

Determination Of Process Conditions Of Epoxy Dispensing Processes Using A Genetic Algorithm Based Neural Fuzzy Networks

K.Y. Chan

Digital Ecosystems and Business Intelligence Institute,
Curtin University of Technology, Perth, Australia
kit.chan@curtin.edu.au

S.H. Ling,

Department and Electrical and Electronic Engineering,
University of Technology, Sydney, Australia
Steve.Ling@uts.edu.au

T.S. Dillon

Digital Ecosystems and Business Intelligence Institute,
Curtin University of Technology, Perth, Australia
Tharam.dillon@cbs.curtin.edu.au

C.K.. Kwong

Department of Industrial and Systems Engineering,
The Hong Kong Polytechnic University, Hong Kong
C.K.Kwong@inet.polyu.edu.hk

Abstract— In this paper, process conditions of epoxy dispensing processes are determined by the proposed genetic algorithm based neural fuzzy networks, which consists of two tasks: a) the approach of neural fuzzy networks, which was shown to be better than the other existing approaches, is proposed to develop models in relating between process parameters and quality characteristics for the epoxy dispensing processes; b) the approach of genetic algorithm is used to determine process parameters with respect to pre-defined quality requirements based on the developed neural fuzzy network models. The results indicate that, based on the proposed genetic algorithm based neural fuzzy network, estimated process parameters can achieve specified requirements of microchip encapsulations with high and robust qualities.

Keywords- Epoxy dispensing process, microchip encapsulation, neural fuzzy networks

I. INTRODUCTION

Epoxy dispensing is one of the most popular processes for microchip encapsulation on chip-on-board packages. In traditional semiconductor manufacturing, a trial-and-error method is usually used to identify proper process parameters setting so as to achieve the required quality characteristics. However, epoxy dispensing processes involve extremely complex inter-relationships among the fluid properties, process conditions, needle design parameters and overall encapsulation quality. Also cognitive uncertainties exist in the epoxy dispensing processes such as the human handling of substrate for measurement, insufficient data for process model development, and human judgment on qualities of the microchip encapsulation. Recently, various approaches have been attempted to model epoxy dispensing process including analytical approach [1], statistical regression [9], artificial neural networks [9] and fuzzy regression [7].

Statistical regression [14] is a commonly used empirical approach to develop manufacturing process models. However, statistical regression can only be applied if the given data are distributed along the developed statistical regression models,

and the relationship between dependent and independent variables is crisp [2]. Artificial neural networks can transform a nonlinear mathematical model into simplified black-box structures for various manufacturing processes [10]. They can be used for modeling nonlinear, complex and noisy manufacturing processes, but they are not recommendable by engineering users, as the models developed are in black-box nature and no information can be obtained within the models. Unlike these two approaches which required certain amount of data to develop models, fuzzy regression [13] has a distinct advantage that it can develop models using incomplete data sets or smaller amount of data. However, linear fuzzy regression models, which can only address linearity but not address nonlinearity, can only be generated by the fuzzy regression approaches.

Neural fuzzy networks have been used to develop models for various manufacturing processes which are both nonlinear and complex [15, 11]. The approach has the capability to transform a nonlinear mathematical model into a simplified input-output structure. Kang et al. [8] have proved that neural fuzzy networks outperform statistical regression in modeling of highly nonlinear systems. Compared with conventional fuzzy logic modeling, neural fuzzy systems can achieve better performance in mathematical function approximations, while the same number of fuzzy sets is used in input variables [5].

In this paper, a genetic algorithm based neural fuzzy networks is proposed to determine process condition for epoxy dispensing process of microchip encapsulation. First the approach of neural fuzzy networks is used to model epoxy dispensing for microchip encapsulation based on a set of experimental data. Validation results show that the process models developed based on neural fuzzy networks outperform those developed based on statistical regression, neural network and fuzzy regression. The developed process models are then used to formulate an optimization problem, which aims at specifying the optimal process parameters for the required quality characteristics. The process optimization is performed

by a genetic algorithm, which involves with the developed process models. The solutions of the optimization problem allow the optimal settings for the parameters of the epoxy dispensing process to be obtained. Validation tests are performed to evaluate the proposed approach to the modeling and optimization of the fluid dispensing process, and the results indicate that effective process parameter settings can be obtained that achieve the specified product quality requirements.

II. EPOXY DISPENSING PROCESS

In electronic packaging, normally, silicon chips are covered using an X-Y numerically controlled dispensing system that delivers epoxy encapsulant through a needle. The epoxy material is commonly dispensed in a pattern, working from the center out. A fluid dam around the die site and second wire bond points can be made to contain the flow of material and make a more uniform looking part as shown in Figure 1. The process is called the epoxy dispensing of microchip encapsulation. Two quality characteristics of encapsulation were also identified as

- a) encapsulation weight (mg), y
- b) encapsulation thickness (mm), z .

Three significant process parameters and their normal operating ranges were identified by the company supporting this research as:

- c) the compressed air pressure (1 bar to 4 bar) x_1 ;
- d) the pump motor speed (400 rpm to 1000 rpm), x_2 ;
- e) the height between the substrate and the needle (250 to 2000 steps of stepping motor), x_3 .

96 experiments were carried out based on a full factorial design with 4 levels in x_1 , 6 levels in x_2 and 4 levels in x_3 . 88 of the 96 experimental data were used to develop the model in relating the process parameters and the quality characteristics, and the rest 8 experimental data were used for the validation test.

To evaluate prediction accuracy of the process model, the mean error (Me) was used as defined below:

$$\text{Merror} = \frac{1}{n} \sum_{k=1}^n \frac{|t(k) - y(x(k))|}{t(k)} \quad (1)$$

where $t(k)$ is the actual value of the quality requirement for the k -th experimental data; $y(x(k))$ is the estimated quality requirement with the recommended process parameters $x(k)$ obtained from the process model; n is the number of experimental data sets used for the validation tests.

The variance of errors (Var) was also involved in the validation as shown below:

$$\text{Var} = \frac{1}{n-1} \sum_{k=1}^n \left(\frac{|t(k) - y(x(k))|}{t(k)} - \text{Merror} \right)^2 \quad (2)$$

III. MODELLING OF EPOXY DISPENSING USING NEURAL FUZZY NETWORK

Figure 2 shows the structure of a neural fuzzy network (NFN) which was used to develop models of the epoxy dispensing process in which the relationship between process parameters (x_1 , x_2 and x_3) and the quality characteristics (y and z) is addressed. NFN consists of an input layer which feeds in an input vector containing x_1 , x_2 and x_3 , and the output layer which produces the output response, y or z . Two hidden layers are connected between the input and output layers. The functions of the four layers of the NFN are described below:

Input layer: Each node in this layer which corresponds to a process parameter transmits input values to the next layer.

Hidden layer 1: Each node in this layer corresponds to one linguistic label (i.e. small, medium or large) with respect to x_1 , x_2 and x_3 . The membership value which specifies the degree to which an input value belongs to fuzzy set, is calculated in Layer 2. The j^{th} membership function of the input variable x_i is a bell-shaped function and is given by

$$A_{ij}(x_i) = e^{-\frac{(x_i - \bar{x}_{ij})^2}{2\sigma_{ij}^2}} \quad (3)$$

where parameters $\bar{x}_{ij}$ and σ_{ij} with $i=1,2,3$ and $j=1,2,3$ are the mean value and the standard deviation of the j^{th} membership function of the input variable x_i respectively.

Hidden layer 2: The node in this layer represents one fuzzy logic rule and performs precondition matching of a rule. For instance, a fuzzy rule $R_{g_1 g_2 g_3}$ can be given as:

$R_{g_1 g_2 g_3}$: IF x_1 is $A_{1g_1}(x_1)$ AND x_2 is $A_{2g_2}(x_2)$

AND x_3 is $A_{3g_3}(x_3)$ THEN $y(x_1, x_2, x_3)$ is $\mu_{g_1 g_2 g_3}$,

where $\mu_{g_1 g_2 g_3}$ is an output singleton y of the rule

$R_{g_1 g_2 g_3}$, and it can be represented as:

$$\mu_{g_1 g_2 g_3} = A_{1g_1}(x_1) \cdot A_{2g_2}(x_2) \cdot A_{3g_3}(x_3) \quad (4)$$

As the number of membership functions of each input variable is 3 and the number of input variables is 3, the total number of fuzzy rules is 27 (i.e. 3^3).

Output layer: The nodes in this layer corresponds to the output variable y . The node integrates all the outputs of hidden layer 2 and it acts as a defuzzifier with defined in (5).

$$y(x_1, x_2, x_3) = \frac{1}{\sum_{g_1, g_2, g_3=1}^3 \mu_{g_1, g_2, g_3} \left(w_1^{g_1, g_2, g_3} x_1 + w_2^{g_1, g_2, g_3} x_2 + w_3^{g_1, g_2, g_3} x_3 + w_0^{g_1, g_2, g_3} \right)} \quad (5)$$

where the weights $w_0^{g_1, g_2, g_3}$, $w_1^{g_1, g_2, g_3}$, $w_2^{g_1, g_2, g_3}$ and $w_3^{g_1, g_2, g_3}$ with $g_1, g_2, g_3 = 1, 2, 3$ are the neural fuzzy parameters.

$w_0^{g_1, g_2, g_3}$, $w_1^{g_1, g_2, g_3}$, $w_2^{g_1, g_2, g_3}$ and $w_3^{g_1, g_2, g_3}$ with $g_1, g_2, g_3 = 1, 2, 3$, $\bar{x}_{ij}$ and σ_{ij} with $i=1, 2, 3$ and $j=1, 2, 3$ (namely NFN parameters) were searched by using a genetic algorithm (GA) [6]. The objective of the genetic algorithm is to minimize the training error of the NFN model by searching the NFN parameters setting. The fitness function is defined as:

$$\text{fitness} = \frac{1}{1 + \text{err}} \quad (6)$$

where $\text{err} = \frac{1}{N} \sum_{\mathbf{x} \in \mathbf{X}} \frac{|y(\mathbf{x}) - y'(\mathbf{x})|}{y'(\mathbf{x})} \times 100\%$; $\mathbf{x} = (x_1, x_2, x_3)$,

$y'(\mathbf{x})$ is the actual output, $\mathbf{X}$ is the domain of all the training data, and N is the total number of training data. The range of fitness in (6) is between 0 and 1.

The GA starts with a population of randomly generated chromosomes, while each chromosome is set to be a set of neural fuzzy parameters. The length of each chromosome is equal to the number of neural fuzzy parameters which is 45. Thereafter, the GA further processes a fixed number of generations using the evolutionary operations (i.e. selection, crossover and mutation). The selection is performed in two steps, chromosome selection and survival. The selection on chromosomes decides who parents become and how many children the parents have. The children are created through crossover which exchanges information between parents, and the mutation which further perturbs the children. The fitness of each child is then evaluated based on the fitness function (6). Finally, the survival step decides who survives in the population. After the operations of selection, crossover and mutation, a new population is generated. The chromosome with the highest fitness is selected to be the setting of the neural fuzzy parameters.

The following parameter values have been used in the GA for searching the neural fuzzy parameters; crossover rate = 0.8; mutation rate = 1/45, where 45 is the length of the chromosome; number of generations = 1000; population size = 200. The GA were implemented using Matlab programming software. The training errors of the NFNs for the encapsulation weight and encapsulation thickness were found as 6.0263% and 5.2591% respectively.

Eight validation tests were carried out in order to evaluate the effectiveness of the developed NFN models in modelling the epoxy dispensing process. Table 1 shows the validation

results are compared with those based on our previously developed statistical regression, artificial neural networks and fuzzy regression for modelling the epoxy dispensing process [9]. Row 1 and Row 3 show the comparisons of the mean testing errors for the encapsulation weight and thickness respectively. Row 2 and Row 4 show that the comparisons of the variances of the testing errors for the encapsulation weight and thickness respectively. It can be seen that the NFN based process models can achieve the smallest mean testing errors and the smallest variance of mean errors compared with other methods. Therefore the NFN based process models for encapsulation weight and thickness predictions were used to perform the process optimization of the epoxy dispensing.

IV. PROCESS OPTIMIZATION USING GENETIC ALGORITHM

Process optimization aims at determining a setting of the process parameters so as to achieve the required encapsulation weight (β mg) and encapsulation thickness (γ mm). To determine the process parameter setting, the following objective function is used:

$$\text{Min } \rho = \lambda_1 \frac{|y - \beta|}{\beta} + \lambda_2 \frac{|z - \gamma|}{\gamma} \quad (7)$$

subject to: $1 \leq x_1 \leq 4$, $400 \leq x_2 \leq 1000$ and $250 \leq x_3 \leq 2000$,

where λ_1 and λ_2 are the weights of the two quality characteristics, encapsulation weight and encapsulation thickness respectively. As the two quality characteristics are supposed to be equal importance, $\lambda_1 = \lambda_2 = 0.5$ were used.

Figure 3 shows a GA for determining process parameters setting for the epoxy dispensing, and it shows how the GA is interconnected to the NFN for modeling the epoxy dispensing process. The GA aims at minimizing the objective function (7) from which the optimal values of x_1 , x_2 and x_3 can be obtained to achieve the required encapsulation weight β mg and encapsulation thickness γ mm. The main operations of the GA are described below.

A. Representation of strings

The first step of GA is to randomly generate strings which represent the population of GA. Real and binary encoding representations are two commonly used approaches for representing strings in GA. In binary encoding representation, strings need to be encoded to real values for fitness evaluation and also they need to be decoded again for reproduction operation. However, in real encoding representation, there is no need for string encoding and decoding, which would save a considerable amount of computational time and would lead to a much faster searching process than that of the binary encoding representation. Therefore, real encoding was adopted in GA, and each string contains three real values, which correspond to x_1 , x_2 and x_3 . The generated random values have the ranges associated with the operating ranges of the process parameters.

B. Fitness function

The objective function (7) was formulated to determine how close the randomly generated strings are to the required values of encapsulation weight (β mg) and encapsulation thickness (γ mm). Based on a string with the three process parameters, the quality measures of encapsulation weight y and encapsulation thickness z can be predicted by the NFN based process models. The fitness function of the GA is defined as follows:

$$\text{fitness} = \frac{1}{1 + \text{error}} \quad (8)$$

$$\text{where error} = \lambda_1 \frac{|y - \beta|}{\beta} + \lambda_2 \frac{|z - \gamma|}{\gamma}.$$

Based on the fitness function (8), the fitness of a string with x_1 , x_2 and x_3 can be evaluated. The objective is to maximize the fitness value of the fitness function (8) by searching the optimal x_1 , x_2 and x_3 .

C. Crossover operation

The intermediate crossover [12] for real encoding representation is used in the GA. It is a method of producing a new string around and between the process parameters of the two selected parent strings. The new string $[x_1^3, x_2^3, x_3^3]$ is produced based on the following formulation:

$$[x_1^3, x_2^3, x_3^3] = [x_1^2, x_2^2, x_3^2] \times \alpha \{ [x_1^2, x_2^2, x_3^2] - [x_1^1, x_2^1, x_3^1] \} \quad (9)$$

where α is a scaling factor chosen uniformly at random over some interval typically $[-0.25, 1.25]$, and $[x_1^1, x_2^1, x_3^1]$ and $[x_1^2, x_2^2, x_3^2]$ are the two selected parent strings. The process parameters in the new string are the result of combining the values of the process parameters in the parent strings according to (9) with a scaling factor α chosen for each process parameter. In geometric terms, intermediate crossover is capable of producing new parameter values within a slightly larger hypercube than that defined by the parent strings but constrained by the range of the scaling factor α .

D. Mutation operation

Mutation is carried out by randomly changing one or more process parameter values of the selected string between its pre-defined upper and lower bound. The mutation operator of Gaussian perturbation of individual parameters is used. For example, the parameter x_j is selected to be mutated. After performing mutation, its value becomes:

$$x_j' = x_j + \text{MutMx} \times R_j \times \delta \quad (10)$$

where $\text{MutMx} = +1$ or -1 with equal probability; $R_j = 0.5 \times$ searching domain of the process parameter x_j ; δ = a value in the range $[0,1]$ for shrinking the mutation range based on Gaussian perturbation.

E. Selection of strings

The population is evolved and improved iteratively until a stopping condition is met. In evolutionary computation, there are several stopping conditions. In this GA, the stopping

criterion is met when the number of generations is equal to a pre-defined number of generations. Otherwise, the population is evolved in the next generation. The approach of roulette-wheel, which is one of the most common selection methods used for selecting strings to perform reproduction operations [6], is used for strings selection in the GA. For example, the fitness of the j -th string in a population is assigned a value fitness_j . The fitness values are used to assign a probability value prob_j to the j -th string. The probability value prob_j is defined as:

$$\text{prob}_j = \frac{\text{fitness}_j}{\sum_{j=1}^{\text{Popsiz}} \text{fitness}_j} \quad (10)$$

where Popsiz is the population size of GA. Equation (10) shows that the string with a larger fitness value has higher probability to be selected. The updated population is produced by reinserting the new reproduced strings into the old population. It can be produced by replacing the least fitness strings in the old population with the new reproduced strings. However, pre-mature convergence likely occurs in early generations of evolution. To avoid the problem, random reinserting approach is used in the GA.

The population size, crossover rate and the mutation rate of the GA are set as 200, 0.8 and 0.001 respectively, which are suggested by DeJong [4]. The pre-defined number of generations is set at 1000. The GA toolbox coded in Matlab [3], which exists in the public domain, has been employed to search the process parameters of the epoxy dispensing process.

Eight validation tests were performed to evaluate the effectiveness of the GA. For each test, required values of encapsulation weight and thickness are input and it is expected that a setting of process parameters can be obtained from the optimization model. Table 2 shows the eight settings of process parameters recommended by the GA corresponding to the eight sets of quality requirements. The second and third columns show the required values of encapsulation weight and encapsulation thickness. The last three columns show the settings of the process parameters recommended by the GA.

V. CONCLUSION

In this paper, a genetic algorithm based neural fuzzy networks was proposed to determine process parameters setting of epoxy dispensing process for microchip encapsulation. 96 experiments based on a full factorial design were conducted in which the parameters of compressed air pressure, pump motor speed, distance between substrate and needle, and two quality characteristics, encapsulation weights and encapsulation thickness, are involved. 88 experimental data were used to develop the NFN based process models. 8 validation tests were carried out to evaluate the effectiveness and robustness of the process parameters settings recommended by the GA. The validation results indicate that, for each test, a setting of process parameters can be obtained to achieve the required quality requirement based on the proposed approaches.

Further work involves the validation of the searched process parameters obtained by the genetic algorithm based

neural fuzzy networks in a real world environment using experiments.

REFERENCES

- [1] D.X. Chen, Modeling and off-line control of fluid dispensing for electronics packaging. PhD thesis, The University of Saskatchewan, 2002.
- [2] Y. Chen, J. Tang, R.Y.K. Fung, and Z. Ren, "Fuzzy regression based mathematical programming for QFD", *International Journal of Production Research*, vol. 42, no. 5, pp. 3583-3604, 2004.
- [3] A. J. Chipperfield, P.J. Fleming, H. Pohlheim, C.M. Fonseca, *Genetic Algorithm Toolbox User's Guide*. ACSE Research Report No. 512, University of Sheffield, 1994
- [4] K.A. DeJong, An analysis of the behaviour of a class of genetic adaptive systems. PhD thesis, Department of Computer and Communication Science, University of Michigan, 1975.
- [5] A. Fiordaliso, "A constrained Takagi-Sugeno fuzzy system that allows for better interpretation and analysis. *Fuzzy Sets and Systems*", vol. 118, pp. 307-318, 2001..
- [6] D.E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. United States of America: Addison Wesley Longman, Inc, 1989.
- [7] K.W. Ip, C.K. Kwong, H. Bai and Y.C. Tsim, "Process modelling of epoxy dispensing for microchip encapsulation using fuzzy linear regression with fuzzy intervals", *International Journal of Advanced Manufacturing Technology*, vol. 22, pp. 417-423, 2003.
- [8] S.Y. Kang, H. Xie, and Y.C. Lee, "Physical and fuzzy logic modeling of a flip-chip thermocompression bonding process", *Journal of Electronic Packaging*, vol. 115, pp. 63-70, 1993.
- [9] C.K. Kwong, K.Y. Chan and H. Wong, "An empirical approach to modeling fluid dispensing for electronic packaging", *International Journal of Advanced Manufacturing Technology*, vol. 34, pp. 111-121, 2007.
- [10] H.L. Li and C.P. Chou., "Optimization of resistance spot welding process using Taguchi method and a neural network", *Experimental Techniques*, vol. 31, no. 5, pp. 30-36, 2007.
- [11] C.H. Ling, F.H.F. Leung, H.K. Lam and P.K.S. Tam, "Short-term electric load forecasting based on a neural fuzzy network", *IEEE Transactions on Industrial Electronics*, vol. 50, no. 6, pp. 1305-1316, 2003.
- [12] H. Muhlenbein and D.S. Voosen, "Predictive models for the breeder genetic algorithm: I. Continuous parameter optimization", *Evolutionary Computation*, vol. 1, no. 1, pp. 25-49, 1993.
- [13] B. Schaible and Y.C. Lee, "Fuzzy Logic Based Regression Models for electronics manufacturing applications", *Advances in Electronic Packaging*, vol. 1, pp. 147-155, 1997.
- [14] G.A.F. Seber, *Linear regression analysis*, Wiley, 2003.
- [15] S. G. Tzafestas and K.C. Zikidis, "NeuroFAST: on-line neuro-fuzzy ART-based structure and parameter learning TSK model", *IEEE Transactions on Systems, Man and Cybernetics—Part B: Cybernetics* vol. 31, no. 5, pp. 797-802, 2001.

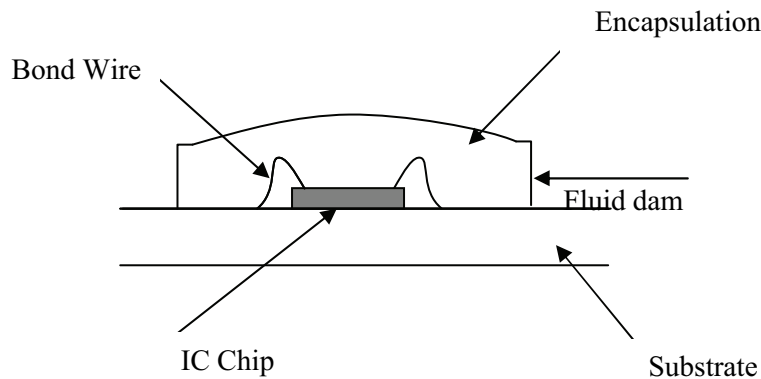


Figure 1 Encapsulation of IC chip using epoxy dispensing

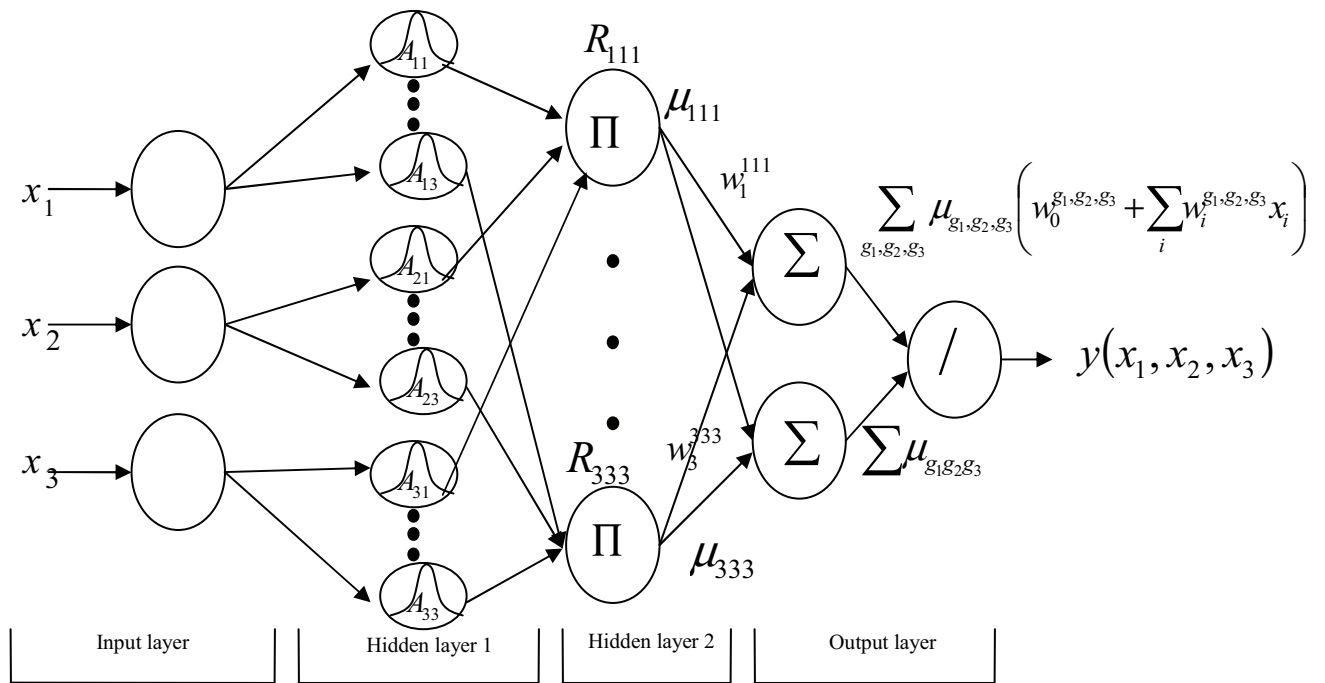


Figure 2 NFN based process model

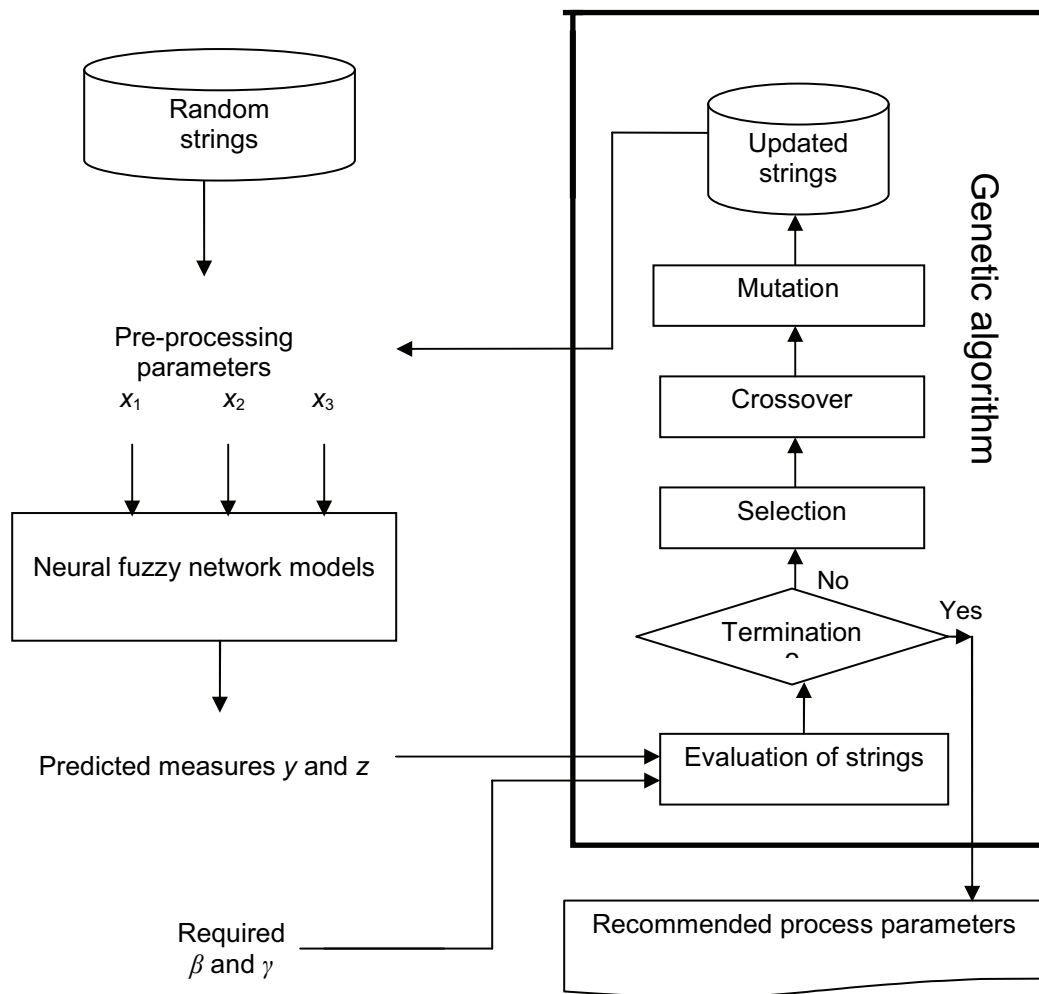


Figure 3 The genetic algorithm for determining optimal process parameters

Table 1 Mean testing errors and variances of testing errors for the various models

	Statistical regression model	Neural network model	Fuzzy regression model	TS neural fuzzy model
Encapsulation weight (mean)	5.3128	5.4532	6.5615	1.5409
Encapsulation weight (variance)	4.8835	3.8691	3.483	3.0637
Encapsulation thickness (mean)	5.3481	3.8271	18.4161	1.6947
Encapsulation thickness (variance)	20.318	4.7447	13.089	5.0879

Table 2 Recommended process conditions with respect to the pre-defined quality requirements

	Requirements of quality characteristics		Recommended process condition		
<i>Validation test</i>	<i>Weight</i> β (mg)	<i>Thickness</i> γ (mm)	x_3 (bar)	x_2 (steps)	x_3 (rpm)
1	72.3	0.58	2.2275	250	561.71
2	43.2	0.48	1.0653	404.15	807.75
3	87.4	0.67	4	2000	666.38
4	37.2	0.46	1.0118	550.75	890.19
5	75.1	0.62	2.5239	2000	635.45
6	59.3	0.57	3.2162	1377.5	855.07
7	62.4	0.53	1.6594	250	613.77
8	53.1	0.53	1.4084	1099.4	734.5