Finite Element Analysis of Structures Based on Linear Saturated System Model

Hai-Bin Li[1, 2], Hong-Zhong Huang[1], and Ming-Yang Zhao[2]

[1] School of Mechanical Engineering, Dalian University of Technology
Dalian, 116023, China
hzhhuang@dlut.edu.cn
[2] Shengyang Institute of Automation, Chinese Academy of Sciences
Shengyang, 110015, China
{hbli, myzhao}@sia.ac.cn

Abstract. According to the property that global stiff matrix is positive definite after being adjusted and specific formation of elastomer potential energy function, linear saturated system model (LSSM) is introduced into finite element neurocomputing. Based on the neural network, a circuit implementation of an example is given and the time, error characteristic and simulation of the circuit are analysed.

1 Introduction

Finite element method was developed quickly after being brought forward in 1953 and it has become a most efficient structural analysis method. With structural analysis becoming more and more complicated, it is impossible to implement real-time calculation of complicated structure with finite element method because of the limitation of memory capacity and running speed of computer. So some structural analyst and mathematicians began to study parallel process of finite element structure after parallel computer came into being in the 1970's.

Neural network is a complicated non-linear dynamic system with great parallel computation ability. Neural network has been widely used in many fields such as optimization, pattern recognition, automatic control and economic prediction and so on since it revived in 1980's. Optimization problem can be mapped into a dynamic circuit with proper neural network, so solution to the problem can be obtained in circuit reaction time. The essence of neurocomputing is to construct proper network structure and learning method which can minimum the energy during running time and make the objective function value minimum when the system attains equilibrium. At present, widely used neural networks of optimization are Hopfield model and its modified models.

Neural network of optimization can be used to resolve structural analysis problems of elastic mechanics because elastic mechanics problems can be summed up to quadratic optimization under equality constraint. Some scholars have done a lot of works on the subject [1, 2]. At present, all the optimization neural network algorithms are all based on gradient descent algorithm of network energy, so it is inevitable that

F. Yin, J. Wang, and C. Guo (Eds.): ISNN 2004, LNCS 3174, pp. 820–825, 2004.

the optimization get into local minimum value. A new method that combines simulated annealing with gradient descent algorithm to searching global optimum solution was developed in [3]. Simulated annealing is a general random searching method that corresponds to the principle of mental annealing in physics. The searching direction can be chosen not only the better one but also the worse one in every step, so it can realize global searching. But simulated annealing is a global searching method based on probability, it can not converge to the global minimum value in complete probability which will bring great economic loss to important structural component. In the other hand, there is not a uniform theorem to choose the parameters in simulated annealing, such as initial temperature, internal cycle time criterion and stop criterion, which will affect the algorithm's performance. Moreover those parameters are chosen according to experience, which can not realized with fixed circuit. For the reasons mentioned above Linear Saturated System Model (LSSM) is introduced into finite element neural network according to the property that global stiff matrix are positive definite after being adjusted and specific expression of elastomer potential energy function. The finite element neural network can obtain solution without error in theorem. Circuit implementation to an example is given and the computation time and computation error are discussed.

2 Theorem Analysis

Finite element calculation of elastic mechanics can be summed up to quadratic optimization as following [1]:

$$\min_{x\in\Omega} \Pi = \frac{1}{2}\delta^{\mathrm{T}}\mathrm{K}\delta - \mathrm{q}^{\mathrm{T}}\delta \qquad (1)$$

$$\text{s.t} \quad \mathrm{A}\,\delta = \overline{\delta}(\text{in } S_u)$$

where

$$\mathrm{K} = \sum_{e=1}^{n}\int_{\Omega}(\mathrm{B}^{\mathrm{T}}D\mathrm{B})\mathrm{d}\Omega$$

$$\mathrm{q} = \sum_{e=1}^{n}\int_{\Omega}(N^{\mathrm{T}}\mathrm{d}b)\mathrm{d}\Omega + \sum_{e=1}^{n_s}\int_{S}(N^{\mathrm{T}}\mathrm{d}\overline{\mathrm{q}})\mathrm{d}S$$

N is displacement function, B is strain matrix, K is global stiffness matrix, q is node loading array, A is constraint matrix.

Above optimization can be expressed with linear equations:

$$\mathrm{K}\delta = \mathrm{q} \qquad (2)$$

where $\delta_{x_i} = \overline{\delta_i}$, $i = 1, 2, \cdots, \mathrm{s}$.

Proper row transformation is applied on Eq. (2), i.e. known constraint variables are moved to the bottom of the equations:

$$\begin{bmatrix} \mathrm{K}'_{(n-s)\times(n-s)} & \mathrm{K}'_{(n-s)\times s} \\ \mathrm{K}'_{s\times(n-s)} & \mathrm{K}'_{s\times s} \end{bmatrix} \begin{bmatrix} \delta_{(n-s)\times 1} \\ \delta_{s\times 1} \end{bmatrix} = \begin{bmatrix} \hat{\mathrm{q}}_{(n-s)\times 1} \\ \hat{\mathrm{q}}_{s\times 1} \end{bmatrix}$$

where $\delta_{s\times 1}$ is constraint vector, $\delta_{s\times 1} = \overline{\delta}_{s\times 1}$.

According to matrix operation theorem:

$$\mathrm{K}'_{(n-s)\times(n-s)}\delta_{(n-s)\times 1} = \hat{\mathrm{q}}_{(n-s)\times 1} - \mathrm{K}'_{(n-s)\times s}\overline{\delta}_{s\times 1}$$

let $\hat{\mathrm{q}}_{(n-s)\times 1} - \mathrm{K}'_{(n-s)\times s}\overline{\delta}_{s\times 1} = \mathrm{q}'$, $\mathrm{K}'_{(n-s)\times(n-s)}$ is denoted as K', then Eq. (1) can be transformed into unconstrained optimization problem:

$$\min_{x\in\Omega} \Pi = \frac{1}{2}\delta^{\mathrm{T}}\mathrm{K}'\delta - \mathrm{q}'^{\mathrm{T}}\delta \tag{3}$$

Building a neural network whose status function is:

$$\frac{\mathrm{d}\delta_i}{\mathrm{d}t} = -\frac{1}{\tau}\frac{\mathrm{d}E}{\mathrm{d}\delta_i} = -\frac{1}{\tau}\left(\sum_{j=1}^{m}\mathrm{K}'_{ij}\delta_j - \mathrm{q}'_i\right) \tag{4}$$

where τ is integral constant of integrator.

For the integrated circuit is restricted by maximum output voltage k i.e. $\delta_i \le k$ during the circuit implement, the neural network system is called Linear Saturated System Model, in brief LSSM.

According to matrix theorem, the analytical solution of Eq. (4) of neural network is:

$$\delta(t) = e^{-\frac{1}{\tau}\mathrm{K}'(t-t_0)}\delta(t_0) + \int_{t_0}^{t} e^{-\frac{1}{\tau}\mathrm{K}'(t-s)}\mathrm{q}'/\tau\mathrm{d}s \tag{5}$$

Suppose the input of neural network is $\mathbf{x}_0$ at the beginning, then solution to the differential equation is:

$$\delta(t) = e^{-\frac{1}{\tau}\mathrm{K}'t}\mathrm{x}_0 + e^{-\frac{1}{\tau}\mathrm{K}'t}\mathrm{q}'\mathrm{K}^{-1}(e^{\frac{1}{\tau}\mathrm{k}t} - 1) = e^{-\frac{1}{\tau}\mathrm{K}'t}\mathrm{x}_0 + \mathrm{q}'\mathrm{K}'^{-1}(1 - e^{-\frac{1}{\tau}\mathrm{K}'t}) \tag{6}$$

When t approximate to $+\infty$, $\delta(t)$ will approximate to a solution vector which has no business with initial value x_0, i.e. $\lim_{t\to\infty}\delta(t) = \mathrm{q}'\mathrm{K}^{-1}$. The equilibrium point of the neural network is the analytical solution to the finite element equations.

Mentioned as above that the integrated circuit is restricted by maximum input voltage k in LSSM, $\delta(t)$ must be bounded in the running process of neural network. The proving of boundness is given as following. According to Eq. (6) and matrix analysis theorem:

$$\|\delta(t)\| \le \left\|e^{-\frac{1}{\tau}\mathrm{K}'t}\right\|\|\mathrm{x}_0\| + \left\|\mathrm{q}'\mathrm{K}'^{-1}\right\|\left\|1 - e^{-\frac{1}{\tau}\mathrm{K}'t}\right\|$$

$$\le \|x_0\| + \|q' K'^{-1}\|$$

$$\le \|x_0\| + \|q'\| \|K'^{-1}\|$$

The result shows that $\delta(t)$ is bounded in the running process of neural network. In practical application, if $\|x_0\| + \|q'\| \|K'^{-1}\| \ge k$, let $q' = q'/n$ (where $q' = q'/n$) at first, then run the neural network. After obtaining calculation result δ, let $\delta = \delta n$, then the solution to the problem can be obtained.

3 Circuit Implement of Neural Network

Dynamic system of neural network with property of Eq. (4) is realized through circuit as following. Neurocomputation circuit system is composed of integrator, adder, feedback loop and reverse controller which realizes negative-resistance. Integrator is composed of simulated operation amplifier (LM324), capacitance and resistance. The I/O relationship is:

$$u_o = \frac{1}{RC} \int u_i \, \mathrm{d}t$$

Adder is a reverse adder that is composed of simulated operation amplifier (LM324) and resistance. The I/O relationship is:

$$u_{oi} = -k_{i1} u_{i1} - k_{i2} u_{i2} - \cdots - k_{in} u_{in}$$

where k_{ij} can be obtained through adjusting ratio of node resistance in the adder. Reverse controller is a special case of adder when the relationship of I/O is $u_{oi} = -u_i$. Neural network system to resolve finite element equations can be obtained through connecting adder, integrator, and reverse controller with lead.

4 Example

An example is given to show the application of neural network in finite element. Suppose a sheet whose thickness t = 0.1mm, Yangs' modulus E, poisson ratio μ = 0, load F = 10kN/m. Neglect gravity, try to find each node displacement with finite element and neural network method.

The 6 orthogonal points are input into neural network and the network is run. Calculation results are shown in Table 1.

The results in Table 1 show that whatever the initial points are chosen the neurocomputing dynamic system will convergence to the same point. But it is easy to find that there is little difference between the calculation result and analytical result (relative error is about 2%) because of the impact of maladjustment of input voltage and input current in integrated circuit. The phenomenon that the input values of neurocomputing are zero but the outputs are not zero are called zero-drift, shown as in Table 2. For improving the calculation accuracy, output should be adjusted. The adjusted value equal real output value subtract zero-drift value because that the network system is a linear one and the I/O property of linear system obeys superposition principle. Adjusted values are listed in Table 3. So it is very close between the adjusted values and analytical values, the maximum relative error is within 0.5%.

Table 1. Circuit simulation results of neurocomputing

Input variable	V_1	V_2	U_3	V_3	U_5	U_6
$\mathbf{X}_1$	-3.255	-1.253	-0.08865	-0.3742	0.1748	0.1726
$\mathbf{X}_2$	-3.255	-1.253	-0.08865	-0.3742	0.1748	0.1726
$\mathbf{X}_3$	-3.255	-1.253	-0.08865	-0.3742	0.1748	0.1726
$\mathbf{X}_4$	-3.255	-1.253	-0.08865	-0.3742	0.1748	0.1726
$\mathbf{X}_5$	-3.255	-1.253	-0.08865	-0.3742	0.1748	0.1726
$\mathbf{X}_6$	-3.255	-1.253	-0.08865	-0.3742	0.1748	0.1726
Analytical solution	-3.2527	-1.2527	-0.0879	-0.3736	0.1758	0.1758

Table 2. Zero-drift values of neural network

Output variable	V_1	V_2	U_3	V_3	U_5	U_6
Zero-drift value	-3.104×10^{-3}	-9.762×10^{-4}	-7.608×10^{-4}	-7.608×10^{-4}	-9.762×10^{-4}	-3.104×10^{-3}

Table 3. Adjusted results of neurocomputing

Output value	V_1	V_2	U_3	V_3	U_5	U_6
Adjusted value	-3.252	-1.253	-0.08857	-0.3734	0.1758	0.1757
Analytical solution	-3.2527	-1.2527	-0.0879	-0.3736	0.1758	0.1758

For comparing the calculation results with the results of Hopfield neural network and the results of M-TH neural network in [4], E=200MPa is adopted and the calculation results are listed in the same table, shown as in Table 4.

Table 4. Results of different calculation methods

Method	V_1	V_2	U_3	V_3	U_5	U_6
M-Hopfield	-0.016263	-0.006263	-0.000440	-0.001868	0.000879	0.000879
M-TH	-0.016257	-0.006261	-0.000439	-0.001862	0.000878	0.000878
LSSM(Protel 99)	-0.01626	-0.006265	-0.00044285	-0.001867	0.000879	0.0008785
Analytical solution	-0.016264	-0.006264	-0.000440	-0.001868	0.000879	0.000879

It can be seen from the Table 4 that the errors between the simulation results of LSSM on Protel 99/sim workstation and that of Hopfield neural network and M-TH neural network is within 0.5%.

According to the definition of circuit constant time, the constant time of analytical expression (5) of neural network status equation (4) can be defined as $\frac{1}{\tau}\|\mathrm{K}\|$. To specific problems, K is known, only $\tau = RC$ is adjustable. It is easy to obtain different computation speed of neurocomputing system only through adjusting R and C. Stabilizing time is listed in Table 5. It can be seen through Table 5 that network's convergence speed increases proportionally with the decrease of product of R and C.

Table 5. Stabilizing time of neural network

Circuit parameter	R=10, C=1uF	R=10, C=10uF	R=100, C=1uF
Stabilizing time	250us	2.5ms	2.5ms

5 Conclusions

Circuit simulation shows that questions can be resolved (error within 0.5%) in circuit constant time with finite element neural network based on LSSM system. The method developed in the paper and finite element neural network method based on Hopfield network both can obtain right result. Due to the need of circuit design, additional energy items, don't exist in object optimization energy function, come out in the energy function of optimization neural network during the optimization process of Hopfield neural network and its modified form (M-TH neural network). To obtain correct result, high gain ideal operational amplifier is usually chosen in circuit implementation of neural network. The amplifier can make the redundant energy approximate to zero. However, it is difficult to realize the ideal operational amplifier and the solution of the circuit will appear periodical oscillation when the gain of amplifier is big enough, which will result in calculation error and solution instability [5]. So finite element neurocomputing based on LSSM system is more practical. Moreover the number of parameters in finite element equations has no effect on computation speed of the neural network because neural network is parallel system.

References

1. Sun, D., Hu, Q., Xu, H.: Real Time Neurocomputing Theory and Numerical Simulation on Elastic Mechanics. Acta Mechanica Sinica 30 (1998) 348-352
2. Chen, X., Huang, H.-Z..: On the Neurocomputing Time and Error of Structural FEM. Chinese Journal of Mechanical Engineering 36 (2000) 22-26
3. Zhao, Z., Huang, H.-Z..: Method of Stochastic Neural Network Method for Structural Optimization Problem and Its Computer Simulation. Chinese Journal of Computational Mechanics 18 (2001) 242-245
4. Xie, M.: Development of Large Scale Integration Technology. World Science-Technologic Research and Development 23 (2001) 40-44
5. Jiao, L.: Neural Network Computation. Xidian University Press Sian (1996)