International Conference on Advances in Nuclear Science and Engineering in Conjunction with LKSTN 2007(31-36)

Pattern Recognition of Material Creep and Fatigue Used in Nuclear Power Plant

Mike Susmikanti[1], Antonius Sitompul[2], Ari Handayani[2], Suratman[1], Arya Adhyaksa[1]
[1] Center for Development of Nuclear Informatics-BATAN, Tangerang, Indonesia
[2]Center for Nuclear Industry Material Technology-BATAN, Tangerang, Indonesia
Email : mike@batan.go.id

Abstract

PATTERN RECOGNITION OF MATERIAL CREEP AND FATIGUE USED IN NUCLEAR POWER PLANT. In the field of industry is an important thing to know the condition of material utilized especially in development of installation of nuclear. In many cases the materials with experiencing creep and fatigue cannot be obviated, but there is many ways to control it. Pattern recognition of material with condition of creep and fatigue bases on artificial intelligence is expected be able to assist some of effort for operations for the experts elaborating problem creep and fatigue. Material microstructure can be depicted and the pattern structure can be converted with image processing. Principal component analysis can determines the characteristic from the pattern. The next, an eigenvalues and eigenvector are to be used in neural network for the image of sample through the characteristic of the image. We choose the method of principal component analysis with covariance matrices. Modeling applies neural network system is studied and adapted from an object. The preceptron method is one of study with observation at neural network system with supervised learning. In learning and training, some example data will be taken in done some simulations. Microstructure from example data that is accurately applies laboratory data and references data. Pattern recognition process covers image processing, principal component analysis and neural network system by using MATLAB.

Keywords. *Creep, fatigue, Pattern Recognition, Image processing, Principal Component, Neural Network, Perceptron*

1. Introduction

Many problems encountered in the industry are a failure material. There are many kind of the material failure, such as creep and fatigue. Creep is engineers and metallurgist concern for components evaluating that operate under high stresses or high temperatures. Creep deformation is important in temperatures are endured such as nuclear power plants and heat exchangers. Creep and fatigue will affect the economic aspect during production, for example the work are stopped by repaired, the maintenance high cost, and energy was leakage effect.

Creep and fatigue have not be anticipated, but there are a lot of business to control it. Some important factors in creep and fatigue control is election of planning age determination, material structure applied, material mixture in precise, veneering and other [5]. In this case pattern recognition of material creep and fatigue will be elaborated, or creep and fatigue test base on artificial intelligence expected able assist some of effort for operations the experts elaborating creep and fatigue problem.

In material observation various equipment will be applied observation of material, with certain chemical process. Microstructure from material can be depicted by optical microscope, SEM, micrograph and other. The pattern structure can be converted with image processing process. The pattern characteristic is analyzed by principally component. Artificial intelligence modeling is learning and adaptation learning and adaptation an object. Identification of creep and fatigue are known. Perception method is a supervised learning method in neural network. The simulation can be used for identification and classification of material which is predicted which have creep or fatigue.

In learning and training process some sample data for simulations is taken. Microstructure laboratory data and reference data will process. The pattern recognition that is image processing, principal component analysis and neural network is done by MATLAB [3]. The result of the simulation can be compared with the result of sample in laboratory.

2. Creep and Fatigue

Creep is the term used to describe the tendency of a material to move or to deform permanently to relieve stresses. Material deformation occurs as a result of long term exposure to levels of stress that are below the yield or ultimate strength of the material. Creep is more destruct in materials that are subjected to heat for long periods near melting point. Creep is a monotonically increasing function of temperature [4]. Fatigue is a failure by repeated stress in materials. Fatigues are the progressive and localized structural damage that occurs when a material is subjected to cyclic loading. The maximum stress values are less than the ultimate tensile stress limit, and may be below the yield stress limit of the material [5]. The micrograph creep, fatigue or with neither creep or fatigue. (figure 1, 2 and 3)

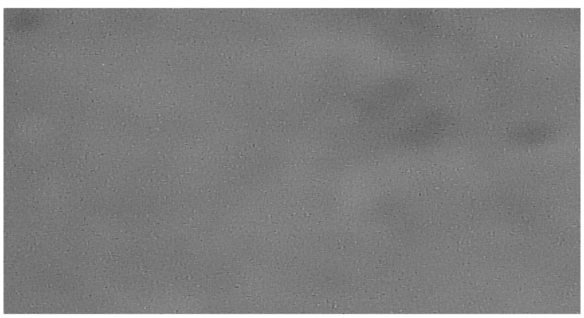

Figure 1. Micrographs neither Creep nor fatigue

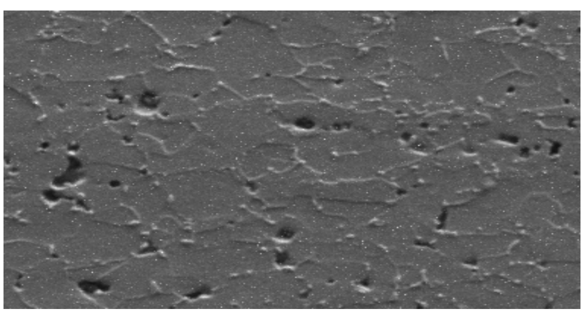

Figure 2. Micrographes creep

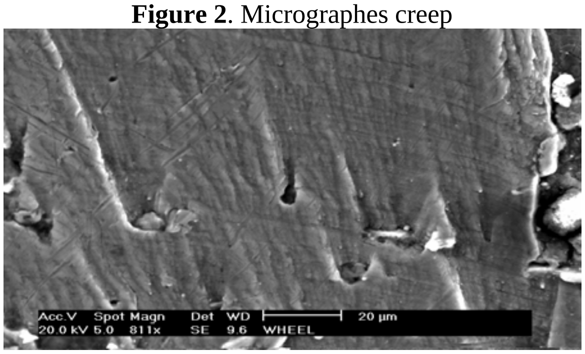

Figure 3. Micrographes fatigue

3. Pattern Recognition Systems of Creep and Fatigue

Pattern recognition of creep and fatigue covers three steps. Microstructure process of sample preparation creep and fatigue done in laboratory by equipment between of micrograph, optical microscope and or SEM. The first phase is applied image processing process, the second phase by principal component analysis and last applies in neural network. Pattern structure in the form of picture or photograph can be converted with image processing process to become digital form. To determine characteristic from the pattern, we used the analyzed principally component [1-12]. To identify is a material creep or neither creep nor fatigue applied by artificial intelligence of neural network [12]. For learning and training data, the sample will be taken some microstructures from preparation sample of result data and or from reference book. The simulation has been done for creep, fatigue and neither creep nor fatigue.

1. Image Processing

The pattern has a special texture, with various variation level of gray color. Average level of gray and standard deviation expressed as moment. Average of attributed to first moment, standard deviation depends on second moment and there are some other measures applied to express characteristic and texture area. In general expressed as [9-10],

$$M_n = \frac{\sum (x - \bar{x})^n}{N} \quad (1)$$

Where N is the number of data x, $\bar{x}$ is average and M_n is a sequence of moment.
Skewness and Kurtosis defined as,

$$S_k = \frac{1}{N}\sum\left(\frac{(x-\bar{x})}{\sigma}\right)^3$$

$$Kurt = \frac{1}{N}\sum\left(\frac{(x-\bar{x})}{\sigma}\right)^4 - 3 \quad (2)$$

This equation applied in the same way as standard deviation σ, calculates statistic to window and pixel value at center window and its surroundings.

2. Principal Component Analysis

Principal components analysis (PCA) is a technique used to reduce multidimensional data sets to lower dimensions for analysis [11]. Depending on the field of application, it is also named as discrete Karhunen-Loève transform, the Hotelling transform or proper orthogonal decomposition (POD). PCA is mostly used as a tool in exploratory data analysis and predictive models. PCA involves the calculation of the eigenvalue decomposition or Singular value decomposition of a data set, usually after mean centering the data for each attribute.

This analysis is an orthogonal transformation to new coordinate systems, so that the biggest variance with a projection stays at first co-ordinate (called as principal of first component), the biggest variance second at second co-ordinate and so on. For a data matrix with null median value (normal distribution of standard), Orthogonal Transformation PCA given as [6]:

$$Y^T = X^T W = V \sum \tag{3}$$

$V \sum W^T$ It is singular value decomposition from X^T. *PCA* for matrix data *X* given as

$$Y = W^T X = \sum V^T$$

(4)

Where $W \sum V^T$ is *single value decomposition* from X. PCA use covariance method or correlation method. In this case covariant method is applied with this algorithm:

(i) Data Set formed in matrix X of M x N
(ii) Data has standard normal distribution
(iii) Determines covariant matrix
(iv) Determines characteristic value and characteristic vector from matrix covariant
(v) Sorts characteristic vector V and characteristic value from matrix D in characteristic value sequence declines.
(vi) Cumulative value is calculated for every characteristic vector
(vii) Select gathering sub from characteristic vector as vector bases

6. Neural Network

Modeling of neural network is learning and adjustment an object. Perceptron method is learning method with observation in neural network systems, so that network yielded must have organically parameter by the way of changing through study order with observation. Neural network consisted of a number of associative neurons and a number of inputs. In designing neural network need attention in many specifications which will be identified. Preparation use perceptron for the application of pattern recognition described as element matrix between 0 and 1. First Layer of perceptron express a corps "sign detector" as input signal to know special sign. Taking second Layer output from special sign in first layer and classifies data pattern given. The learning process making relevant relationship between weight (w_i) and threshold value (θ). For problem two classes, layer output had only one node. Function of g_i in layer-1 is constant calculated before all, maps all or some of input patterns into binary value x_i {-1, 1} or value bipolar x_i {0, 1}. Set of output is element of linear threshold with threshold value following [2-7-8];

$$o = f(\sum_{i=0}^{n} w_i x_i - \theta)$$

$$o = f(\sum_{i=0}^{n} w_i x_i + w_0),\ w_0 \equiv -\theta \tag{5}$$

$$o = f(\sum_{i=0}^{n} w_i x_i),\ x_0 \equiv 1 \tag{6}$$

w_i: weight can be modified due to arrival of signal x_i, and w_0 = - (θ) be approach. Equation (5) indicates that threshold described as weight the relation of between set of output and an arrival signal of shadow x_0 (Fig 4). Function of f(.) be function of activation perceptron and this thing is special applied to a function of signum sgn(x) or function of step step(x):

sign(x) = 1 if x > 0
= 0 if other
step(x) = 1 if x > 0
= 0 if other

Learning procedure takes weight correlating to set of output (in last layer). If only predecessor weight at last layer altered, perceptron in figure 4 considered as perceptron layer unique. Learning algorithm of perceptron single layer is repeated to follows step following until convergent;

1. Select an input vector x from training data
2. If perception gives wrong answer, modification of all weights w_i as according ,

$\Delta w_i = \eta\, t_i x_i$; t_i : target of output and η : level of learning

Learning rule can be used with changing threshold value θ (=-w_0) according to equation (5). Level value of learning η can become constant through training or making convergent faster but can cause unstable learning.

For example architecture from neural network consisted of by neuron s and input r can expressed in figure 4;

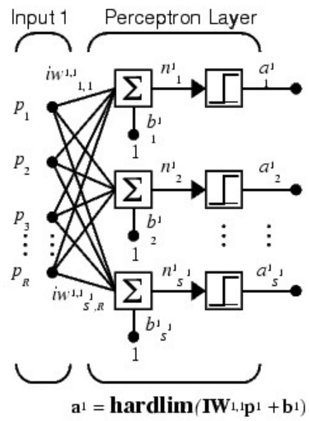

Figure 4. Architecture neural network architecture with *s* neuron and *r* input

7. Solution

The first phase covers process of pattern recognition creep, fatigue and neither creep or fatigue. The second phase covers image processing process and the third phase principal component analysis. The samples of material creep, fatigue and neither creep or fatigue are result of preparation which analyzed in laboratory. The sample data from reference book was taken to. At second phase, according to figure 4, we have matrix data 3 x 6 will be entered for every associative neuron concurrently. Taken six or the biggest characteristic values dominant from result (Appendix A), with three example of input given, what consisted of two example of input for material creep and fatigue and one example of input for material neither creep nor fatigue. Training data is taken three sample of input, the one sample of material with creep, the one of fatigue and the one for material neither creep nor fatigue.

Table 1. The output neural network identification

Sample	Right Quantity	Wrong Quantity	Accuracy
Creep	3	0	100%
Fatigue	2	1	67%
Neither creep nor Fatigue	3	0	100%

8. Conclusion

Pattern recognition process covers image processing steps, principal component analysis and neural network. Microstructure pattern of creep and fatigue material and without creep or fatigue can be expressed and converted in the digital form with image processing process. The principal component analysis can determine dominant characteristic from the pattern which can be applied at neural network systems. Modeling to differentiate classifies creep and fatigue material and neither creep nor fatigue material of neural network with method perceptron. In learning, the training and simulation result are appropriate as expected.

9. References

1. ADRIAN, GUNAWAN; cs, "*Hand Gesture Recognition Using Neural Network for Robotic Arm Control*", Proceedings of National Conference on Computer Science & Information Technology 2007 page 412, Faculty of Computer Science, University of Indonesia, Jakarta.
2. FU, LIMIN, "*Neural Networks in Computer Intelligence*", McGraw-Hill, Inc., Singapore, 1994..
3. HANSEL MAN, DUANE; LITTLEFIELD, BRUCE, "*Mastering MATLAB 5, A Comprehensive Tutorial and Reference*", Prentice-Hall Inc, 1998.
4. http://en.wikipedia.org/wiki/Creep (deformation) , 02/10/2007, page 1-6.
5. http://en.wikipedia.org/wiki/fatigue (material), 02/10/2007, page 1-9.
6. http://en.wikipedia.org/wiki/Principal_ components_analysis, 05/10/2007, page 1-10.
7. JANG, J.-S. R.; SUN, C.-T.; MISTUNE, E., "*Neuro Fuzzy And Soft Computing, A Computational Approach to Learning and Machine Intelligence*", Prentice-Hall International Editions, 1997.
8. KOSKO, BART, "*Neural Networks and Fuzzy Systems, A Dynamical Systems Approach to Machine Intelligence*", Prentice-Hall International Editions, 1992.
9. PARKER J. R., "*Algorithms for Image Processing and Computer Vision*", Wiley Computer Publishing, John Wiley & Sons, Inc, 1997
10. PEARSON, DON, "*Image Processing*", McGraw-Hill International Editions, 1991.
11. SUSMIKANTI, MIKE, "*Penyederhanaan Pemetaan Struktur Ketergantungan Variabel Menggunakan Teknik Prinsipal Komponen*",

Prosiding Lokakarya Komputasi dalam Sains dan Teknologi Nuklir XII, 2001, BATAN, Jakarta.

12. Susmikanti, Mike, ”*Pengenalan Pola Bahan Terkorosi Menggunakan Metoda Pembelajaran Perceptron Pada Sistim Jaringan Syaraf*’, Prosiding Seminar Nasional Aplikasi Teknologi Informasi (SNATI), 2006, Universitas Islam Indonesia (UII), Yogyakarta.

Appendix A: Image conversion program and calculates dominance characteristic value for one of image creep and fatigue

```
clear;
clc;
fb=fopen('fatique2_out.m','w');
X=imread('fatigue2.bmp');
DX=double(X);
COVDX=cov(DX);
fprintf(fb,'\n Eigen Value S \n');
EIGDX=eigs(COVDX);
fprintf(fb,'%9.2f\n',EIGDX);
[VS,DS,FLAG]=eigs(COVDX);
fprintf(fb,'\n VS \n');
fprintf(fb,'%9.2f',VS);
fprintf(fb,'\n DS \n');
fprintf(fb,'%9.2f %9.2f %9.2f %9.2f %9.2f
%9.2f\n',DS);
fprintf(fb,'\n Flag \n');
fprintf(fb,'%9.2f\n',FLAG);
fprintf(fb,'\n Principal componen from covariance
matrix:');
SVDS=svds(COVDX)
fprintf(fb,'\n SVDS : \n');
fprintf(fb,'%9.2f\n',SVDS);
[PC,LATENT,EXPLAINED=pcacov(COVDX);
fprintf(fb,'\n LATENT \n');
fprintf(fb,'%9.2f\n',LATENT);
fprintf(fb,'\n Explained \n');
fprintf(fb,'%9.2f\n',EXPLAINED);
fclose(fb);
```

Output::

```
Principal component from covariance matrix :
Eigen Value S :
   283350
   75900
   60630
   45440
   40890
   36820
SVDS :
283350  75900  60630 45440 40890 36820
```

Appendix B: The program of creep and fatigue test

```
clear;
clc;
fb=fopen('out_ciri_creepfatigue.txt','w');
% A. Tahap pembelajaran jaringan
% A.1 Deklarasi dan Inisialisasi jaringan neuron
  net1=newp([0 1;0 1;0 1;0 1;0 1;0 1],1);
  net2=newp([0 1;0 1;0 1;0 1;0 1;0 1],1);
  net3=newp([0 1;0 1;0 1;0 1;0 1;0 1],1);
  w1=net1.IW{1,1,1,1,1,1};
  w2=net2.IW{1,1,1,1,1,1};
  w3=net3.IW{1,1,1,1,1,1};
  b1=net1.b{1};
  b2=net2.b{1};
  b3=net3.b{1};
% A.2 Deklarasi matriks karakteristik
  p=[332687.6 134561.3 95660.9 62138.0 53141.5
46934.53;
  6281.0 5541.0 4633.4 4061.9 3405.2 3241.83;
  2969.8 2084.7 1086.8 898.3 635.5 600.44];
  pTotal=p';
% A.3 Declaration Target
  t1 = [1 0 0];
  t2 = [0 1 0];
  t3 = [0 0 0];
% A.4 Training proses
       net1=train(net1,pTotal,t1);
   net2=train(net2,pTotal,t2);
   net3=train(net3,pTotal,t3);
 fprintf(fb,' Learning process :\n');
       alearn=[sim(net1,pTotal)]
       fprintf(fb,'%4.0f',alearn,'\n');
       alearn=[sim(net2,pTotal)]
      fprintf(fb,'%4.0f\n',alearn);
      alearn=[sim(net3,pTotal)]
       fprintf(fb,'%4.0f\n',alearn);
% B. The Simulation Step
% B.1 Declaration and Initialization network
net11=newp([0 1;0 1;0 1;0 1;0 1;0 1],1);
net21=newp([0 1;0 1;0 1;0 1;0 1;0 1],1);
net31=newp([0 1;0 1;0 1;0 1;0 1;0 1],1);
  w1=net11.IW{1,1,1,1,1,1};
  w2=net21.IW{1,1,1,1,1,1};
  w3=net31.IW{1,1,1,1,1,1};
  b1=net11.b{1};
  b2=net21.b{1};
  b3=net31.b{1};
% B.2 Declaration Matrix Characteristic
  p1=[283354.0 75903.7 60627.2 45440.1 40892.3
36816.5;
  38318.9 16305.4 13079.4 9131.7 8699.7 8131.6;
  679.5 493.6 262.3 185.8 125.4 84.4];
  pTotal1=p1';
% B.3 Declaration Target
```

```
      t11 = [1 0 0];
   t21 = [0 1 0];
   t31 = [0 0 0];
% B.4 Training Proses
  net11=train(net11,pTotal1,t11);
  net21=train(net21,pTotal1,t21);
  net31=train(net31,pTotal1,t31);
fprintf(fb,' Training stage :\n');
atrain=[sim(net11,pTotal1)]
       fprintf(fb,'%4.0f\n',atrain);
atrain=[sim(net21,pTotal1)]
       fprintf(fb,'%4.0f\n',atrain);
atrain=[sim(net31,pTotal1)]
       fprintf(fb,'%4.0f\n',atrain);
% C. The first simulation
% C.1 declaration
  p2=[460912.9 54956.7 46000.4 35697.5 30608.2
24251.4;
   5923.8 5177.8 4457.7 4197.3 3933.7 3682.8;
   81156.8 36831.6 30236.2 23580.1 22749.1
16847.74];
  pTotal2=p2';
% C.2 Training Proses
  net11=train(net11,pTotal2,t11);
  net21=train(net21,pTotal2,t21);
  net31=train(net31,pTotal2,t31);
fprintf(fb,' Training process :\n');
  asim=[sim(net11,pTotal2)]
  fprintf(fb,'%4.0f\n',asim);
  asim=[sim(net21,pTotal2)]
  fprintf(fb,'%4.0f\n',asim);
  asim=[sim(net31,pTotal2)]
  fprintf(fb,'%4.0f\n',asim);
  fclose(fb);
```

Output :

TRAINC, Epoch 0/100
TRAINC, Epoch 25/100
TRAINC, Epoch 34/100
TRAINC, Performance goal met.
TRAINC, Epoch 0/100
TRAINC, Epoch 25/100
TRAINC, Epoch 31/100
TRAINC, Performance goal met.
TRAINC, Epoch 0/100
TRAINC, Epoch 2/100
TRAINC, Performance goal met.
alearn = 1 0 0
alearn = 0 1 0
alearn = 0 0 0
TRAINC, Epoch 0/100
TRAINC, Epoch 14/100
TRAINC, Performance goal met.
TRAINC, Epoch 0/100
TRAINC, Epoch 25/100
TRAINC, Epoch 50/100
TRAINC, Epoch 75/100
TRAINC, Epoch 100/100
TRAINC, Maximum epoch reached.
TRAINC, Epoch 0/100
TRAINC, Epoch 2/100
TRAINC, Performance goal met.
atrain = 1 0 0
atrain = 0 1 1
atrain = 0 0 0

TRAINC, Epoch 0/100
TRAINC, Performance goal met.
TRAINC, Epoch 0/100
TRAINC, Epoch 2/100
TRAINC, Performance goal met.
TRAINC, Epoch 0/100
TRAINC, Performance goal met.
asim = 1 0 0
asim = 0 1 0
asim = 0 0 0