Title of paper

Rheological Parameter Estimation for the Prediction of Long-term Deformations in Conventional Tunnelling

Author's Name

Zhenchang Guan [1,*)], Yujing Jiang [2)] and Yoshihiko Tanabashi [2)]

Author's Affiliation

[1)] College of Civil Engineering of Fuzhou University, Fuzhou 350108, Fujian, China
[2)] Department of Civil Engineering, Nagasaki University, Nagasaki 852-8521, Japan

[*]Corresponding author:

College of Civil Engineering of Fuzhou University

No.2 Xueyuan Rd., Fuzhou 350108, China

Email: zcguan@fzu.edu.cn

Rheological Parameter Estimation for the Prediction of Long-term Deformations in Conventional Tunnelling

Abstract: The long-term deformations of mountain tunnels, which attract more and more attentions, are closely related to the time-dependent features of the surrounding rock mass. However, it is not easy to determine an appropriate rheological model and its corresponding parameters for a certain engineering instance. This paper presents a rheological parameter estimation technique by using error backpropagation neural network (BN) and genetic algorithm (GA). The application of the proposed technique to an engineering instance, Ureshino Tunnel Line I on Nagasaki Expressway, is expatiated in detailed. The stochastic nature of the proposed technique is also discussed through case studies. It is proved that the proposed technique can provide the engineer with an optimal estimation of the rheological parameters, which can help the prediction of long-term deformations of mountain tunnels in the future.

Keywords: Rheology, Parameter Estimation, BN, GA.

1 Introduction

The long-term deformations of mountain tunnels, which attract more and more attentions, are closely related to the time-dependent features of the surrounding rock mass. These time-dependent features, named rheology in general, have been studied carefully through a great deal of specified creep, relaxation and quasistatic compression tests (Maranini and Brignol, 1999; Nawrocki *et al*., 1999; Li and Xia, 2000; Shin *et al*., 2005; Fabre and Pellet, 2006). According to the results from laboratory (or in-situ) tests and the experience from engineering practice, many rheological models have been proposed to account for the time-dependent features of rock mass from manifolds (Hudson and Harrison, 1997; Jin and Cristescu, 1998; Pellet *et al*., 2005; Shao *et al*., 2006; Guan *et al*., 2008). For a certain engineering instance, however, it is not easy to determine an appropriate rheological model and its corresponding rheological parameters due to the high cost and the long duration of rheological tests.

On the other hand, parameter estimation technique by using artificial neural network and genetic algorithm developed rapidly during last decade in the field of rock mechanics and rock engineering. It has been applied successfully to generalize the basic properties of rock mass from laboratory experiment data (Meulenkamp and Grima, 1999; Sonmez *et al*., 2006), to determine the model parameters from numerical simulations and monitoring data (Feng *et al*., 2000; Pichler *et al*., 2003), to help the design and maintenance of tunnels, embankments, slopes *etc*. (Cai *et al*., 1998; Mayoraz and Vulliet, 2002; Pichler *et al*., 2003; Rangel *et al*., 2005). The attractiveness of artificial neural network comes from its remarkable information processing capability pertinent to nonlinearity, high parallelism, fault and noise tolerance, self-learning and generalization. Therefore, it is feasible to explore the artificial neural network to estimate the rheological parameters that are commonly difficult to determine, according to in-situ

monitoring data when they are available. This paper represents this kind of application with regard to an engineering instance, the Ureshino Tunnel Line I on Nagasaki Expressway.

Section 2 presents an introduction to the Ureshino Tunnel Line I on Nagasaki Expressway, including the in-situ conditions, the monitoring data *etc*. Section 3 presents the philosophy of the proposed rheological parameter estimation technique. The basic concepts of the multilayer error backpropagation neural network (abbreviated as BN hereafter) and the genetic algorithm (abbreviated as GA hereafter) are also introduced briefly in this section. Section 4 presents the application details of the proposed technique to that engineering instance and some discussions on the stochastic nature of the proposed technique.

2 The Outline of Ureshino Tunnel Line I

The Ureshino Tunnel Line I on Nagasaki Expressway, which had a total length of 683 m, was constructed from May 1990 and completed until Nov. 1992 (JPHC, 2000). The geological profile of the tunnel is schematically illustrated in Fig. 1. The cross section of STA 211+90, where the cover was about 250 m, is mainly concerned in this paper. The cross section dimensions and the convergence monitoring positions (mainly including the convergence at the crown u_c, at the springline u_s and at the invert u_i) are schematically illustrated in Fig. 2. In the longitudinal direction as illustrated in Fig. 3, the span of each excavation cycle was 1.0 m; rock bolts and shotcrete lining were installed immediately after each excavation cycle completed; then second lining was cast in place after two or three months when tunnel face advanced away.

However, the tunnel experienced continuous convergence during its first five years' service life (JPHC, 2000). Taking the section of STA 211+90 for example, the delayed deformations from Nov. 1992 when the tunnel Line I was completed to Nov. 1997 when the tunnel Line II commenced working are depicted in Fig. 4. They are measured by the total station; u_c and u_i are the absolute crown settlement and invert upheaval leveled from a datum mark outside the tunnel; u_s is half of the relative convergence between two wall sides at the springline. For convenience, u_c, u_s and u_i at 12^{th}, 24^{th}, 36^{th}, 48^{th} month are defined as key convergences $\boldsymbol{u}_{key}$ here, on which more concerns are focused throughout this paper.

To avoid further deformations of this tunnel, in fact, ground reinforcement on Ureshino Tunnel Line I (including the backfill grouting beneath the invert and the additional bolting at the foot) was performed after the Ureshino Tunnel Line II was completed in 2000. From a long-term perspective, however, the long-term deformation mechanism for this mountain tunnel should be clarified. Through numerical simulations, Guan *et al*. (2008) proved that these delayed deformations are closely related to the time-dependent features of the surrounding rock mass, and the Burger-MC deterioration rheological model (see details in Appendix A) is more suitable than others to account for these delayed deformations that occurred during the tunnel's service life. Because the deformation characteristics of the Burger-MC deterioration rheological model agree more qualitatively with those in the monitoring data, where the

developments of u_c, u_s and u_i are characterized by the exponential, linear and stair-typed increasing statuses respectively.

However, another problem arises that how to estimate the rheological parameters involved in this model more realistically and quantitatively. Therefore, a rheological parameter estimation technique by using BN and GA jointly is proposed in this paper, to estimate the six rheological parameters (G^K, η^K, η^M, ω_c, ω_ϕ, R_{thr}) involved in this model in a more realistic way.

3 Rheological Parameter Estimation Technique

The basic concepts about the multilayer error backpropagation neural network (BN) and the genetic algorithm (GA), which are two main tools used in the proposed technique, are first introduced. It is followed by a brief and literal introduction to the proposed rheological parameter estimation technique. The application and implement details of the proposed technique are expatiated in section 4.

3.1 Multilayer error backpropagation neural network

There are many types of artificial neural network according to the different network architecture and different learning algorithm, among which BN is the most mature and popular one. The basic concepts and learning algorithm about BN are introduced briefly in this section, which are based on Schalkoff (1997) and Haykin (1999).

The neuron is the basic processing element that sums up the input signals by synaptic weights and converts them into the output signal through a transfer function. The artificial neural network is considered as a computing system composed of a number of neurons that are interconnected by a particular topology. Specifically, the BN consists of an input layer corresponding to the independent variables of the problem, an output layer corresponding to the dependent variables of the problem and one (or more) hidden layer(s) according to the nonlinearity of the problem. The architecture of a fully-connected BN is schematically illustrated in Fig. 5. Hereafter, the symbol of BN(N_0, N_1, N_2, N_3) is used to represent this kind of network consisting of an input layer with N_0 input signals, an output layer with N_4 neurons and two hidden layers with N_2 and N_3 hidden neurons in each layer.

Consider a neuron n_k fed by a set of input signals y_j produced by its former (left) layer neurons, the output signal y_k of this neuron is computed by the following.

$$y_k = \chi_k(v_k) = \chi_k\left(\sum_{j=0}^{N_l} w_{kj} y_j\right) \qquad \text{(for a certain neuron } n_k\text{)} \qquad (1)$$

In the above equations, N_l is the total number of input signals fed to neuron n_k, and w_{kj} is the synaptic weight connecting neuron n_j to neuron n_k (w_{k0} is the synaptic weight connecting an fixed unit bias input signal to neuron n_k). v_k is the weighted sum of all the input signals. χ_k is the transfer function of neuron n_k

that transfers the weighted sum to the output signal y_k. It may be a threshold, sigmoid or linear transfer function depending on the problem.

When neuron n_k is an output neuron, comparing its output signal to the desired output d_k, the error signal of this output neuron n_k is defined as:

$$e_k = d_k - y_k \quad \text{(for an output neuron } n_k\text{)} \tag{2}$$

For a pair of input signals to desired output signals (named a dataset hereafter), the sum-square-error for this dataset is defined as Eq. (3-1). Similarly, the average sum-square-error for the entire datasets is defined as Eq. (3-2).

$$sse = \frac{1}{2}\sum_{N_l} e_k^2 \tag{3-1}$$

$$sse_{av} = \frac{1}{2N_T}\sum_{N_T}\sum_{N_l} e_k^2 \tag{3-2}$$

where N_l is the total number of output neurons; N_T is the total number of datasets.

The synaptic weights are adjusted iteratively to minimize the error function (*sse* or sse_{av}) by the means of error backpropagation learning algorithm. Therefore, there are two sweeps in each iteration: forward activation to map the input signals into the output ones, and error backpropagation to adjust the synaptic weights to minimize the error function. There are many algorithms to adjust the synaptic weights, among which the gradient descent method is the most straightforward one.

For a neuron n_k being an output neuron, the weights correction in the n^{th} iteration, $\Delta w_{kj}(n)$, are defined by the delta rule.

$$\begin{aligned}\Delta w_{kj}(n) &= -\eta\frac{\partial sse}{\partial w_{kj}} + \mu\Delta w_{kj}(n-1) = -\eta\frac{\partial sse}{\partial v_k}\frac{\partial v_k}{\partial w_{kj}} + \mu\Delta w_{kj}(n-1) \\ &= \eta\delta_k y_j + \mu\Delta w_{kj}(n-1) \qquad \text{(for an output neuron } n_k\text{)}\end{aligned} \tag{4-1}$$

$$\text{with } \delta_k = -\frac{\partial sse}{\partial v_k} = -\frac{\partial sse}{\partial y_k}\frac{\partial y_k}{\partial v_k} = -\frac{\partial sse}{\partial e_k}\frac{\partial e_k}{\partial y_k}\frac{\partial y_k}{\partial v_k} = e_k\chi_k' \tag{4-2}$$

In the above equations, η is the learning rate and μ is the momentum coefficient. δ_k is the local gradient of the output neuron n_k and χ_k' is the first order differential of the transfer function with respect to v_k.

For a neuron n_j being a hidden neuron, the weights correction in the n^{th} iteration, $\Delta w_{ji}(n)$, are similarly defined by the delta rule.

$$\begin{aligned}\Delta w_{ji}(n) &= -\eta\frac{\partial sse}{\partial w_{ji}} + \mu\Delta w_{ji}(n-1) = -\eta\frac{\partial sse}{\partial v_j}\frac{\partial v_j}{\partial w_{ji}} + \mu\Delta w_{ji}(n-1) \\ &= \eta\delta_j y_i + \mu\Delta w_{ji}(n-1) \qquad \text{(for an hidden neuron } n_j\text{)}\end{aligned} \tag{5-1}$$

$$\text{with } \delta_j = -\frac{\partial sse}{\partial v_j} = -\frac{\partial sse}{\partial y_j}\frac{\partial y_j}{\partial v_j} = -\left(\sum_k e_k \frac{\partial e_k}{\partial v_k}\frac{\partial v_k}{\partial y_j}\right)\chi'_j = \chi'_j \sum_k \delta_k w_{kj} \qquad (5\text{-}2)$$

where δ_j is the local gradient of the hidden neuron n_j and χ_j' is the first order differential of the transfer function with respect to v_j.

The training mode presented above, where the synaptic weights are updated immediately after the presentation of one training dataset, is called the sequential mode. In contrast, the batch mode uses sse_{av} as the error function, and updates the synaptic weights in an averaged means after the entire training datasets has been presented. In general, the sequential mode needs a smaller local storage and has a better stochastic search ability to prevent entrapment in local minima. The batch mode has a better estimation of the error gradient and a more representative measurement of the required weight correction.

A high learning rate will accelerate training process by changing the synaptic weights significantly with a large step, but increase the risk of overshooting a near-optimal solution or oscillating on the error surface. A small learning rate drives the search steadily in the direction of the global minimum, though slowly. Similarly, a high momentum coefficient will reduce the risk of being stuck in local minima, but increase the risk of overshooting the solution as a high learning rate does. Commonly, the adaptive technique of varying these two parameters in relation to the error gradient information is a better solution for this issue (Schalkoff, 1997; Haykin, 1999).

3.2 Genetic algorithm

The genetic algorithm imitates the evolution of a generation of individuals following Darwin's principle of the survival of the fittest. The basic idea is introduced in the following, which is based on Goldberg (1989).

The basic idea is to maintain a generation of individuals (representing candidate solutions to a given problem) that evolves over time through a process of natural competition and controlled variation. As schematically illustrated in Fig. 6, it starts with a generation of randomly generated individuals, and advances towards better generations by applying the following three genetic operations iteratively.

Reproduction: Firstly, the fitness of each individual to the given problem is examined by a well-devised fitness function. Then individuals are selected into a so-call mating pool probabilistically. The probability to be selected is higher for the individuals characterized by an above-average fitness, so that they may appear two or more times in mating pool. In contrast, the individuals characterized by a below-average fitness may not be selected and then die out. Reproduction is stopped when the number of individuals in the mating pool is equal to the number of individuals in a generation.

Crossover and mutation: Individuals in mating pool are arranged randomly in couples, so-called parents. Each couple is designated to yield two offspring from randomly exchanging one of their parental parameter. During the crossover operation, the parameters in the offspring can be randomly mutated (i.e.

modified) with a very small probability. This mutation operation is used to protect the generations from losing some potentially useful genetic material.

After genetic operations, the individuals in the next generation are generally characterized by an increased average fitness. This iteration is terminated after several generations when some conditions or criteria are reached (e.g. the improvement on the average fitness converges). Then the individual that appears most frequently in the latest generation is regarded as a solution to the given problem.

3.3 Philosophy of the proposed technique

The proposed technique consists of four steps, as schematically illustrated in Fig. 7. In step one, a numerical model (codes: FLAC^{3D}) is set up to simulate the tunnel excavating process according to the in-situ conditions. Then the aforementioned numerical model proceeds to rheological calculation when a set of rheological parameter $\boldsymbol{p}^{FL}$ is applied to the surrounding rock mass. The key convergences computed from the numerical simulations are recorded as $\boldsymbol{u}_{key}^{FL}$. This kind of input-output pair provided by numerical simulations is called one dataset. Similarly, a certain number (fifty for example) of datasets can be obtained, by applying fifty sets of randomly generated rheological parameter to the same numerical simulations.

In step two, these fifty datasets are scaled into their dimensionless (within 0 and 1) counterparts $\boldsymbol{p}^{BN}$ and $\boldsymbol{u}_{key}^{BN}$, and employed to train the BN. Feeding a set of scaled rheological parameter $\boldsymbol{p}^{BN}$ to the BN, and comparing the output $\boldsymbol{o}^{BN}$ with the scaled key convergences $\boldsymbol{u}_{key}^{BN}$, the synaptic weights of BN are adjusted via error backpropagation learning algorithm. After a certain number of training iterations, the post-trained network is expected to approximate the aforementioned numerical simulations in the sense of statistics.

In step three, the post-trained network is employed to find out an estimation for the rheological parameter set via genetic algorithm. Firstly, the key convergences measured in site $\boldsymbol{u}_{key}^{MO}$ should also be scaled into its dimensionless counterpart $\boldsymbol{u}_{key}^{GA}$, which serves as the desired output for the post-trained network aforementioned. An individual $\boldsymbol{p}^{GA}$ represents a set of rheological parameter in this context. Then a generation of randomly initiated individuals (three hundred individuals for example) is fed to the post-trained network, and the fitness of each individual is defined as the sum-square-error between the network output $\boldsymbol{o}^{GA}$ and the desired output $\boldsymbol{u}_{key}^{GA}$. Then this generation of individuals undergoes evolution via the genetic operations. This evolution is terminated after several generations when some conditions or criteria are reached, and the individual appearing most frequently in the latest generation is regarded as an estimation for the rheological parameter set. All these works presented in step two and step three are implemented in Matlab (Mathworks, 2006).

In step four, the rheological parameter set estimated by BN and GA should be reversely scaled (from the dimensionless numbers to their real values) and then revalidated by the identical numerical simulations aforementioned. This estimation is considered a reliable one, if the key convergences

computed from numerical simulations agree with those measured in-situ. The implement details of the proposed technique to the Ureshino Tunnel Line I and some discussions on the stochastic nature of the proposed technique are expatiated in section 4.

4 Implementation details and Discussions

4.1 Dataset preparation

The training datasets are provided by numerical simulations (codes: FLAC^{3D}). According to the geological conditions and tunnel dimensions presented in section 2, a full 3D model including thirty excavation cycles from STA 212+00 to STA 211+70 in longitudinal direction (y axis) is set up firstly. The surrounding rock mass is basically hornblende andesite lava with a cover about 250 m, whereas the simulation regions in the cross section are only five times and three times of tunnel diameter in vertical direction (z axis) and horizontal direction (x axis). The surrounding rock mass is assumed as a Mohr-Coulomb material at this stage, with its mechanical properties listed in Table 1 (JPHC, 2000).

Then according to the construction cycles presented in section 2, the tunnel face advances and the rock bolts and shotcrete lining are modeled gradually. The second lining is modeled after all these thirty exaction cycles are completed. The mechanical properties of shotcrete lining and the second lining are listed in Table 2 (JPHC, 2000).

After the construction process completed, the Burger-MC deterioration rheological model is applied to the surrounding rock mass, and the aforementioned numerical simulations proceed to rheological calculation focusing on the first five years during the tunnel's service life. The numerical model consists of 35898 gridponits, 33450 zones and 2100 structure elements. The computing time for the rheological calculation is nearly 10 hours per case, under the configuration of AMD Athlon $^{(tm)}$ 3800+ and 1.0 GB memory.

The rheological parameters applied to the numerical simulations are initiated in such a random way. Fifty sets of random number (within 0 and 1) are initiated first, so that the rheological parameters applied to the numerical simulations can be scaled from them by the following equation.

$$\boldsymbol{p}^{FL} = \boldsymbol{p}^{lower} + (\boldsymbol{p}^{upper} - \boldsymbol{p}^{lower}) \times \boldsymbol{p}^{BN} \tag{6}$$

In the above equation, $\boldsymbol{p}^{BN}$ is a 6×1 vector recording the random numbers within 0 and 1 that are fed to the BN as the input signals; $\boldsymbol{p}^{FL}$ is the counterpart vector recording the six rheological parameters that are applied to the numerical simulations. $\boldsymbol{p}^{lower}$ and $\boldsymbol{p}^{upper}$ are two 6×1 vectors recording the lower and upper limits for the six rheological parameters, which can take reference to the literatures (Hudson and Harrison, 1997; Li and Xia, 2000; Shin *et al*., 2005; Fabre and Pellet, 2006). The informations for these fifty cases are partly listed in Table 3.

Then the twelve key convergences (i.e. u_c, u_s and u_i at 12th, 24th, 36th, 48th month respectively)

computed from the numerical simulations are recorded and then scaled into dimensionless number within 0 and 1 by the following equation.

$$\boldsymbol{u}_{key}^{BN} = \frac{(\boldsymbol{u}_{key}^{FL} - \boldsymbol{u}_{key}^{\min})}{(\boldsymbol{u}_{key}^{\max} - \boldsymbol{u}_{key}^{\min})} \tag{7}$$

In the above equation, $\boldsymbol{u}_{key}^{FL}$ is a 12×1 vector recording the key convergences that are computed from the numerical simulations; and $\boldsymbol{u}_{key}^{BN}$ is its counterpart vector recording the scaled key convergences that serve as the desired outputs for BN training. $\boldsymbol{u}_{key}^{min}$ and $\boldsymbol{u}_{key}^{max}$ are two 12×1 vectors recording the minimum and maximum key convergences amongst these fifty cases of numerical simulations.

The normalization for $\boldsymbol{p}^{BN}$ and $\boldsymbol{u}_{key}^{BN}$ is very important for the training process due to these two reasons: it can prevent the large input (output) components from overriding the small ones; and the sigmoid transfer function that is commonly used in the hidden neuron is sensitive to the value within -1 and 1.

4.2 Network training

4.2.1 Network architecture

It is no doubt that the input layer includes six input signals corresponding to the six rheological parameters under consideration, and the output layer includes twelve output neurons corresponding to the twelve key convergences. Although the output layer, theoretically, can include as many output neurons as the monitoring data available, increasing the number of output neurons excessively is neither economic nor accurate due to the following reason (Schalkoff, 1997; Haykin, 1999). The number of datasets required to train a BN well (i.e. having a good generalization ability) is closely related to the number of synaptic weights involved in the network. When the training datasets are very limited, increasing the number of output neurons would increase the number of synaptic weights greatly and consequently increase the risk of overfitting significantly.

The transfer function employed in the output neurons is linear function, as described in the following.

$$\chi(v) = v \tag{8-1}$$

Determining the numbers of hidden layers and hidden neurons are more complicated. Generally speaking, the number of hidden layers and hidden neurons are highly dependent on the nonlinearity and multimodality of the problem under consideration. More hidden layers and neurons are able to detect more hidden features in the training datasets, but increase the risk of overfitting at the same time (Schalkoff, 1997; Haykin, 1999). However, one knows nothing about the relationship between the network architecture and the problem under consideration prior to some attempts. Therefore, four types of network architecture with different hidden neurons and hidden layers are tried, with more detail discussions expatiated in section 4.5.

The transfer function employed in the hidden neurons is log-sigmoid function, as described in the following.

$$\chi(v) = \frac{1}{1+\exp(-v)} \tag{8-2}$$

4.2.2 Leave-n-out corotation strategy

The available datasets should be divided into two subsets (training subset and validating subset) by a certain proportion. Generally, the learning curve (i.e. error function plot) for the training subset decreases monotonically with the increasing of training epoch. In contrast, the learning curve for the validating subset decreases monotonically to minimum and then starts to increase as the training epoch continues, as schematically illustrated in Fig. 8 (Schalkoff, 1997; Haykin, 1999). This heuristic suggests that the point of minimum on the validation learning curve can be used as a stopping criterion to prevent excessive training.

However, the subset partitioning is an intractable issue, since the available datasets are very limited. Herein, a leave-n-out corotational strategy is employed to deal with this issue (Basheer and Hajmeer, 2000; Abraham, 2004). Ten datasets are randomly selected for validation and the other forty datasets for training. Then train the network until the stopping criterion is met. Repeat this iteration and train the network continuously, until each dataset appeared at least once in both validating subset and training subset.

4.2.3 Random initial weights

The initialization of synaptic weights has an effect not only on the network convergence speed, but also on the network's generalization ability. Herein, a random initial weights strategy is employed to deal with this issue (Basheer and Hajmeer, 2000; Abraham, 2004). The synaptic weights are initiated randomly within the intervals of $(-3N_l^{1/2}, 3N_l^{1/2})$, where N_l is the number of neurons in l^{th} layer. The network is then trained by the leave-n-out corotation process as presented above. After that, all the fifty datasets are fed to the network entirely and the network performance (i.e. the error function sse_{ave}) is recorded.

This kind of iteration of random initializing and corotative training is repeated dozens of times, and the network with best performance (i.e. minimum sse_{ave}) is considered as a post-trained network that can approximate the numerical simulations aforementioned in the sense of statistics.

4.3 Genetic searching

The aforementioned post-trained network together with genetic algorithm is employed to estimate the six rheological parameters according to the monitoring data. The key convergences measured in site $\boldsymbol{u}_{key}^{MO}$ should be scaled into its dimensionless counterpart $\boldsymbol{u}_{key}^{GA}$ by the following equation, and serve as the desired output for the post-trained network aforementioned.

$$\boldsymbol{u}_{key}^{GA} = \frac{(\boldsymbol{u}_{key}^{MO} - \boldsymbol{u}_{key}^{\min})}{(\boldsymbol{u}_{key}^{\max} - \boldsymbol{u}_{key}^{\min})} \tag{9}$$

Three hundred individuals $\boldsymbol{p}^{GA}$ (each individual represents a set of scaled rheological parameters within 0 and 1) are initiated randomly and fed to the post-trained network. Comparing the network output $\boldsymbol{o}^{GA}$ with the desired output $\boldsymbol{u}_{key}^{GA}$, the error function *sse* is defined as the fitness for each individual. Then a reproduction indicator ω is devised to select these individuals into the mating pool probabilistically, as formulated in the following.

$$\omega(sse) = (\frac{sse - sse_{av}}{sse_{av}}) + gauss(0, \alpha) \tag{10}$$

In the above equation, sse_{av} is the average error function for the whole generation, $gauss(0, \alpha)$ is a Gaussian distributed random number generator with a mean of zero and a variance of α. An individual is selected into the mating pool when its reproduction indicator is greater than zero, or is prohibited on the inverse condition.

Individuals in mating pool are arranged randomly in couples, and yield their offspring from randomly exchanging one of their parental parameter. With a small probability, the parameters in their offspring can be randomly mutated (i.e. modified). Some other random number generators are also employed in the crossover and mutation processes (although not explained in details here), to ensure the generation undergoes evolution in a natural form.

The evolution is terminated after 8~10 generations, when the improvement on the average fitness converges. Then the individual that appears most frequently in the latest generation is regarded as a solution to the given problem.

4.4 Revalidation by numerical simulations

The rheological parameter estimation technique presented above is stochastic in nature, therefore, the estimation provided by the BN and GA should be revalidated again by the identical numerical simulations. Trying the estimation into the identical numerical simulations and comparing the results to the monitoring data, the error of crown convergence between them, e_{uc}, is defined as:

$$e_{uc} = \frac{\left\| \boldsymbol{u}_c^{FL} - \boldsymbol{u}_c^{MO} \right\|}{\left\| \boldsymbol{u}_c^{MO} \right\|} \tag{11}$$

where $\boldsymbol{u}_c^{MO}$ is the crown convergence measured in-situ, and $\boldsymbol{u}_c^{FL}$ is the counterpart that is computed from numerical simulations. $\|\cdot\|$ is the denotation of vector norm calculation. The errors of springline convergence and invert convergence, e_{us} and e_{ui}, can also be defined by the similar means.

For example, a network with the architecture of BN(6,9,12) (i.e. six input neurons in the input layer, nine hidden neurons in a hidden layer and twelve output neurons in the output layer) is trained by the

strategy and the datasets presented above, then the post-trained network is employed to search for a parameter set estimation via GA. This estimation is revalidated by the identical numerical simulations again, with the results depicted in Fig. 9. The results from numerical simulations agree well with the monitoring data, with the three errors being 4.0%, 12.1% and 9.3%, respectively. It implies that this estimation for the rheological parameter set is rather reliable.

However, it is expected that there exist numerous estimations due to the stochastic nature of the genetic operations and the training process, and there is no guarantee that all these estimations are always reliable. The section 4.5 presents more discussions on this issue.

4.5 Discussions

4.5.1 Stochastic nature

The rheological parameter estimation technique presented above is stochastic in nature. Some repetitive trials are performed to examine the randomness of the proposed technique. For a certain network architecture, the same training process is repeated five times to yield five post-trained networks (named T1~T5). For a certain post-trained network, the same genetic searching process is repeated five times to provide five estimations (named A1~A5).

In the genetic searching process, the individual initialization and the genetic operations are random. However, the randomness is relatively small, when the number of individuals in each generation is large enough. Taking the post-trained network BN(6,9,12)_T3 for example, five parameter set estimations provided by the genetic searching are listed in Table 4. The difference between these five estimations is relatively small, and the medium one between them is selected as the possible estimation for the post-trained network BN(6,9,12)_T3. The similar tendency can be observed for other post-trained networks. Therefore, it can be concluded that the possible estimation provided by the genetic searching is generally determinate for a particular post-trained network, when the number of individuals in each generation is large enough.

On the other hand, the randomness in network training is comparatively large, especially when the available datasets are limited. The BN is expected to approximate the numerical simulations in the sense of statistics, after it has been well trained by the datasets provided by the numerical simulations. Therefore, the number of datasets required to train a BN well is closely related to the number of synaptic weights and the desired goal of error function. It is expected to be a very large number (Schalkoff, 1997; Haykin 1999). Therefore, merely fifty datasets are insufficient on this occasion. Taking another post-trained network BN(6,9,12)_T2 for example, the estimation provided by this post-trained network is proved a bad one when revalidated by the numerical simulations. The simulation results are depicted in Fig. 10, with the three errors being 8.6%, 14.3% and 34.1% respectively. Table 5 summarizes five possible estimations provided by those post-trained networks (T1~T5) sharing the same architecture of BN(6,9,12). These five estimations are quite different from each other, which implies that the

randomness in training process is comparatively large. Herein, the estimation with the minimum errors is selected as the optimal one for the network architecture of BN(6,9,12). Theoretically, increasing the number of datasets can reduce the randomness in training process and increase the accuracy of estimation. In practice however, trying and validating seems to be a more realistic solution for this issue.

4.5.2 Hidden neurons and hidden layers

To discuss the influence of hidden neurons and hidden layers on network's learning and generalization ability, another three different network architectures are tried by the similar means presented above. Taking the architecture of BN(6,6,12,12) for example, Table 6 summarizes five possible estimations provided by those post-trained networks sharing the same architecture. After revalidating, the possible estimation provided by the network T1 has a minimum average error and is selected as the optimal estimation for BN(6,6,12,12), The revalidating simulation results are depicted in Fig. 11, with the three errors being 8.1%, 13.2% and 6.0% respectively.

Table 7 summarizes the four optimal estimations for different network architectures. In fact, all these four estimations are rather good, since the average errors of them are below 10%. However, one cannot tell easily that which architecture is more proper for this problem, because the estimation is stochastic in nature and no statistical characteristic can be found through such few attempts. On the precondition that the errors are acceptable (for example, the average error is below 10%), it is the author's opinion that the networks with less hidden neurons are preferred.

4.5.3 Characteristic of the long-term deformations

As far as Ureshino Tunnel Line I is concerned, twenty possible estimations (five estimations for each four network architectures) are provided by the proposed technique presented above. Based on the revalidating results (although not totally listed in this paper), it can be concluded that the strength deterioration parameters (ω_c, ω_ϕ and R_{thr}) would influence u_i more significantly, and the viscoelastic parameters (G^K, η^K and η^M) would influence both u_c and u_i to some extend. However, it is difficult (or impossible) to find out an estimation whose simulation results could agree well with the monitoring data of u_c, u_i and u_s simultaneously, since there are no numerical simulations that can take a comprehensive account of the in-situ conditions and the rock mass behaviors. Therefore, it is up to the engineer to select a best estimation according to his/her judgment. For example, the 1st estimation in Table 7 is preferred, since it has a minimum average error.

In fact, ground reinforcement on Ureshino Tunnel Line I was conducted after the Ureshino Tunnel Line II was completed in 2000. The reinforcement mainly included the backfill grouting beneath the invert and the additional bolting at the foot, which proved to have a good effect on controlling the tunnel convergence at the invert. Although the in-site conditions have changed, the rheological parameters estimated by the proposed method are still useful to predict the tunnel's long-term deformations in the

future.

5 Conclusions

The long-term deformations of mountain tunnels, which attract more and more attentions, are closely related to the time-dependent features of the surrounding rock mass. Although many rheological models have been proposed to account for the time-dependent features of rock mass from manifolds, for a certain engineering instance however, it is difficult to estimate the rheological parameters due to the high cost and long duration of rheological tests. Therefore, the rheological parameter estimation technique by using BN and GA is proposed in this paper, and applied to an engineering instance the Ureshino Tunnel Line I on Nagasaki Expressway.

The Burger-MC deterioration rheological model is applied to the surrounding rock mass to account for the mechanism of delayed deformations that occurred during the tunnel's service life. The six rheological parameters involved in this model are estimated via such four steps: 1) numerical simulations are utilized first to generate a certain number of datasets for BN training; 2) the BN is trained with those datasets to approximate the numerical simulations in the sense of statistics; 3) the post-trained network combined with the genetic algorithm is employed to provide an optimal estimation for the rheological parameter set according to the in-situ monitoring data; 4) this optimal estimation should be revalidated by the identical numerical simulations aforementioned.

The stochastic nature of the proposed technique is discussed. The randomness in the genetic searching is relatively small, when the number of individuals in each generation is large enough. In contrast, the randomness of in network training is comparatively large, because the available datasets are insufficient comparing with the required dataset to train the network well. Therefore, it is suggested that the network with less hidden neurons are preferred, on the precondition that the errors between the revalidating simulation results and monitoring data are acceptable.

The characteristic of long-term deformations for the Burger-MC deterioration rheological model is also summarized in the paper. It is difficult (or impossible) to find out an estimation whose simulation results could agree well with all the monitoring data simultaneously. Therefore, it is up to the engineer to select a best estimation from those ones provided by the BN and GA, according to his/her judgment.

Appendix A: Burger-MC deterioration rheological model

The constitutive laws of Burger-MC deterioration rheological model are introduced briefly in this appendix, which is based on Guan *et al*. (2008). The deviatoric behavior can be schematically illustrated in Fig. A1, where a Kelvin unit, a Maxwell unit and a MC plastic unit are connected in series and

subjected to some certain deviatoric stresses jointly. The constitutive laws for the Kelvin and Maxwell units are formulated by Eq. (A1) and Eq. (A2).

$$s_{ij} = 2\eta^K \dot{e}_{ij}^K + 2G^K e_{ij}^K \tag{A1}$$

$$\dot{e}_{ij}^M = \frac{\dot{s}_{ij}}{2G^M} + \frac{s_{ij}}{2\eta^M} \tag{A2}$$

For the plastic unit, a Mohr-Coulomb failure criterion f is first defined as Eq. (A3). After the failure criterion being reached, the deviatoric behavior of the plastic unit is specified by flow rule and plastic potential g, as shown in Eq. (A4) and Eq. (A5).

$$f = \sigma_1 - \sigma_1 \frac{1+\sin\phi}{1-\sin\phi} + 2c\sqrt{\frac{1+\sin\phi}{1-\sin\phi}} \tag{A3}$$

$$\dot{e}_{ij}^P = \lambda \frac{\partial g}{\partial \sigma_{ij}} - \frac{\dot{\varepsilon}_{kk}^P}{3}\delta_{ij} \quad \text{with} \quad \dot{\varepsilon}_{kk}^P = \lambda\left[\frac{\partial g}{\partial \sigma_1} + \frac{\partial g}{\partial \sigma_2} + \frac{\partial g}{\partial \sigma_3}\right] \tag{A4}$$

$$g = \sigma_1 - \sigma_3 \frac{1+\sin\psi}{1-\sin\psi} \tag{A5}$$

In the above equations, e_{ij} and s_{ij} are the deviatoric strain and stress tensors; ε_{kk} and σ_{kk} are the volumetric strain and stress. σ_1, σ_2 and σ_3 are the three principle stresses. G^K and η^K are the shear modulus and the viscosity of Kelvin unit. G^M and η^M are the shear modulus and the viscosity of Maxwell unit. c, ϕ and ψ are the cohesion, the friction angle and the dilation angle of the MC plastic unit. λ is a multiplier that can be eliminated in the calculation afterwards. The variables denoted by a dot mark refer to their first-order derivatives with respect to rheological time. The superscripts *K*, *M* and *P* denote the Kelvin, the Maxwell and the MC plastic partitions for corresponding variables.

In addition, the Burger-MC deterioration rheological model assume that the cohesion c and the friction angle ϕ will decrease with rheological time, regardless of whether the loss of strength is caused by cyclic loading fatigue, by clay mineral hydration or by some other reasons. It is assumed that the loss of strength is controlled by its current stress state. Furthermore, there exists a threshold (i.e. R_{thr}) to initiate this kind of strength deterioration and two lower limits (i.e. c_{res} and ϕ_{res}) to circumscribe the strength deterioration.

$$\frac{dc}{dt} = -\omega_c R \qquad (R \geq R_{thr}, \quad c \geq c_{res}) \tag{A6}$$

$$\frac{d\phi}{dt} = -\omega_\phi R \qquad (R \geq R_{thr}, \quad \phi \geq \phi_{res}) \tag{A7}$$

$$R = \frac{\sigma_1 - \sigma_3}{2c\cos\phi + (\sigma_1 + \sigma_3)\sin\phi} \tag{A8}$$

In the above equations, the parameter R is named stress coefficient in this paper and indicates the "distance" from the current stress state to the MC failure envelope. When the stress coefficient is greater than a certain threshold R_{thr}, the rock strength initiates to deteriorate. The multipliers ω_c and ω_ϕ are two deterioration ratios that control the deteriorating of the cohesion and friction angle by some certain rates. c_{res} and ϕ_{res} are residual cohesion and residual friction angle that can be estimated from conventional triaxial tests.

The proposed rheological model can be implemented in the numerical codes FLAC^{3D}, although it is not included directly in the FLAC^{3D} library of constitutive laws. The programming language FISH can help implement the proposed model under the framework of classic Burger-MC rheological model in such an indirect way (Itasca Consulting Group): reevaluate the cohesion and the friction angle for every rock zone according to Eqs. (A6)~(A8), after a certain rheological period being computed.

Commonly, the shear modulus of Maxwell unit G^M is considered the same as the ordinary shear modulus G. Therefore, this model includes six rheological parameters (G^K, η^K, η^M, ω_c, ω_ϕ and R_{thr}) that should be estimated by using BN and GA according to the in-situ monitoring data.

Reference

Abraham A. Meta learning evolutionary artificial neural networks. *Neurocomputing* 2004; 56 (1): 1-38.

Basheer I, Hajmeer M. Artificial neural network: fundamentals, computing, design and application. *J. Microbiological Methods* 2000; 43 (1): 3-31.

Cai J, Zhao J, Hudson J. Computerization of rock engineering system using neural network with expert system. *Rock Mech. Rock Engng.* 1998; 31 (3): 135-152.

Fabre G, Pellet F. Creep and time-dependent damage in argillaceous rocks. *Int. J. Rock Mech. Min. Sci.* 2006; 43 (6): 950-960.

Feng X, Zhang Z, Sheng Q. Estimating mechanical rock mass parameters relating to the Three Gorges Project permanent shiplock using an intelligent displacement back analysis method. *Int. J. Rock Mech. Min. Sci*. 2000; 37 (7): 1039-1054.

Goldberg D. Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, 1989.

Guan Z, Jiang Y, Tanabashi Y, Huang H. A new rheological model and its application in Mountain Tunnelling. *Tunnelling and Underground Space Technology* 2008; 23 (3), 292-299.

Haykin S. Neural networks: A comprehensive foundation. Prentice-Hall Inc., 1999.

Hudson J, Harrison J. Engineering rock mechanics. Pergamon, 1997.

Itasca Consulting Group Inc. FLAC^{3D}, Fast Lagrange Analysis of Continua in 3 Dimensions, Version 2.0, User Manual. Itasca Consulting Group Inc., 1997.

Japan Public Highway Corporation. The construction and monitoring reports of the Nagasaki Expressway (in Japanese). JPHC, 2000.

Jin J, Cristescu N. An elastic viscoplastic model for transient creep of rock salt. *International Journal of Plasticity* 1998; 14

(1): pp. 85-107.

Li Y, Xia C. Time-dependent tests on intact rocks in uniaxial compression. *Int. J. Rock Mech. Min. Sci*. 2000; 37 (3): 467-475.

Maranini E, Brignoli E. Creep behaviour of a week rock: experimental characterization. *Int. J. Rock Mech. Min. Sci.* 1999; 36 (1): 127-138.

Mathworks Inc. Matlab R2006a, User Help Document. Mathworks Inc., 2006.

Mayoraz F, Vulliet L. Neural networks for slope movement prediction. *Int. J. Geomechanics ASCE* 2002; 2 (2), 153-173.

Meulenkamp F, Grima M. Application of neural networks for the prediction of the unconfined compressive strength from Equotip hardness. *Int. J. Rock Mech. Min. Sci*. 1999; 36 (1): 29-39.

Nawrocki P, Cristescu N, Dusseault M, Bratli R. Experimental methods for determining constitutive parameters for nonlinear rock modeling. *Int. J. Rock Mech. Min. Sci*. 1999; 36 (5): 659-672.

Pellet F, Hajdu A, Deleruyelle F, Besnus F. A viscoplastic model including anisotropic damage for the time dependent behaviour of rock. *Int. J. Numer. Anal. Meth. Geomech.* 2005;. 29 (9): 941-970.

Pichler B, Lackner R, Mang H. Back analysis of model parameters in geotechnical engineering by means of soft computing. Int. J. Numer. Meth. Engng. 2003; 57 (14): 1943-1978.

Rangel J, Iturraran-Viveros U, Ayala A, Cervantes F. Tunnel stability analysis during construction using a neuro-fuzzy system. *Int. J. Numer. Anal. Meth. Geomech*. 2005; 29 (15), 1433-1456.

Schalkoff J. Artificial neural network. McGraw-Hill, 1997.

Shao J, Chau K, Feng X. Modeling of anisotropic damage and creep deformation in brittle rocks. *Int. J. Rock Mech. Min. Sci*. 2006; 43 (4): 582-592.

Shin K, Okubob S, Fukuib K, Hashibab K. Variation in strength and creep life of six Japanese rocks. *Int. J. Rock Mech. Min. Sci.* 2005; 42 (2): 251-260.

Sonmez H, Gokceoglu C, Nefeslioglu H, Kayabasi A. Estimation of rock modulus: for intact rocks with an artificial neural network and for rock masses with a new empirical equation. *Int. J. Rock Mech. Min. Sci*. 2006; 43 (2): 224-235.

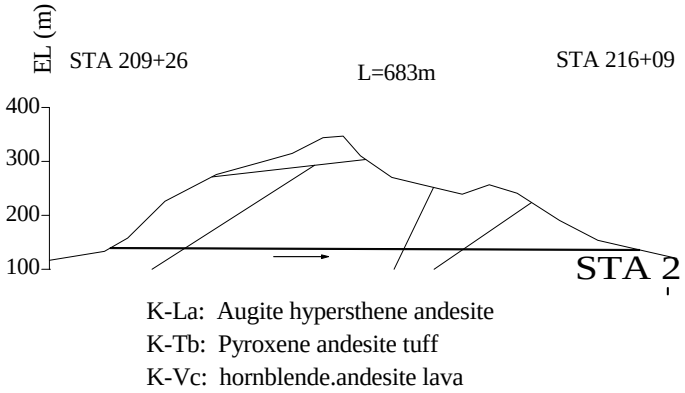


Fig. 1. Schematic representation of the geological profile of Ureshino Tunnel Line I (JPHC, 2000).

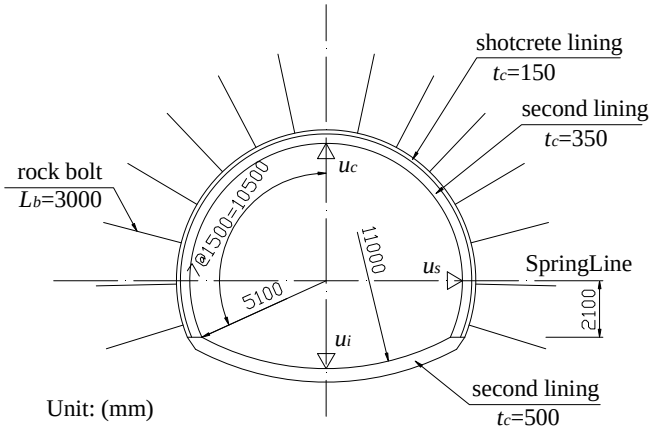


Fig. 2. The cross section dimensions and the convergence monitoring positions (JPHC, 2000).

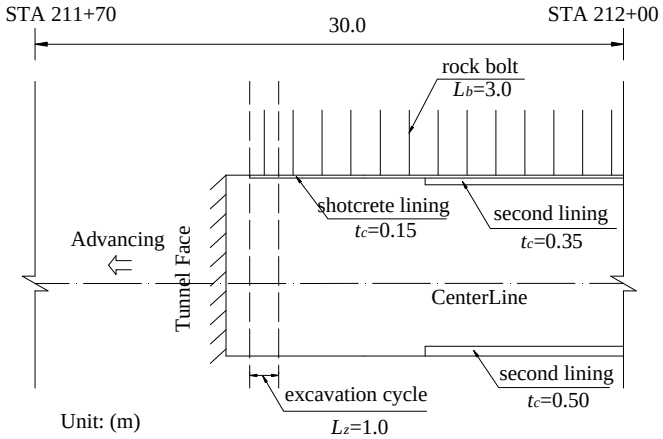


Fig. 3. Schematic representation of the excavation cycling in longitudinal direction (JPHC, 2000).

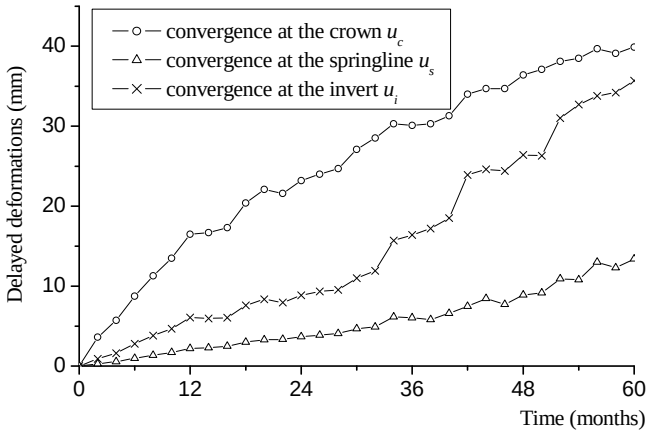


Fig. 4. The delayed deformations of Ureshino Tunnel Line I (STA 211+90) during its first five years of service life (JPHC, 2000).

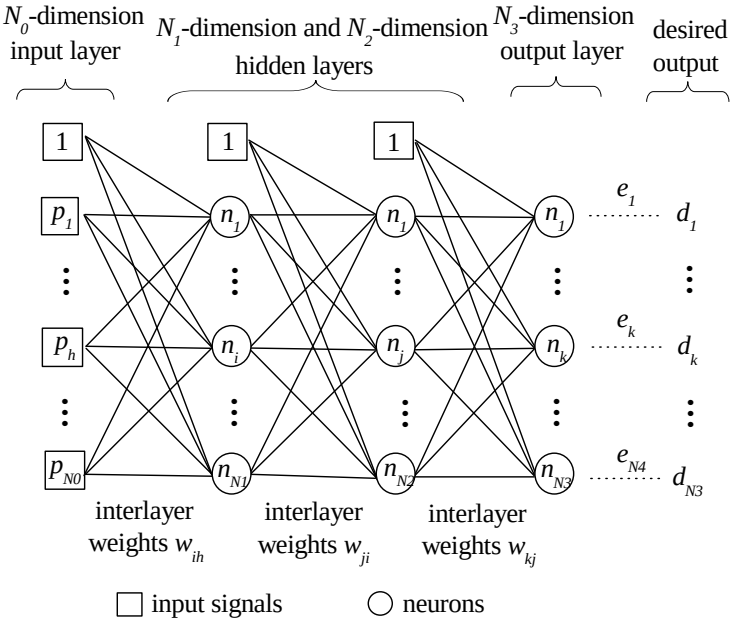


Fig. 5. A fully-connected BN with the architecture of BN(N_0, N_1, N_2, N_3).

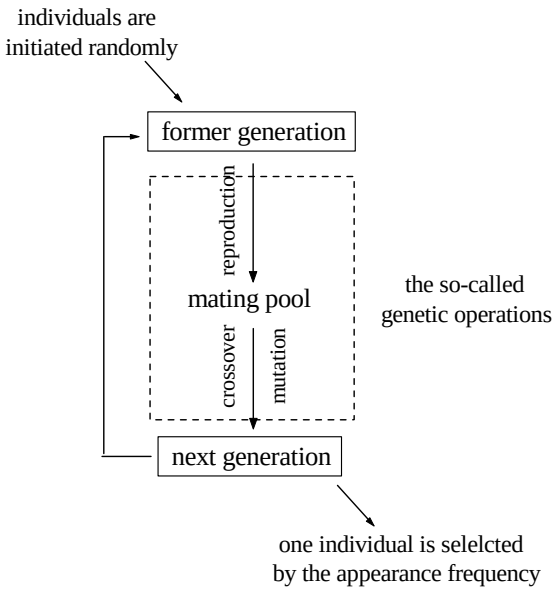


Fig. 6. Schematic illustration for the genetic algorithm.

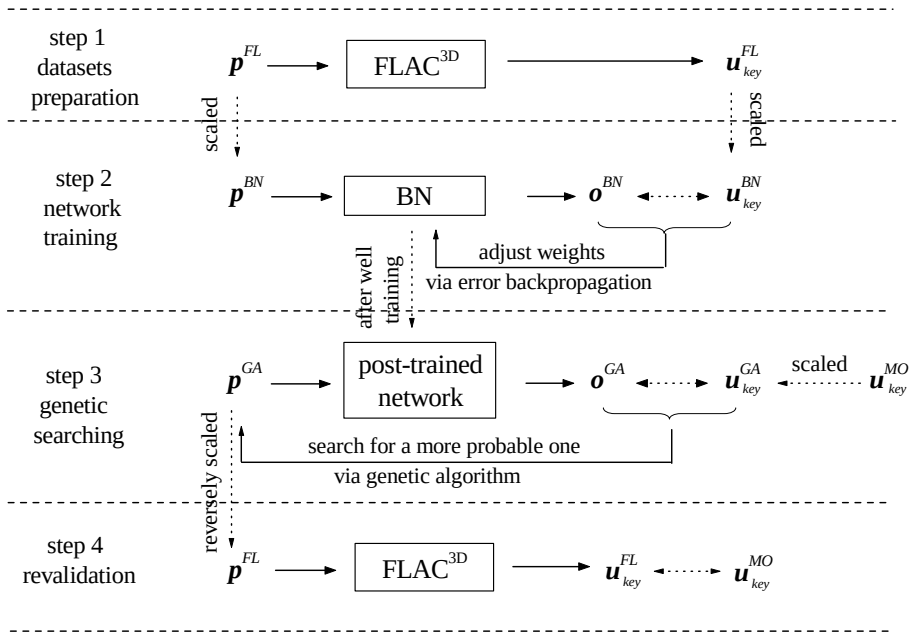


Fig. 7. The philosophy of the proposed rheological parameter estimation technique.

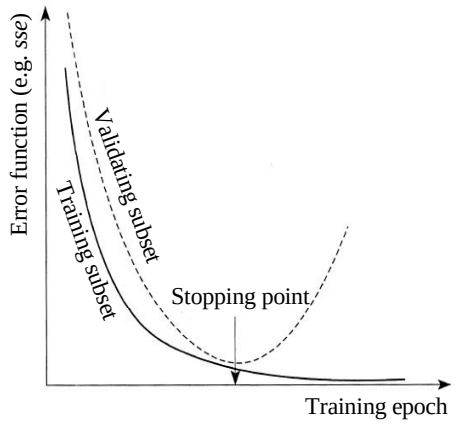
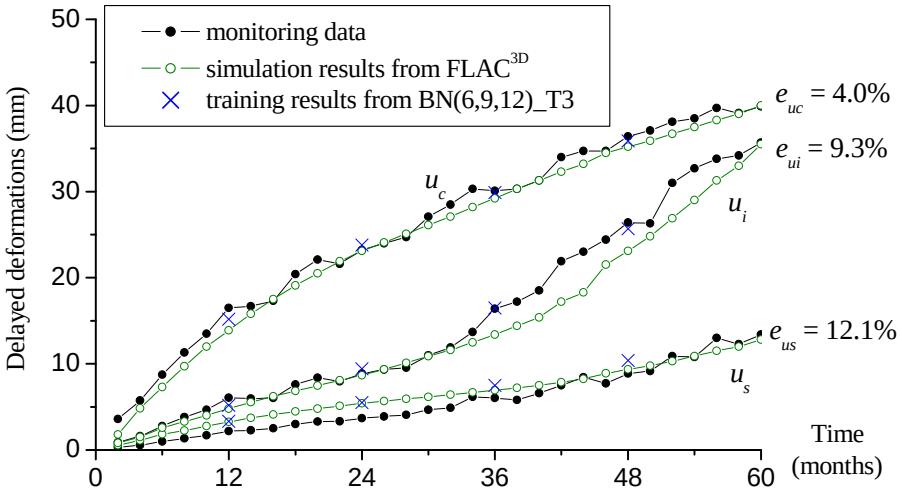
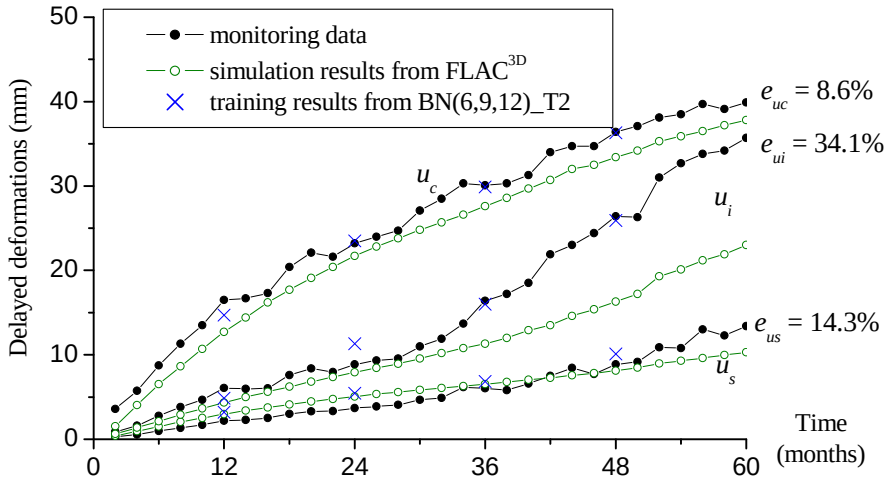


Fig. 8. Schematic illustration of early stopping criterion (after Haykin, 1999).



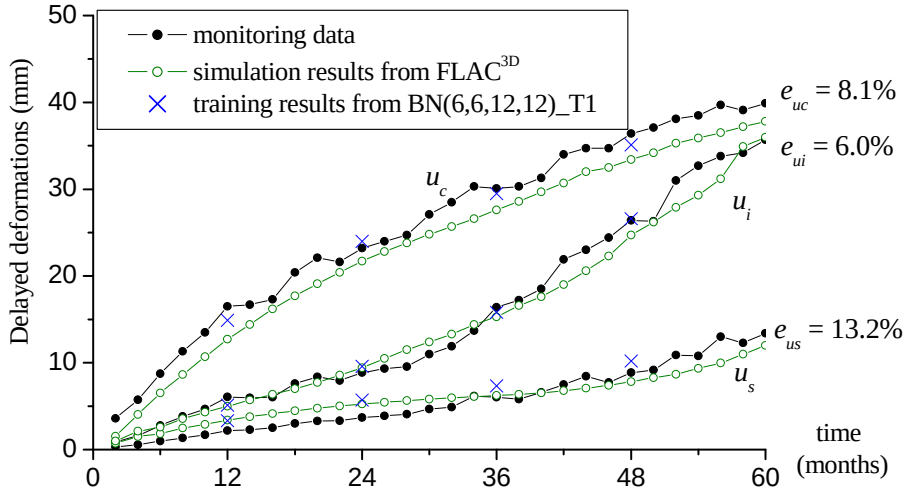
	G^K	η^K	η^M	ω_c	ω_ϕ	R_{thr}
BN((6,9,12)_T3	0.560	0.337	0.314	0.320	0.585	0.363
FLAC^{3D}	6.04E+08	4.04E+16	3.83E+17	3.88E+04	2.84	0.49

Fig. 9. Comparison of the in-situ monitoring data to the numerical simulation results and the network training results (for BN(6,9,12)_T3).



	G^K	η^K	η^M	ω_c	ω_ϕ	R_{thr}
BN(6,9,12)_T2	0.476	0.388	0.583	0.139	0.537	0.642
FLAC^{3D}	5.28E+08	4.49E+16	6.24E+17	2.25E+04	2.65	0.71

Fig. 10. Comparison of the in-situ monitoring data to the numerical simulation results and the network training results (for BN(6,9,12)_T2).



	G^K	η^K	η^M	ω_c	ω_ϕ	R_{thr}
BN(6,6,12,12)_T1	0.652	0.275	0.357	0.453	0.660	0.508
FLAC^{3D}	6.87E+08	3.48E+16	4.21E+17	5.08E+04	3.14	0.61

Fig. 11. Comparison of the in-situ monitoring data to the numerical simulation results and the network training results (for BN(6,6,12,12)_T1).

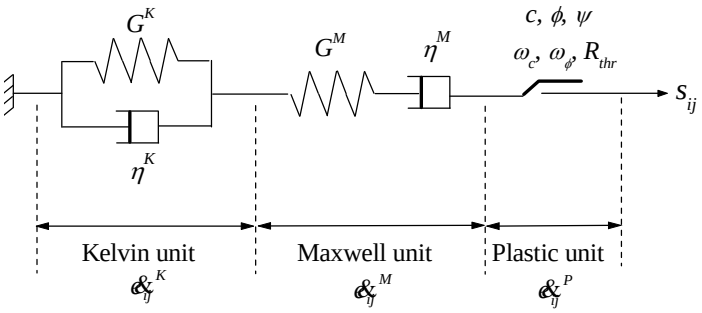


Fig. A1. Schematic representation of the deviatoric behavior of the Burger-MC deterioration rheological model (after Guan *et al*, 2008).

Table 1. Properties of the rock masses employed in numerical simulations (JPHC, 2000).

Properties	Value	Properties	Value
Bulk modulus K (MPa)	833	Density ρ (kg/m^3)	2500
Shear modulus G (MPa)	385	Residual cohesion c_{res} (MPa)	0.310
Cohesion c (MPa)	0.577	Residual friction angle ϕ_{res} (^{o})	25.5
Friction angle ϕ (^{o})	30.0	In-situ σ_{zz} (MPa)	8.0
Dilation angle ψ (^{o})	5.1	In-situ σ_{xx} (MPa)	8.0

Table 2. Properties of the linings employed in numerical simulations (JPHC, 2000).

	shotcrete lining	second lining
Young's Moduls E (MPa)	1.0e4	2.0e4
Poisson ratio μ	0.25	0.25
UCS σ_c (MPa)	15	30
Friction angle ϕ (^{o})	35	40
Dilation angle ψ (^{o})	3.0	3.0
Thickness of lining t_c (m)	0.15	0.35

Table 3. Parameter informations of 50 randomly generated cases.

Case No.	G^K	η^K	η^M	ω_c	ω_ϕ	R_{thr}
01	0.527	0.729	0.468	0.516	0.650	0.650
02	0.791	0.250	0.138	0.461	0.525	0.613
⋮	⋮	⋮	⋮	⋮	⋮	⋮
49	0.412	0.556	0.444	0.111	0.125	0.750
50	0.333	0.667	0.345	0.778	0.550	0.163
	Lower and upper limits for each parameter					
	G^K: 1e8 ~ 1e9 Pa		η^K: 1e16 ~ 1e17 Pa · s		η^M: 1e17 ~ 1e18 Pa · s	
	ω_c: 1e4 ~ 1e5 Pa/year		ω_ϕ: 0.5 ~ 4.5 ^{o}/year		R_{thr}: 0.2 ~ 1.0	

Table 4. Five estimations for the rheological parameter set (for BN(6,9,12)_T3).

Algorithm No.	G^K (Pa)	η^K (Pa · s)	η^M (Pa · s)	ω_c (Pa/y)	ω_ϕ (^{o}/y)	R_{thr}
A1	6.55E+08	3.74E+16	3.59E+17	3.56E+04	2.56	0.42
A2	5.95E+08	3.54E+16	3.71E+17	4.62E+04	2.79	0.43
A3	5.63E+08	4.20E+16	4.12E+17	4.35E+04	2.56	0.44
A4 (✔)	6.04E+08	4.04E+16	3.83E+17	3.88E+04	2.84	0.49
A5	6.35E+08	3.94E+16	4.01E+17	3.96E+04	2.56	0.51

(✔): The medium one is selected as the possible estimation

Table 5. Five possible estimations for the rheological parameter set (for BN(6,9,12)).

Train No.	G^K (Pa)	η^K (Pa · s)	η^M (Pa · s)	ω_c (Pa/y)	ω_ϕ (°/y)	R_{thr}	e_{uc} (%)	e_{us} (%)	e_{ui} (%)
T1	5.96E+08	4.38E+16	2.63E+17	4.69E+04	2.70	0.61	6.9	20.1	11.3
T2	5.28E+08	4.49E+16	6.24E+17	2.25E+04	2.65	0.71	8.6	14.3	34.1
T3 (✔)	6.04E+08	4.04E+16	3.83E+17	3.88E+04	2.84	0.49	4.0	12.1	9.3
T4	5.63E+08	4.20E+16	5.05E+17	4.49E+04	2.46	0.44	6.2	25.7	8.2
T5	7.35E+08	4.45E+16	2.27E+17	3.91E+04	3.61	0.62	5.1	19.4	7.6

(✔): The one with minimum errors is selected as the optimal estimation

Table 6. Five possible estimations for the rheological parameter set (for BN(6,6,12,12)).

Train No.	G^K (Pa)	η^K (Pa · s)	η^M (Pa · s)	ω_c (Pa/y)	ω_ϕ (°/y)	R_{thr}	e_{uc} (%)	e_{us} (%)	e_{ui} (%)
T1 (✔)	6.87E+08	3.48E+16	4.21E+17	5.08E+04	3.14	0.61	8.1	13.2	6.0
T2	9.03E+08	3.12E+16	1.22E+17	5.34E+04	2.94	0.53	24.4	57.1	24.6
T3	6.87E+08	3.48E+16	5.92E+17	6.02E+04	3.14	0.61	14.1	18.7	8.5
T4	5.36E+08	4.11E+16	4.47E+17	4.31E+04	1.50	0.79	5.6	21.5	7.9
T5	7.07E+08	3.32E+16	4.93E+17	2.89E+04	3.44	0.65	13.0	21.7	16.7

(✔): The one with minimum errors is selected as the optimal estimation

Table 7. Four optimal estimations for the rheological parameter set (for BNs with different architectures).

Architecture	G^K (Pa)	η^K (Pa · s)	η^M (Pa · s)	ω_c (Pa/y)	ω_ϕ (°/y)	R_{thr}	e_{uc} (%)	e_{us} (%)	e_{ui} (%)
BN(6,9,12)	6.04E+08	4.04E+16	3.83E+17	3.88E+04	2.84	0.49	4.0	12.1	9.3
BN(6,16,12)	6.11E+08	3.82E+16	3.28E+17	4.94E+04	2.55	0.61	3.1	15.3	12.4
BN(6,6,12,12)	6.87E+08	3.48E+16	4.21E+17	5.08E+04	3.14	0.61	8.1	13.2	6.0
BN(6,8,10,12)	7.10E+08	3.99E+16	2.97E+17	5.45E+04	2.71	0.65	3.5	12.1	12.0