to a nonlinear expression of quasistatic stress through Eq. (35). Combined methods of numerical analysis for solving nonlinear equations and differential equations are required to solve this differential equation.

5. Materials and mechanical testing

The matrix material used in this study is Aeropoxy™. This epoxy is manufactured by PTM&W Industries, Inc. under the brand name PR2032. It is a medium viscosity, unfilled, light amber laminating resin that is designed for structural production applications. This resin laminates very easily, and wets out fiberglass, carbon, and aramid fibers readily. Mixed with PH3665 hardener, this system cures at a temperature of 70 °C and should be cured at least 4 h before moving the structure. The fiber used in this investigation is Thornel Carbon Fiber T-300 12K. This is a continuous length, high strength fiber consisting of 12,000 filaments in a one-ply construction. The typical properties of composite components are listed in Table 1. The composite was fabricated via hand-lay up method and consisted of 3 layers of unidirectional fibers, Fig. 2. To preserve the fiber direction, the fabrics were rolled along the axis of the fibers. The cure cycle for the current thermosetting polymer matrix composite is a two-step cycle. In this cycle, the temperature of the material is increased from room temperature to some elevated temperature (40 °C), which is held constant for the first dwell period (1 h). Afterwards, the temperature is increased again to a second temperature (75 °C) and held constant for the second dwell period (4 h). Next, the material is cooled to room temperature. More detailed procedures of the composite processing can be found in Al-Haik (2002). After curing the composite standard tensile samples were shaped based on the standard ASTM D3039 (250 mm $\times$ 25.4 mm $\times$ 1 mm). The tabs of the sample were made of G-10 fiberglass for its ability to sustain the compression produced by the gripping system. The final microstructure of the composite together with a tensile test fracture surface are shown in Fig. 3(b). The optical microscopy analysis of the microstructure shows that the fibers are distributed in the matrix at random with a volume fraction of 45%.

A mechanical testing system (MTS) was used to conduct tensile, creep, and stress relaxation tests. The setup consists of: testing frame driven by a hydraulic unit and, an extensometer capable of measuring strain with an accuracy of 1 μm strain. A Test Star II controller was used to design, run, and control different tests procedures. Thermal chambers built by ATS (American Testing Standards) company, along with

Table 1
Properties of the composite components

Property	PH2032/PH3660 cured epoxy	Thornel Carbon Fiber T-300 12K
Density	1109.6 kg/m^3	1760 kg/m^3
Mixed viscosity at 25 °C	0.900 N s/m^3	–
Elongation at failure	1.98%	1.4%
Tensile strength	67.76 MPa	3650 MPa
Tensile modulus	2.88 GPa	231 GPa
Glass transition (T_g°)	91.11 °C	–
Filament diameter	–	7 μm
Mix ratio by weight	100:27	–
Volume fraction	55%	45%

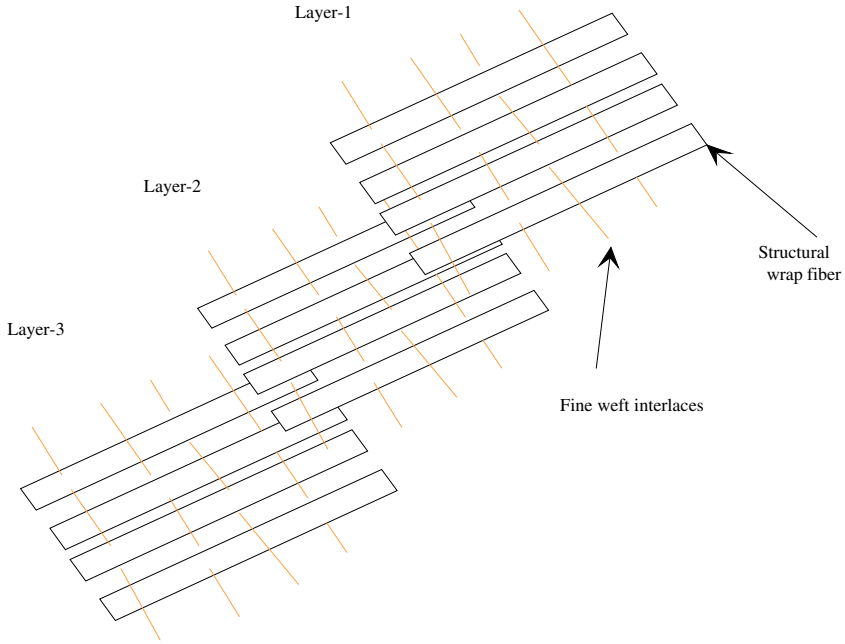

Fig. 2. Three layers of unidirectional fibers were laid-up.

temperature controller were used to simulate different thermal environments at which some tests will be performed.

Tensile tests were based on the ASTM standard D3039/D3039M-95a that describes the specifications for tensile tests to produce mechanical properties of polymeric composites. Initially, tensile tests were performed to produce stress–strain curves at room temperature. Each specimen was loaded at a constant cross head rate of 2 mm/min until failure occurs. The samples were loaded in direction parallel to the fiber axis; x_1 direction as shown in the coordinate system in Fig. 3(c). Then, another tensile test was performed for a sample that was annealed for 4 h at a temperature of 35 °C using the thermal chamber. Similarly, the tensile tests were performed for samples that were annealed for 4 h at temperature of 45, 50, 55, 60, 65, and 75 °C, respectively. Ten samples were tested at each temperature. The results of the tensile tests (average modulus and strength) are presented in Fig. 4 together with Table 2

During load (stress) relaxation test, the sample is subjected to a constant strain while the stress is monitored as a function of time. Although this test is easy to perform for low-modulus materials (e.g., rubber), it is not as easy to do for a composite material with a high-modulus (1 GPa or more). The problem is maintaining the sample at a constant strain. Due to sample slippage or readjustment in the specimen grips, servocontrol is often employed to maintain a constant deformation. Hence, stress relaxation tests conducted on composites are usually controlled using a servohydraulic machine, such as the MTS frame used in the current investigation.

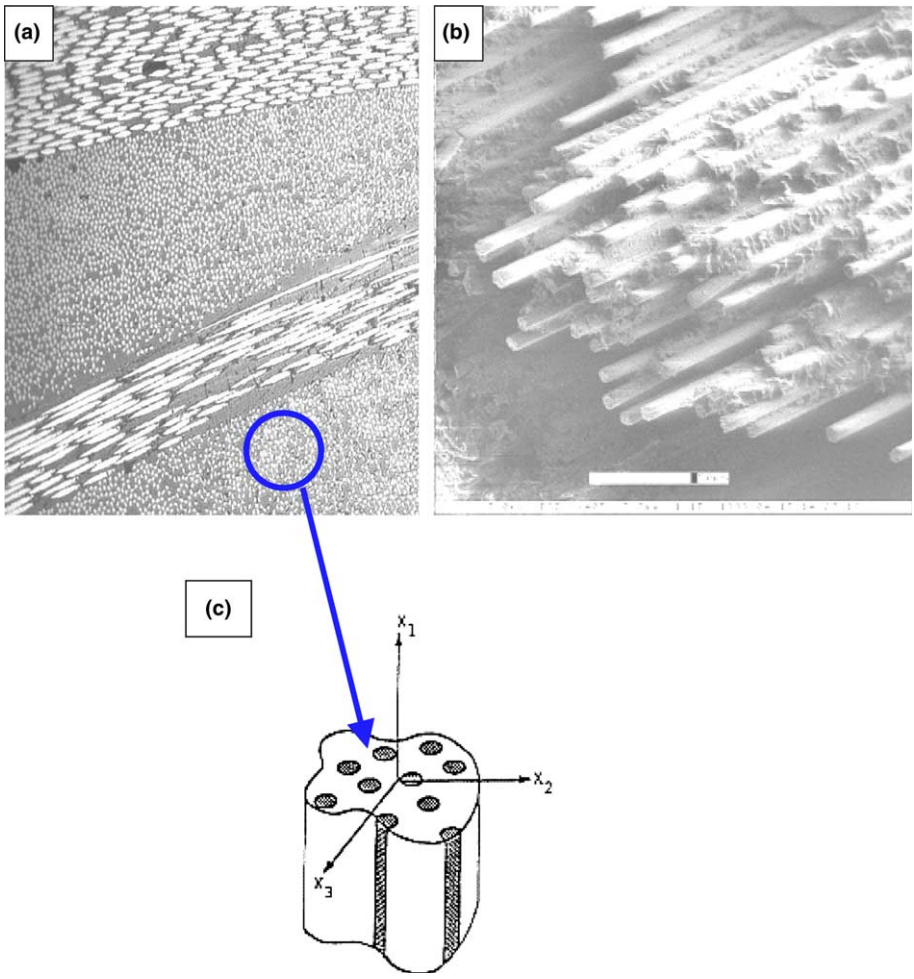

Fig. 3. (a) The microstructure of the Thornel Carbon Fiber/Aeropoxy™ matrix, using optical microscopy (500×). The overall volume fraction of the unidirectional fibers was found to be 45%. (b) ESEM micrograph of the fracture surface of the composite sample failed under tensile test at 25 °C (1400×). (c) Coordinate system for mechanical testing. All mechanical tests were performed along the fiber direction; X_1 axis.

Standard tensile specimens were loaded in the same load frame. Each sample was loaded at a constant displacement rate of 2 mm/min until a desired strain was achieved. At this point, a constant strain was maintained in the specimen using the TestStar II digital controller. The values of the desired strains were chosen from the tensile tests. These are the strain values corresponding to stress levels that were 30%, 40%, 50%, 60%, 70%, and 80% of the strength of the composite at the corresponding test temperature. Each sample was tested to a single combination of temperature-load level, no preloading was performed. By maintaining a constant strain in the

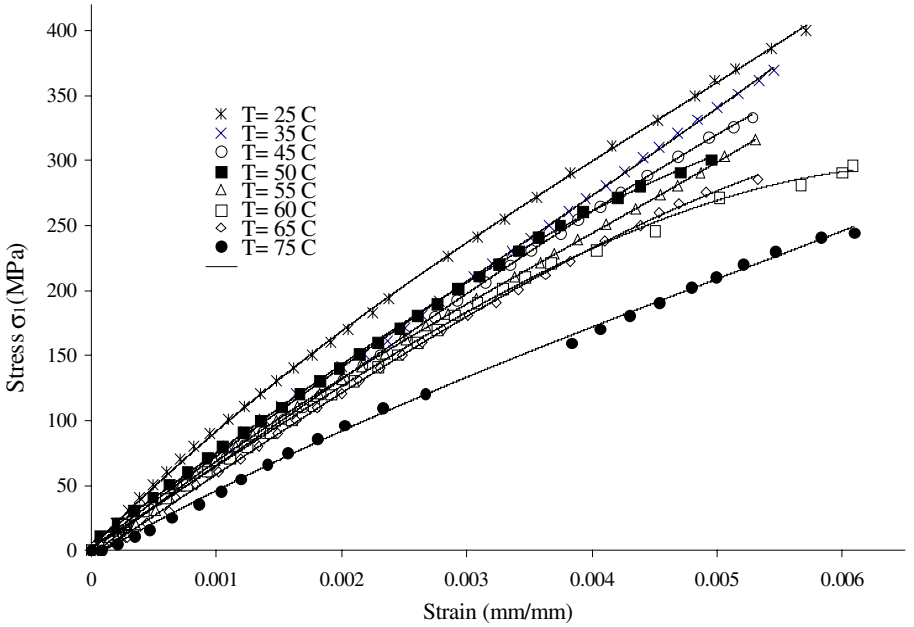

Fig. 4. Tensile tests evolution for the carbon-fiber/Aerpoxy matrix composite at different temperatures. The stress σ_1 is measured along the fiber axis X_1.

Table 2
The composite properties and parameters for viscoplastic model

T (°C)	E (GPa)	σ_u (MPa)	m	K (MPa)	n	A (MPa)
25	81.01	400	0.73	4.10×10^6	3.619	0.0088
35	72.04	369	0.69	1.04×10^5	1.166	0.0749
45	64.46	333	0.70	9.41×10^3	0.630	0.0749
50	62.01	320	0.62	1.16×10^3	1.372	0.0563
55	59.25	315	0.67	2.10×10^3	2.705	0.0247
60	56.84	296	0.69	1.57×10^5	1.518	0.0616
65	49.94	285	0.70	5.80×10^3	1.266	0.0637
75	41.45	244	0.66	1.39×10^6	1.793	0.0975

specimen, the concern about the machine compliance was eliminated. The reduction of nominal stress with time was then measured for 2 h per sample at each strain level using the data acquisition system. Similarly, at each temperature level (25–75 °C), the relaxation tests were conducted for six different samples at different strain levels, 30–80% of the strength at that temperature. Representative curves of the load relaxation experiments at different thermomechanical environments are depicted in Fig. 5.

Creep tests were performed on the same MTS load frame. A constant stress level was achieved using "Test Star II" closed loop servo-hydraulic controller. The strain signal measured by the extensometer was recorded by a data acquisition system.

Each specimen was tested for 1 h under temperature ranges 25–75 °C. Six different samples were subjected to six applied stresses with values of 30–80% of the corresponding strength under each temperature condition. The results for some creep tests are depicted in Fig. 6.

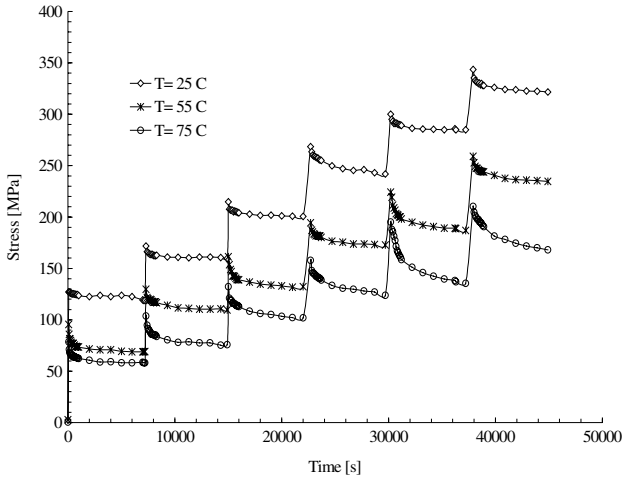

Fig. 5. Representative load relaxation experiments at $T = 25$, 55 and 75 °C temperature regimes. The stress levels are corresponding to the 30%, 40%, 50%, 60%, 70% and 80% of the strength at each temperature.

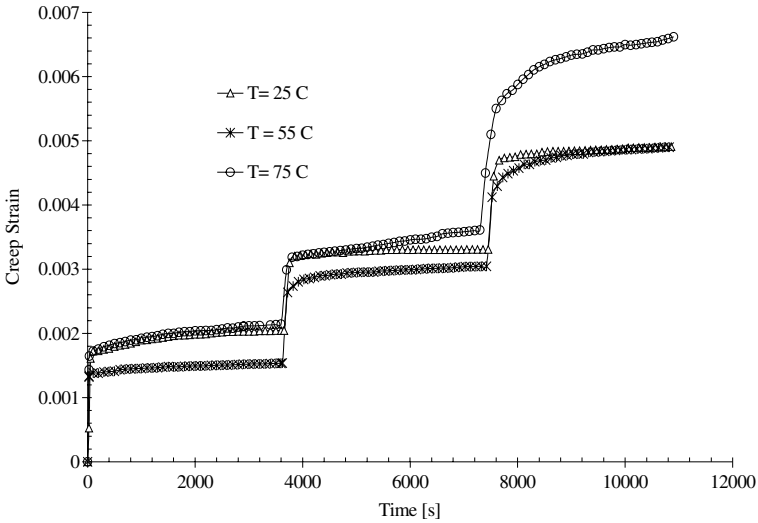

Fig. 6. Representative creep evolution of the composite at $T = 25$, 55, and 75 °C at different strain levels.

6. Results and discussion

6.1. Results for the viscoplastic model

Applying polynomial regression to the stress–time data (from a load relaxation test) and differentiating the resulting stress–time curve with respect to time, stress rate σ is determined. The evolution of strain rate and stress is simulated at different temperature–stress level combinations. Using this type of representation of the stress/time data during relaxation, an asymptotic value was found to represent the minimum or quasistatic value that the stress achieved. This implies that the time for stress relaxation needs only be long enough to capture the decaying curve; hence, the 2-h relaxation tests performed in the current investigation, are actually more than enough to capture this stress decay. Applying Gauss–Newton and nonlinear regression methods to Eq. (37) yields σ^*. Once σ and $\sigma*$ are known from a plot of the overstress against viscoplastic strain rate $\dot{\varepsilon}^{vp}$, a logarithmic plot of overstress versus the plastic strain rate, as shown in Fig. 7, will reveal the values of the parameters m and K at different temperature levels.

Note that m and K are independent on the initial applied load in a load relaxation test. By repeating load relaxation tests for different applied loads (strain levels) and calculating the quasistatic stress for each test as mentioned above, a quasistatic stress–strain curve is created. This curve is fit to Eq. (35) in a logarithmic fashion to yield the values of A and n, as shown in Fig. 8.

It was reported by Gates (1993a) that there is no uniform trend on how the materials parameters (A, n, m and K) are behaving with temperature. However, he reported one consistent trend that deals with a remarkable increase in the parameter

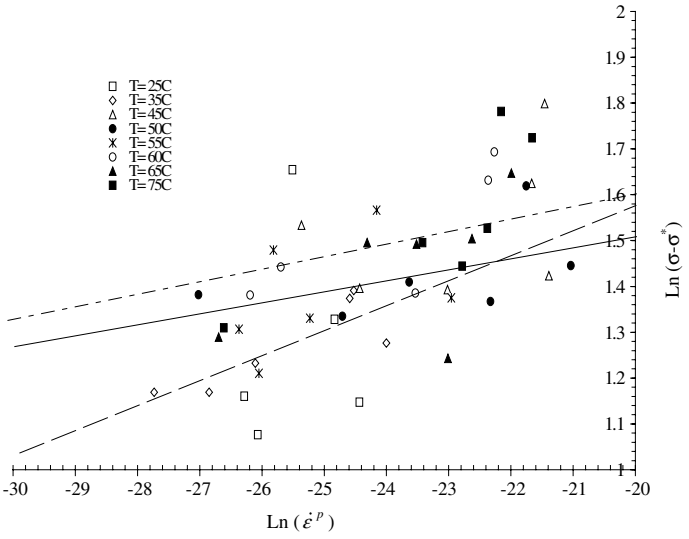

Fig. 7. Logarithm of overstress versus logarithm of strain rate at eight temperatures. These curves are used to determine the material constants K and m at each temperature. Overstress is in MPa.

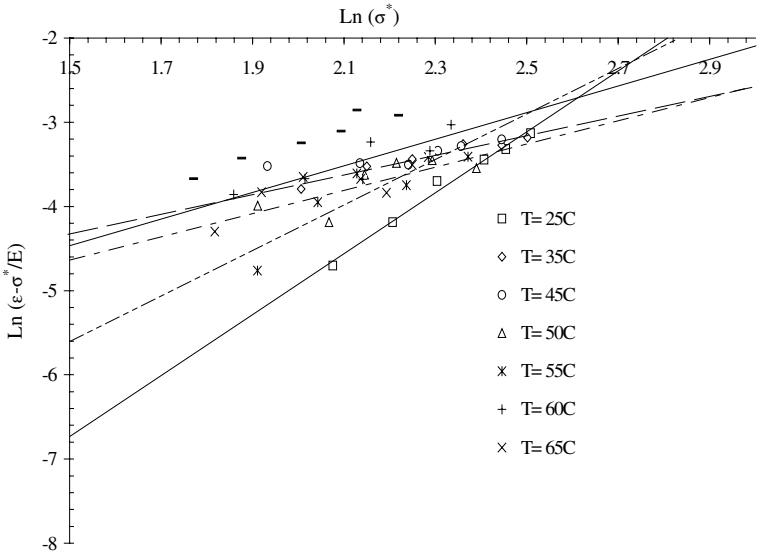

Fig. 8. Logarithm of plastic strain ε^{p} versus logarithm of quasistatic stress σ^{*} at eight different temperatures. These curves are used to determine the material constants A and n at each temperature. Quasistatic stress is in MPa.

A as the glass transition temperature (T_g°) of the matrix was approached. This trend is reported in the current work: when approaching 75 °C, which is closer to the *measured* glass transition temperature of the polymer matrix ($T_g = 86$ °C, using DSC), the parameter A reached a larger value. Gates used a parametric approach to investigate the effect of temperature by fixing the value of one of the parameters throughout the temperature variation, and then investigating the relation between the temperature and other parameters (Gates, 1993a). This ambiguity suggests that there are other factors affecting those constants such as the stress levels and essentially the reduction in the glass transition range at elevated temperature conditions. Glass transition temperatures for samples annealed at 70 °C for 4 h were found to be 71 °C compared to 86 °C for samples not annealed before testing. These results were obtained via differential scanning calorimetry (DSC).

The creep strain is calculated by solving Eq. (38). Combined methods of numerical analysis for solving nonlinear equations (Newton method) and differential equations (Runge–Kutta) were implemented to solve this differential equation. The simulation of the viscoplasitc model was carried out for three temperatures: 35, 50, and 65 °C, under different stress levels. It is important to emphasize that the averaging of the parameters n and m was not implemented in the current study, all parameters were assumed to be temperature-dependent as opposed to the argument given by Gates (1993a). The results of the simulation together with the experimental and neural network results (will be presented later) are shown in Figs. 15–17.

The advantage of the empirical elastic/viscoplastic model is that: it utilizes the representation of stress/time data during relatively short-term load relaxation tests to

predict a relatively longer-duration creep behavior. For example, in Gates (1992) investigation, stress relaxation test duration were 100 s and they used to predict creep up to 450 s. However, the model assumes that the behavior of the composite is uniform over different levels of temperature and stress, even at temperatures close to the glass transition of the composite.

The results of the viscoplastic model at a low temperature of 35 °C and minimum load levels (Fig. 15) are comparable to the experimental data. However, as the load level rises, the model shows discrepancies compared to the experimental results. At higher temperature levels, 50 °C (Fig. 16) and 65 °C (Fig. 17), there are significant errors between the model and the actual creep tests; the model tends to predict higher strains compared to the actual creep experiments.

The simulation of the viscoplastic model in the current study concludes that: while the model predicts the creep at low temperature and low stress levels accurately, it fails to produce an acceptable evolution of the creep at high stress levels and/or at temperatures close to the T_g° of the composite.

The viscoplastic model is phenomenological in nature; a number of tests with different loading conditions (tensile and creep) should be performed over reasonable levels of temperature–stress levels, such that the internal parameters of the composite can be extracted. There are no clear trends on how the parameters are affected by the temperature. Moreover, the viscoplastic creep model carried out in this study is not expected to provide explicit description of all the many rate-dependent phenomena that may occur in polymer-based composites at elevated temperatures.

6.2. Results for neurocomputational creep model

The proposed input to the neural network is an array of 3×1 cells where the elements of each cell represents the [*temperature*; *normalized stress level*; *time*], respectively. The targets were chosen to be the corresponding values of the creep strain for each input cell. The normalization of the stress level was done with respect to the strength at the working temperature T.

The datasets were produced from a combination of eight temperatures (25, 35, 45, 50, 55, 60, 65, and 75 °C) that were selected according to the measured glass transition temperature of the polymer ($T_g = 86$ °C), six normalized stress levels (30%, 40%, 50%, 60%, 70%, and 80%), and 36 time steps with 100 s increment, i.e. $100, 200, \ldots, 3600$ s. The total number of data points produced was calculated as: $8(T^\circ) \times 6(\sigma) \times 36(t) = 1728[T^\circ, \sigma, t]$ input-target cells.

Since neural networks computations do not favor the "*raw*" data to be used for training, the input data $[T^\circ, \sigma, t]$ and the targets $[\varepsilon]$ were normalized to values ranging between -1 and 1 using the following formula:

$$x_n = 2\frac{x - x_{\min}}{x_{\max} - x_{\min}} - 1, \tag{39}$$

where x_n is the normalized value of the vector $x = [T^\circ, \sigma, t, \varepsilon]$, x_{min} and x_{max} are the minimum and maximum values in the database for the vector x. After scaling the 1728 cells of inputs-targets, the scaled results will be split into three subsets; one set will be

used for training, another set for validation, and the last set for testing the network performance. The training and validation sets consist of 1200 (roughly 2/3 of the entire dataset) pairs of scaled data that covers the temperature ranges of 25, 45, 55, 60, and 75 °C. These 1200 data points (37) will be split into 800 pairs for training and 400 pairs for validation. The remaining 528 cells left that cover the creep tests at temperatures: 35, 50, and 65 °C will be kept aside to test the neural network prediction of creep after the network has been trained and validated. The scaled data sets should be randomized so that the training process of the network does not consist of a table look up problem, and to eliminate any bias that might exist in the training dataset.

Based on the Universal Approximation Theorem, Hornik et al. (1989) proved that "*A two hidden layer network is capable of approximating any useful function*". Also, Hornik stated that the mapping power of feed forward neural network is not inherent in the choice of a specific activation function; rather "*it is the multilayer feed forward structure that leads to the general function approximation capability*".

Following the reasoning provided by Hornik et al. (1989), the current investigation adopts a two-hidden layer neural network structure. The number of neurons at each hidden layer was obtained through training the network using a standard backpropagation algorithm with two design parameters: learning rate and momentum. The performance of several structures of neural networks was investigated by Al-Haik (2002) where the **[6-20-1]** structure (six neurons at the first hidden layer, 20 neurons at the second hidden layer and a single neuron at the output layer) achieved an optimal number of neurons. This optimal structure produced MSE = 0.12105 (Al-Haik, 2002). The results of this crude network will be used to further improve the

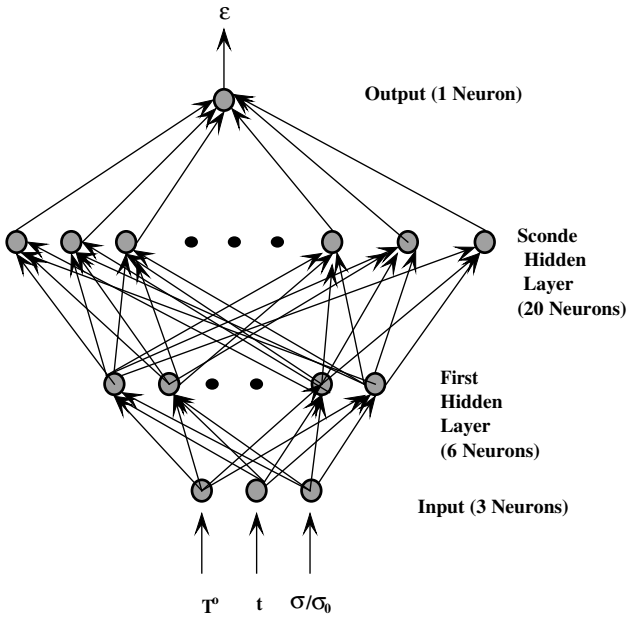

Fig. 9. The structure of the proposed neural model to predict the creep behavior of PMC.

performance index; i.e. the mean square error. The structure of this optimal-size network is shown in Fig. 9. For this network, both hidden layers have tansigmoidal activation functions φ_1 and φ_2, respectively (Eq. (2)), while the activation function of the output layer is a linear function φ_3. Thus, the objective function built through the proposed neural networks structure can be written in a compact notation as

$$f_k(\mathbf{x},\mathbf{w}) = \varphi_3\left(\mathbf{w}^3_{1\times 20}\varphi_2\left(\mathbf{w}^2_{20\times 6}\varphi_1\left(\mathbf{w}^1_{6\times 3}\mathbf{x}\right)\right)\right), \tag{40}$$

where the superscript of the matrices $\mathbf{w}$ represents the layer number, and subscript represents the size of the weight matrix. The corresponding objective function to be minimized is then given as the mean square error between the approximation function f and the actual target t for each training pair $k = 1, 2, \ldots, 798$

$$E(\mathbf{w}) = \frac{1}{2}\sum_{k=1}^{798}(t_k - f_k)^T(t_k - f_k). \tag{41}$$

This function should be minimized with respect to the weight values $\mathbf{w}^1$, $\mathbf{w}^2$, and $\mathbf{w}^3$ that can be stacked into one vector $\mathbf{w}$ consisting of: 3×6 values, at first hidden layer, 6×20 values at second hidden layer, and finally 20×1 values at the output layer, hence $\mathbf{w}$ is a vector of 158 variables. Training the neural network consists of finding the optimal values of $\mathbf{w}$ that minimize the error function $E(\boldsymbol{w})$ using the optimization techniques described earlier.

Using the steepest descent algorithm with momentum requires fixed values for both the learning rate (α) and the momentum coefficient (λ). Several combinations of these tow parameters were used to determine an optimal value of the error function.

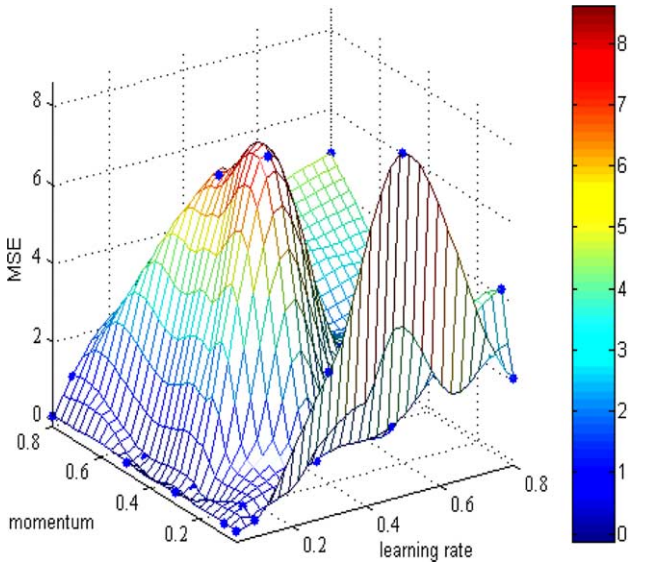

Fig. 10. The behavior of the MSE error as calculated using different values of learning rate and momentum for the steepest descent method.

The corresponding MSE surface can be visualized as shown in Fig. 10. The visualized MSE surface is highly nonsmooth. However, the values of the learning rate of $\alpha = 0.10$ and momentum $\lambda = 0.3$ generated the least MSE = 0.119678, that is much higher than the pre-assigned MSE goal of 1×10^{-5}.

For testing the network we introduce a completely new dataset that does not belong to the training and validation data. For example, one can test the network at $T = 35$ °C with stress levels 30%, 50%, and 80%, respectively, over 1 h simulation time span. The performances for training, validation, and test sets are simulated as shown in Fig. 11. The network was trained for 1000 iteration (epoch) to check if the performance (MSE) for either validating or testing sets might diverge, which did not occur as shown in Fig. 12.

By applying it to the implicit creep model, the conjugate gradient algorithm will be used to determine the weights updates. The **[6 × 20 × 1]** structure with tansgimoidal activation functions in a batch mode (update of all neurons weights takes place after entire training session is fulfilled) will be used; the gradient of the error function is computed after the entire training has been presented to the entire network.

Similarly, the training with the conjugate gradient algorithm, validation, and testing sets as in the standard backpropagation were used. Compared to the steepest descent backpropagation, the conjugate gradient minimization algorithm produced a smaller MSE for all the three phases of training, validation, and testing. Fig. 12 shows that the resulting MSE error for the conjugate gradient training algorithm is 50% less than that for the standard backpropagation. Another important conclusion that can be drawn from Fig. 12 is that the conjugate gradient with line search backpropagation is an order of magnitude faster than the steepest descent backpropagation. The conjugate gradient with line search required exactly 113 epochs for the MSE (for the training set) to drop to a value of 0.06997, compared to 1000

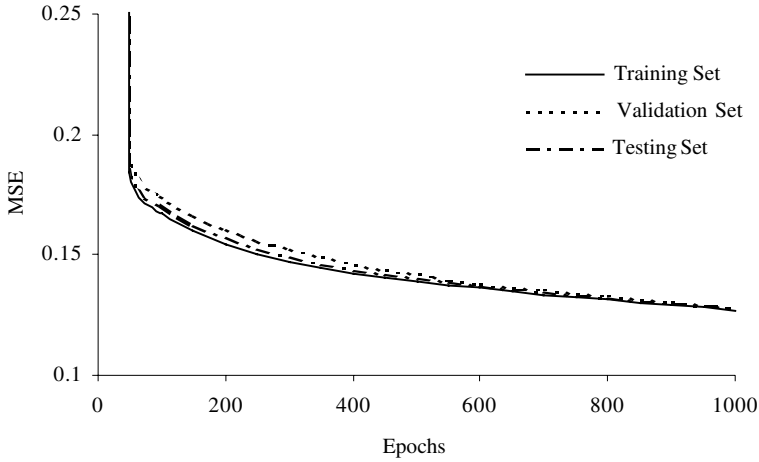

Fig. 11. MSE error for training, validation, and testing sets, for the **[6 × 20 × 1]** ANN based on the steepest descent with momentum backpropagation training algorithm.

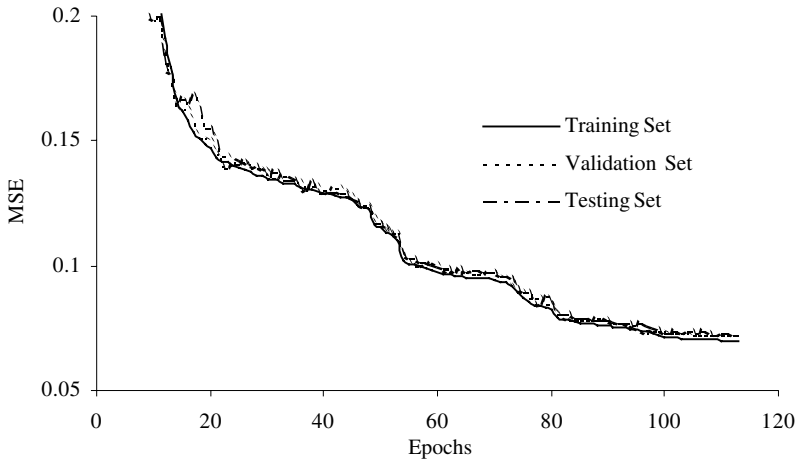

Fig. 12. MSE error for training, validation, and testing sets for the **[6 × 20 × 1]** ANN with backpropagation training algorithm that utilizes the conjugate gradient (Polak–Ribiere) algorithm.

epochs required to reach a value of 0.12707 MSE for the standard backpropagation method.

We conclude that the standard conjugate gradient algorithm is by far more superior to the standard backpropagation in both reducing the mean squared error in less number of epochs, and in better generalization for the network for the test dataset. However, these attractive results were achieved at the expense of additional computational effort, namely the line search technique in order to achieve the optimal learning rate that will be modified at each successive step to reach the goal of minimizing the mean square error function.

While also converging linearly, the conjugate gradient method has better convergence rate in particular when the condition number of the Hessian matrix **H** is large (Nash and Sofer, 1996). Condition number κ, is given by the multiplication of the Hessian norm by the inverse Hessian norm

$$\kappa(H) = \|H\|\|H^{-1}\|. \tag{42}$$

Implementing the same architecture of the implicit creep model network; **[6 × 20 × 1]**, and using now the truncated Newton minimization as the training algorithm, the error function as a stopping criteria, the algorithm converged to a satisfactory error of 0.01323 after 160 epochs as shown in Fig. 13.

The performance of the truncated Newton algorithm was compared to that of both the steepest descent and the nonlinear conjugate gradient as shown in Table 3 The truncated Newton algorithm attained the lowest MSE requiring only a relatively moderate number of epochs. This is to be expected in view of the quadratic convergence rate of the TN.

Before concluding the neuro-creep model, it is recommended to investigate how the initial weight matrices affect the MSE results for the scaled conjugate gradient algorithm. This is mainly because the weight is the argument of the performance

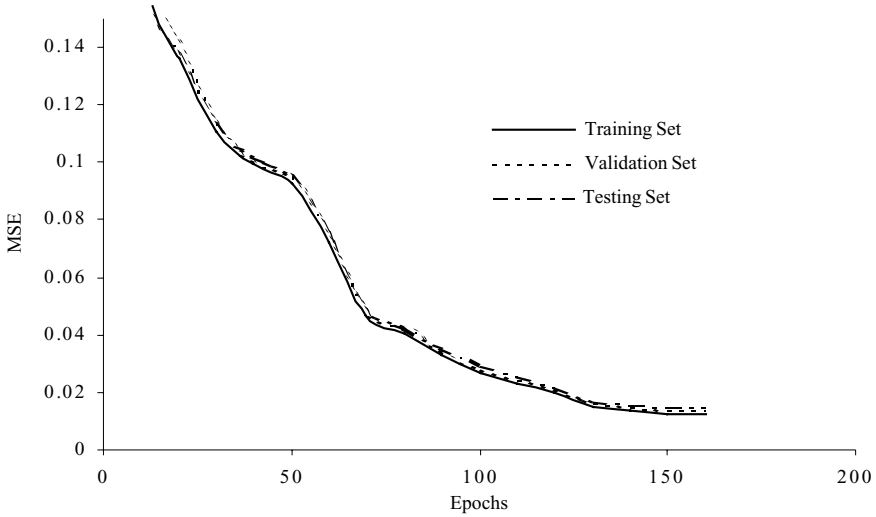

Fig. 13. MSE error for training, validation, and testing sets, for **[6 × 20 × 1]** ANN with backpropagation training algorithm that utilizes the truncated Newton method.

Table 3
Performance of three different algorithms for the implicit creep model using the **[6 × 20 × 1]** topology with tansigmoid activation functions in the hidden layers

Training algorithm	MSE	Epochs
Steepest descent	0.11967	1000
Conjugate gradient (Polak–Ribiere)	0.06997	113
Truncated Newton with CG-preconditioner	0.01323	160

function so when minimizing the MSE function, basically we are looking for the weight vector w that will minimize $\mathbf{E}(\mathbf{w})$ in Eq. (41). The following analysis will investigate how good was the initial estimate $\mathbf{w}_0$.

From the dimension of the hidden layers one can conclude the size of the first weight matrix between the input and the first hidden layer to be $[3 \times 6]$ and the size of the weight matrix between the first and the second hidden layers is $[6 \times 20]$. Finally, the size of the weight matrix between the second hidden layer and the output is $[20 \times 1]$.

It is feasible to do some analysis regarding the weight matrix between the input and the first hidden layers because the weight structures can be represented in 3-D space, obviously we need a 6-D space to do similar analysis for the weight matrix between the two hidden layers, which is clearly cannot be done, unless we use several contours to project it onto 3-D space. The 3×6 structure of the weight matrix between the input and the first hidden layers can be tracked through out different epochs using the scaled conjugate gradient algorithm. The purpose is to see how the

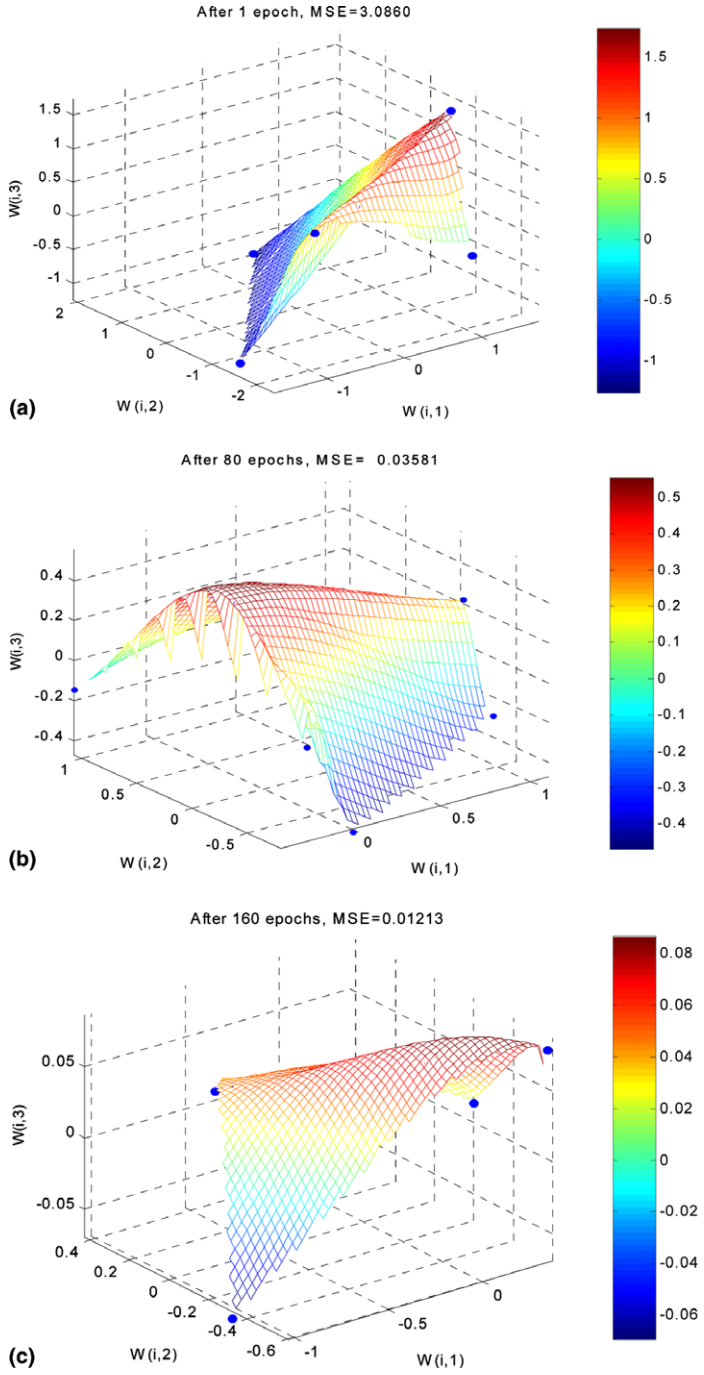

Fig. 14. The simulation of weights optimization for the 3×6 weight matrix between the input layer and the first hidden layer after implementing the truncated Newton training algorithm for: (a) 1, (b) 80 and (c) 160 epochs.

initial weight matrix with preassigned entries of 0.5 behaves as the training algorithm is implemented at successive iterations. The space of the 3×6 weight structure was simulated at Fig. 14. The weight space was noticed to be smooth at the different epochs, also the span of the weight structure shrank from $[(-1,1),(-2,2),(-1.5,1.5)]$ after the first iteration to $[(-1,0.5),(-0.6,0.4),(-0.1,0.1)]$ when convergence was achieved after 160 epochs. The most significant shrinking of the weight space is attributed to the third component, which is the creep time (first component corresponds to temperature and second component to the stress level). This domination of the weights belonging to the time component of the model is not surprising, since for the training dataset the time had larger domain (36 different pattern: 100, 200, 3600 s) compared to stress level (six different patterns: 30%, 40%, 80%) and temperature (five different patterns: 25, 45, 55, 60, and 75 °C). This implies that for each variable in the training model (time, stress level, and temperature), the more different patterns exist for each variable in the training set, the more involved this variable in determining the optimal weights for the training algorithm.

The experimental creep behavior of the composite together with viscoplastic model simulation and the neural network simulation are presented in Figs. 15–17.

Unlike the explicit viscoplastic model, neural network model predicted more accurate results at different stress–temperature conditions especially under conditions of relatively high temperature. Simulating the creep behavior of the composite at temperatures (65 °C) closer to the glass transition temperature of the composite (86

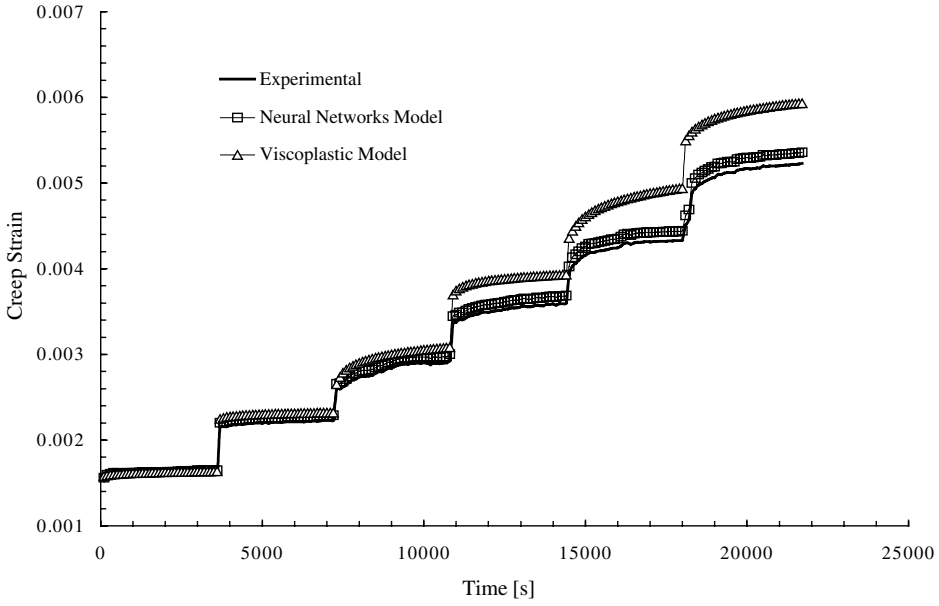

Fig. 15. Validation of the viscoplastic and neural network models (truncated Newton) for the creep evolution at $T = 35$ °C and stress levels 30%, 40% and 50%, 60%, 70% and 80% of the composite strength at this temperature.

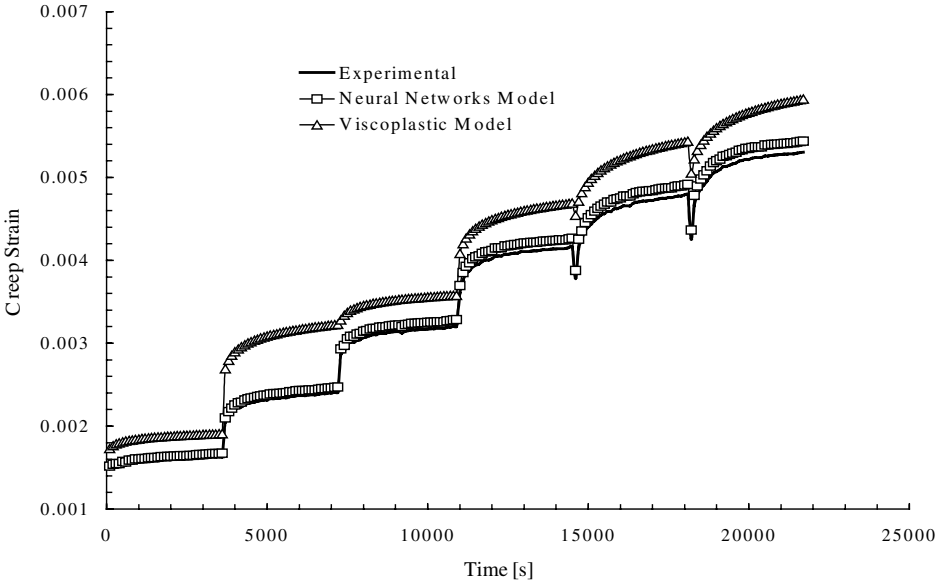

Fig. 16. Validation of the viscoplastic and neural network models (truncated Newton) for the creep evolution at $T = 50$ °C and stress levels 30%, 40%, 50%, 60%, 70% and 80% of the composite strength at this temperature.

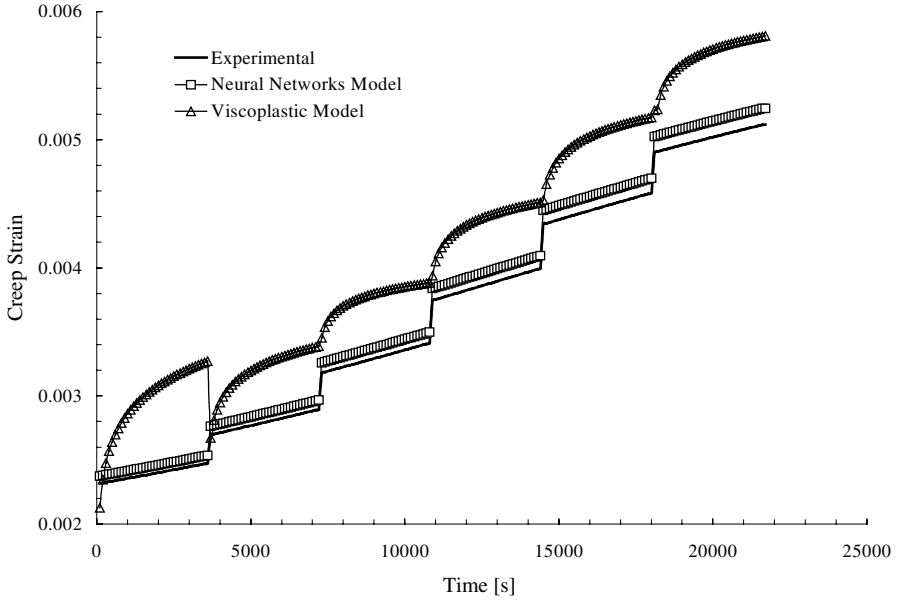

Fig. 17. Validation of the viscoplastic and neural network models (truncated Newton) for the creep evolution at $T = 65$ °C and stress levels 30%, 40%, 50%, 60%, 70% and 80% of the composite strength at this temperature.

°C), revealed the discrepancy of the explicit viscoplastic model to capture the actual behavior of the composite in the vicinity of the glass transition region.

The proposed model can be extended to account for off axis testing. However, the off axis angle should be added as a forth parameter together with temperature, time and stress level. In this case the creep data should be expanded to include this new parameter. Hence some creep tests should be conducted at different off axis angles (for example 0°, 45° and 90°). Then the size of neural network weight matrices will be changed, for example when we had three parameters the input layer weight matrix was $[3 \times 1]$, accounting for the off axis angle θ, the size of this matrix will be $[4 \times 1]$. Correspondingly finding an optimum number of neurons will be similar for the three parameters cases and it might yield different number of neurons at each hidden layer, but the output layer will still have one neuron since we are predicting one output that is the creep strain. So in short to account for the off axis angle, some data from creep tests performed under different off axis angles must be added to the training pairs.

In a similar approach, Mukherjee et al. (1995) investigation had demonstrated the capability of neural networks to predict the strengthening in the transverse direction of metal matrix composites. The fibers were loaded at two different angles 0° and 45°. The results of the two loading angles were presented by two different neurons. It was possible to mix the two loading angles in different proportion (different weight), which roughly simulates different loading angles. To mix the two loading angles a value between 0 and 1 had been input in the neurons corresponding to 0° and 45° loading directions. The summation of the two entries is always 1. Hence, an entry of 1 in the neuron for 0° loading and 0 for the neuron for 45° loading signifies that all fibers are loaded at 0° and vise versa. By changing the above two inputs the two directions of loading could be mixed in different proportions to simulate other loading directions, i.e., by assigning different weights values for each of the two neurons.

Similar approach can be carried out in our investigation, by adding two neurons one accounts for creep at $\theta = 0°$ and $\theta = 90°$ loading directions. By changing the values of the weights for these two neurons, they can account for angles within the 0–90° loading angle range.

7. Conclusions

The modulus and strength of the carbon-fiber/Aeropoxy™ matrix composite system decrease rapidly at high temperatures; both strength and stiffness dropped to almost 50% at a temperature of 75 °C. The viscoplastic behavior of the composite was captured using load relaxation and creep data. An elastic–viscoplastic constitutive model was used for the modeling efforts. The temperature-dependent material constants were obtained by applying the results from load relaxation tests to the theoretical model. Short-term creep tests for different temperatures were performed. The experimental results for the short-duration creep tests were compared to the results of the model. The phenomenological model showed a close agreement with experimental data at low temperature–low stress conditions. However, the elevated temperature

investigation revealed the discrepancy of the explicit viscoplastic model to capture the actual behavior of the composite in the vicinity of the glass transition region.

An alternative model employed neural networks formulation to capture the creep behavior of the composite. The neural network model was built directly from the experimental results obtained via creep tests performed at various stress–temperature conditions. The optimal structure of the neural network was achieved through the universal approximation theory and the dimensionality of the creep problem (stress, temperature, and time). The neural network model was trained to predict the creep strain based on the stress–temperature–time values. The performance of the neural model is represented by the mean squared error between the neural network prediction and the experimental creep strain results. To minimize this error, several optimization techniques were examined. The minimization of the error when carried out by the truncated Newton method outperforms both the steepest descent and conjugate gradient methods in terms of convergence rate and accuracy. Truncated Newton achieved a desired quadratic convergence rate while limiting the waste of oversolving the model at points far from the solution, where the steepest descent method usually fails. The truncated Newton method was easily customized to fit the neurocomputational creep model, eliminating the randomness encountered for choosing an optimal learning rate and momentum for the steepest descent method.

Unlike the explicit viscoplastic model, neural network model utilizing the truncated Newton algorithm predicted more accurate results at different stress–temperature conditions. Moreover, in building the neural network creep model, only one type of data is required, that is creep data at different thermomechanical histories, while viscoplastic model requires both tensile tests data together with load relaxation data, and of course creep data still required to verify the performance of the model.

Acknowledgements

The authors are grateful to the Structural Research Center, Florida Department of Transportation for providing the Materials used in this research. We also acknowledge Prof. I.M. Navon from the School of Computational Science and Information Technology-Florida State University for his valuable remarks on the optimization algorithms.

References

Al-Haik, M.S., 2002. Durability of a polymer matrix composite: a neural network approach. Ph.D. Thesis, Mechanical Engineering, Florida State University, Tallahassee, FL.

Al-Haik, M.S., Garmestani, H., 2000. Implicit creep model of polymer matrix composites by means of neural network. In: ICCE/7 Conference, Denver, CO.

Al-Haik, M.S., Garmestani, H., 2001. Durability study of polymeric composites for structural application. Journal of Polymeric Composites 22 (6), 779–792.

Al-Haik, M.S., Garmestani, H., Navon, I.M., 2003. Truncated-Newton training algorithm for neurocomputational viscoplastic model. Computer Methods in Applied Mechanics and Engineering 192, 2249–2267.

Al-Haik, M., Vaghar, M., Shahaway, M., Garmestani, H., 2001. Viscoplastic analysis of structural polymer composites using stress-relaxation and creep data. Composites B 32, 165–170.

Brinson, L.C., Gates, T.S., 1995. Effects of physical aging on long-term creep behavior of polymers and polymer matrix composites. International Journal of Solids and Structures 32, 827–846.

Brinson, L.C., Lin, W.S., 1998. Comparison of micromechanics methods for effective properties of multiphase viscoelastic composites. Composite Structures 41, 353–367.

Chan, K.S., 1988. The constitutive representation of high-temperature creep damage. International Journal of Plasticity 4 (4), 355–370.

Dillard, D.A., 1991. Viscoelastic Behavior of Laminated Composite Materials. Elsevier Applied Science Publishers Ltd, England.

Daoheng, S., Qiao, H., Hao, X., 2000. A neurocomputing model for the elastoplasticity. Computer Methods in Applied Mechanics and Engineering 182, 177–186.

Dembo, P.S., Steihaug, T., 1983. Truncated Newton algorithms for large-scale unconstrained optimization. Mathematical Programming 26, 190–212.

Fletcher, R., Reeves, C.M., 1964. Function minimization by conjugate gradients. Computer Journal 7, 149–154.

Fotiu, P.A., Nemat-Nasser, S., 1996. Overall properties of elastic–viscoplastic periodic composites. International Journal of Plasticity 12 (2), 163–190.

Freed, A.D., Walker, K.P., 1993. Viscoplasticity with creep and plasticity bounds. International Journal of Plasticity 9 (2), 213–242.

Furukawa, T., Yagawa, G., 1998. Implicit constitutive modeling for viscoplasticity using neural networks. International Journal for Numerical Methods in Engineering 43, 195–219.

Furukawa, T., Okuda, H., Yagawa, G., 1995. A neural network constitutive law based on yield and back stresses. The 8th Computational Mechanics Conference 95 (4), 121–122.

Furukawa, T., Okuda, H., Yagawa, G., 1996. Implicit constitutive modeling using neural networks. In: XIXth International Congress of Theoretical and Applied Mechanics, JJ-3, p. 531.

Garmestani, H., Al-Haik, M.S., Downey, S., 2001. Durability analysis of an autoclave structural polymer composite using stress relaxation and creep data. In: Third Canadian International Composite Conference, Montréal, Canada.

Gates, T.S., 1992. Experimental characterization of nonlinear rate dependent behavior in advanced polymer matrix composites. Experimental Mechanics 32 (1), 68–73.

Gates, T.S., 1993a. Effects of elevated temperature on the viscoplastic modeling of graphite/polymeric composites. In: Harris, C.E., Gates, T.S. (Eds.), High Temperature and Environmental Effects on Polymeric Composites. American Society for Testing and Material, ASTM STP 1174, pp. 201–221.

Gates, T.S., 1993b. Matrix dominated stress/strain behavior in polymeric composites: effects of hold time, nonlinearity and rate dependency. In: Camponeschi Jr., E.T. (Ed.), Composite Materials: Testing and Design. American Society for Testing and Materials, ASTM STP 1206, pp. 177–189.

Gates, T.S., Sun, C.T., 1991. An elastic/viscoplastic constitutive model for fiber reinforced thermoplastic composites. AIAA Journal 29 (3), 457–463.

Gates, T.S., Chen, J.L., Sun, C.T., 1995. Micromechanical characterization of nonlinear behavior of advanced polymer matrix composites. In: Saff, C.R., Deo, R.B. (Eds.), Composite Materials: Testing and Design. American Society for Testing and Materials, ASTM STP 1274, pp. 295–319.

Govindarajan, S., Langrana, N.A., Weng, G.J., 1997. An experimental and theoretical study of creep of a graphite/epoxy woven composite. Polymer Composites 17 (3), 353–361.

Haplin, J.C., 1968. Introduction to Viscoelasticity. Technomic Publishing Co, London, UK.

Haykin, S., 1999. Neural Networks: A Comprehensive Foundation, second ed. Prentice-Hall, Englewood Cliffs, NJ.

Ho, K., Krempl, E., 2002. Extension of the viscoplasticity theory based on overstress (VBO) to capture non-standard rate dependence in solids. International Journal of Plasticity 18 (7), 851–872.

Hornik, K., Stinchocombe, M., White, H., 1989. Multilayer feedforward networks are universal approximators. Neural Networks, 359–366.

Huber, N., Tsakmakis, Ch., 2001. A neural network tool for identifying the material parameters of a finite deformation viscoplasticity model with static recovery. Computer Methods in Applied Mechanics and Engineering 191, 353–384.

Kan, K.N., Reutov, A.I., Reutov, Y., Fishko, V.N., 1987. Prediction of the reliability of products made of polymer composites for conditions of creep and stress relaxation. Mechanics of Composite Materials 23 (4), 523–527.

Khan, A., Pamies, O.L., 2002. Time and temperature dependent response and relaxation of a soft polymer. International Journal of Plasticity 18 (10), 1359–1372.

Khan, A., Zhang, H., 2001. Finite deformation of a polymer: experiments and modeling. International Journal of Plasticity 17 (9), 1167–1188.

Krempl, E., Khan, F., 2003. Rate (time)-dependent deformation behavior: an overview of some properties of metals and solid polymers. International Journal of Plasticity 19 (7), 1069–1095.

Li, J., Weng, G.J., 1998. Time-dependent creep of a dual-phase viscoplastic material with lamellar structure. International Journal of Plasticity 14 (8), 755–770.

Liu, M.C., Krempl, E., 1979. A uniaxial viscoplastic model based on total strain and overstress. Journal of Mechanics and Physics of Solids 27, 377–391.

Lubarda, V.A., Benson, D.J., Meyers, M.A., 2003. Strain-rate effects in rheological models of inelastic response. International Journal of Plasticity 19 (8), 1097–1118.

Mahnken, R., Stein, E., 1996. Parameter identification for viscoplastic models based on analytical derivatives of a least squares functional and stability investigations. International Journal of Plasticity 12 (4), 451–479.

Megnis, M., Varna, J., 2003. Micromechanics based modeling of nonlinear viscoplastic response of unidirectional composite. Composites Science and Technology 63, 19–31.

Moreno, V., Jordan, E.H., 1986. Prediction of material thermomechanical response with a unified viscoplastic constitutive model. International Journal of Plasticity 2 (3), 223–245.

Mukherjee, A., Schmauder, S., Ruhle, M., 1995. Artificial neural networks for the prediction of mechanical behavior of metal matrix composites. Acta Metallurgica Materialia 43 (11), 4083–4091.

Nash, S.G., Sofer, A., 1989. Block truncated-Newton methods for parallel optimization. Mathematical Programming 45, 529–546.

Nash, S.G., Sofer, A., 1990. Assessing a search direction within a truncated Newton method. Operation Research Letters 9, 219–221.

Nash, S.G., Sofer, A., 1996. Linear and Nonlinear Programming. McGraw-Hill, New York.

Perzyna, P., 1966. Fundamentals problems in viscoplasticity. Advances in Applied Mechanics 9, 244–377.

Polak, E., Weiss, G.H., 1972. Computational methods in optimization: a unified approach. Operations Research 20 (2), 456–456.

Qingbin, L., Zhong, J., Mabao, L., Schichum, W., 1996. Acquiring the constitutive relationship for a thermal viscoplastic material using an artificial neural network. Journal of Materials Processing Technology 62, 206–210.

Rubin, M.B., 1996. On the treatment of elastic deformation in finite elastic viscoplastic theory. International Journal of Plasticity 12 (7), 951–965.

Sanchez-Sinencio, E., Lau, C., 1992. Artificial Neural Networks Paradigms, Applications and Hardware Implementation. IEEE Press, New York.

Schapery, R.A., 1968. Stress Analysis of Viscoelastic Composite Materials. Technomic Publishing Co, London, UK.

Schapery, R.A., 1997. Nonlinear viscoelastic and viscoplastic constitutive equations based on thermodynamics. Mechanics of Time Dependent Materials 1 (2), 209–240.

Sun, C.T., Chang, I., 1992. Modeling of elastic–plastic behavior of LDF and continuous fiber reinforced AS-4/PEEK composites. Composites Science and Technology 43, 339–345.

Sun, C.T., Chen, J.L., 1991. A micromechanical model for plastic behavior of fibrous composites. Composites Science and Technology 40, 115–129.

Sun, C.T., Zhu, C., 2000. The effect of deformation-induced-change of fiber orientation on the nonlinear behavior of polymeric composite laminates. Composites Science and Technology 60, 2337–2345.

Suvorova, J.V., 1985. The Influence of Time and Temperature on the Reinforced Plastic Strength. In: Hand Book of Composites, vol. 3. Elsevier Applied Science, England.

Thiruppukuzhi, S.V., Sun, C.T., 1998. Testing and modeling high strain rate behavior of polymeric composites. Composites B 29, 535–546.

Thiruppukuzhi, S.V., Sun, C.T., 2001. Model for strain rate dependent of polymer composites. Composites Science and Technology 61, 1–12.

Werbos, P.J., 1974. Beyond regression: new tools for prediction and analysis in the behavioral sciences. Ph.D. Dissertation, Applied Mathematics, Harvard University, Boston, MA.

Yoon, K.J., Sun, C.T., 1991. Characterization of elastic–viscoplastic properties of an AS4/PEEK thermoplastic composite. Journal of Composite Materials 25, 1277–1296.