A VIRTUAL INSPECTION SYSTEM FOR PECVD MANUFACTURING PROCESS UTILIZING FUZZY LOGIC TECHNIQUE

Ming, Prof Chang[1], Jen-Cheng, Mr. Chen[2], Jia-Sheng, Prof Heh[3]

[1] Department of Mechanical Engineering, Chung Yuan Christian University, Chung Li, Taiwan 32023

[2] Department of Electronic Engineering, Chung Yuan Christian University, Chung Li Taiwan 32023

[3] Department of Information and Computer Engineering, Chung Yuan Christian University, Chung Li, Taiwan 32023

Abstract

A virtual inspection system for the on-line monitoring of the chemical vapor deposition (CVD) manufacturing processes utilizing fuzzy logic technique is presented. The system is a hybrid intelligence system combining the expert system with a neural network. In general a neural network is work as a black box; it means that the decision process of the system is unknown. Here the main structure is the expert system, but the expert knowledge library and the initiative learning mechanism are achieved by neural network with the on-line manufacturing process parameters. Combining the merits of these two techniques, a powerful and efficient rule extraction system is then established with a fuzzy logic method. These extracted rules can be used as an on-line diagnosis tool of manufacturing processes that offers a quantitative analysis of the variation caused by the change of process parameters. Experiments were implemented with a plasma enhanced chemical vapor deposition (PECVD) process. The obtained results show that the accuracy of the prediction of the deposited membrane thickness was roughly 90%, which prove this model can simulate the situation of membrane deposition and the on-line monitoring and e-maintenance of the manufacturing equipment could be expected with this technique.

1. Introduction

The semi-conductor and TFT-LCD are the two most important high-technology industrials currently. The manufacturing processes in the deposition stage of the twos are similar, with only the substrate material different. The R&D results in these two products are compatible and can be applied each other. In general, there are many factors would influence the deposited membrane quality. Most of them can be adjusted by changing the recipe, which are the process parameters of the working machines. Finding out a suitable and steady recipe and on-line real-time controlling the recipe is the target that process engineers devote to. Unfortunately, these processes are usually operated with complex and nonlinear reactions and the process parameters always drifting and varying [1], thus the development of on-line inspection and control of parameters is becoming increasingly important in these two industrials.

Generally speaking, the recipe adjustment is based on the accumulation of experiences or learning from the try and error results. However, the process of thin film deposition is a very complicate and nonlinear system. It is very difficult to find out the relationships between the variation of process parameters and membrane quality. Therefore, a hybrid intelligence system was developed here to simulate the CVD's process by combining the expert system with a technique of neural network [2, 3]. The expert system [4] is used for on-line inspection of manufacturing process and was set up by extracting out the regular rule between process inputs and outputs from the trained neural network [5], which could provide references to engineers for the need of on-line recipe adjustment. The neural network model is developed by using parameters of PECVD process and deposited membrane thickness, then extracting the ration fuzzy rules between input and output from neural network with fuzzy rule extraction algorithm [6, 7].

The data of analysis was collected from the PECVD machines in the TFT-Array factory of ChungHwa Picture Tubes (CPT), LTD. The machine parameters including the rates of all gases, temperature, pressure and bias of the chamber in membrane deposition process, etc. After arraying the data, use the parameters of machine as input, and membrane thickness as output, the neural network model was established with the technique of back-propagation network (BPN) [8] with 500 sets of training data.

The model was tested with 200 sets of test data and the error rate found was roughly ±9% in average, that prove this model can simulate and predict the quality of membrane deposition. The extract rules were also created from the trained neutral network to describe the expert knowledge with fuzzy logic rules [9]. After checking, explaining, and integrating the rules into the knowledge base, a fuzzy expert system [10] was then constructed and enables to provide a reference to the engineers for the work of recipe adjustment. Therefore, an intelligent decision making tool for the inspection and control of production process can be expected with this system.

2. Fuzzy Rule Extraction Method

2.1. Neural network and cube of input vectors

A *neuron* is an information processing unit that is the fundamental component of a neural network [11]. Generally, a neuron has N inputs and the inputs can be expressed as $\boldsymbol{x} = \{x_1, x_2, \dots, x_N\}$, where x_i .is usually normalized to a value between 0 and 1. The output of the neuron is $y = f(u(x))$, where $f(\cdot)$ is an specified active function and $f(u) = \mathrm{logsig}(u)$ [11] was adopted here; $u(\mathbf{x}) = \sum_{i=1}^{N} w_i x_i + w_0 = \mathbf{w}^T\mathbf{x} + w_0$, where $\boldsymbol{w} = \{w_1, w_2, \dots, w_N\}$ is the weight vector, and w_0 is the bias.
In general, the region of input x_i can be sorted and described with a fuzzy sets $Z_i \in [Z_{i1}, Z_{i2}, \mathbf{L}, Z_{im}]$, and each fuzzy set Z_{ij} (i = 1~N, j = 1~m) has a corresponding fuzzy range $[v_{ij}^L, v_{ij}^U]$. The value of x_i must fit in one of the sets, that is $0 \le v_{ij}^L \le x_i \le v_{ij}^U \le 1$ [12]. Similarly, the output y of the neuron is fit in one set of a fuzzy sets $Z^y \in [Z_1^y, Z_2^y, \mathbf{L}, Z_p^y]$, and each fuzzy set Z^y_k (k = 1~p) has a fuzzy range $[vy_k^L, vy_k^U]$ with values of $0 \le vy_k^L \le y \le vy_k^U \le 1$. In addition, Z^u_k is the fuzzy set corresponds to Z^y_k in u(x) domain, and it also has a fuzzy range $[vu_k^L, vu_k^U]$, where $vu_k^L = \mathrm{logsig}^{-1}(vy_k^L)$ and $vu_k^U = \mathrm{logsig}^{-1}(vy_k^U)$. For better understanding, some definitions are described below.

Cdelnlshlnm0'bt ad(9

A cube is a set which contains all input vectors of **x** in a neuron and can be expressed as ***cube*(w*)** , where $\boldsymbol{w}^* = \{w_1, w_2, \dots, w_N, w_0\}$ includes the weights and bias of the neuron.

Cdelnlshlnm1'ant mc ne bt ad(9

The *bound* of a cube is the maximum and minimum of $u(\mathbf{x})$ in the cube and can be shown as

bound*(*cube*(w*))** = [**L*bound , **U*bound***],

where the *lower bound* ***Lbound*** is the minimum of *u*(x) in the cube, and is equal to $\sum_{i, w_i < 0} w_i + w_0$; the *upper bound* ***Ubound*** is the maximum of *u*(x) in the cube, and is equal to $\sum_{i, w_i > 0} w_i + w_0$.

It is easy to find that the absolute maximum of $u(\mathbf{x})$ occurs at $x_i = 1$ for the weights w_i got positive value and $x_i = 0$ for the weights w_i is negative, and the minimum of $u(\mathbf{x})$ occurs at $x_i = 1$ with w_i is negative and $x_i = 0$ with w_i is positive. Accordingly, the upper bound of the cube is the sum of the bias w_0 and all w_i with positive values; and the lower bound of the cube is the sum of w_0 and all w_i with negative values.

2.2. Sub-cube

In order to extract out the fuzzy rules of a neuron for activating the neuron, the cube of the neuron must be sorted into sub-cubes and to find out a sub-cube with bound values lie between the fuzzy range $[vu_k^L, vu_k^U]$ of a fuzzy set Z^u_k. That means all inputs $\boldsymbol{x}$ of the cube would make the output y of the neuron equals Z^y_k. What we have done here is to assign x_1 to be sorted into certain ranges of Z_{11}~Z_{1m}, so that the original cube is divided into m sub-cubes, and each input of x_1 would be assigned to fit in one set of the sets Z_1. Similarly, each obtained sub-cube can be sorted again and divided into smaller sub-cubes with x_2 be assigned. Then with more x_i terms be assigned into sub-cubes, the smaller subset of the sub-cube of input vectors can be obtained.

Cdelnlshlnm2'rt a,bt ad(9

A *sub-cube* of a cube is a subset of the input vectors **x** in a neuron. When the first k terms of an input vector x_i in a cube are sorted and assigned, a sub-cube undergo k times sorting is obtained and can be expressed as ***cube*(w*,** $Z_1 Z_2 \dots Z_k$**)**, where $Z_1 Z_2 \dots Z_k$ are the related fuzzy sets of the first assigned x_i terms.

As known, each sub-cube is a sub-set of the input vector, and each sub-cube with different inputs would cause different values of $u(\mathbf{x})$.

Cdelnlshlnm3'ant mc ne rt a,bt ad(9

The *bound* of a sub-cube is the maximum and minimum of $u(\mathbf{x})$ in the sub-cube and can be shown as

$$\textbf{\textit{bound}}(\textbf{\textit{cube}}(\mathbf{w^*}, Z_1Z_2 \ldots Z_k)) = [\textbf{\textit{lbound}}, \textbf{\textit{ubound}}]$$

where the *lower bound* ***lbound*** of the sub-cube is the minimal $u(\mathrm{x})$ in the sub-cube, and the *upper bound* ***ubound*** of the sub-cube is the maximal $u(\mathrm{x})$ in the sub-cube, where

$$\textbf{\textit{lbound}}(\textbf{\textit{cube}}(\mathbf{w^*}, Z_1Z_2 \ldots Z_k))$$
$$= \min\{u(x) \mid x \in \textbf{\textit{cube}}(\mathbf{w^*}, Z_1Z_2 \ldots Z_k)\}$$
$$= \sum_{i=1, w_j \geq 0}^{k} v_i^{\,L} w_i + \sum_{i=1, w_j < 0}^{k} v_i^{\,U} w_i + \sum_{j=k+1, w_j < 0}^{n} w_j + w_0 ,$$

(1)

and

$$\textbf{\textit{ubound}}(\textbf{\textit{cube}}(\mathbf{w^*}, Z_1Z_2 \ldots Z_k))$$
$$= \max\{u(x) \mid x \in \textbf{\textit{cube}}(\mathbf{w^*}, Z_1Z_2 \ldots Z_k)\}$$
$$= \sum_{i=1, w_j \geq 0}^{k} v_i^{\,U} w_i + \sum_{i=1, w_j < 0}^{k} v_i^{\,L} w_i + \sum_{j=k+1, w_j \geq 0}^{n} w_j + w_0 . \quad (2)$$

The calculations of bounds via above equations are complicate. Since the bounds of a cube have been counted when the cube being divided into sub-cubes, a more effective method to find out the new bounds of the sub-cube can be achieved with a simpler calculation from the original bounds of the cube. That is, for any cube ***cube*(w*,** Z_1Z_2 ... Z_k**)** with bounds of [lbound$_k$, ubound$_k$], the bounds of the new sub-cube ***cube*(w*,** Z_1Z_2 ... Z_kZ_{k+1}**)** with x_{k+1} be sorted can be obtained from

$$[lbound_{k+1}, ubound_{k+1}] =$$
$$\begin{cases} [lbound_k + v_{k+1}^L * w_{k+1}, ubound_k - (1 - v_{k+1}^U) * w_{k+1}], w_{k+1} \geq 0 \\ [lbound_k - (1 - v_{k+1}^U) * w_{k+1}, ubound_k + v_{k+1}^L * w_{k+1}], w_{k+1} < 0 \end{cases} \quad (3)$$

The above equations are much simpler, so that the values of the bounds of the sub-cube could be real-time obtained. Through the calculated values of the bounds, some special cubes are defined as following.

Cdehmhshnm4'bdqs`hmbt ad(9

A sub-cube is called a ***certain cube*** if its **lbound** > $vu_k^{\,L}$ and **ubound** < $vu_k^{\,U}$. That means the output of the neuron would always fit in the fuzzy range of the set Z^y_k for all inputs of the cube.

Definition 6(uncertain cube):

A sub-cube is an *uncertain cube* if it is not a certain cube.

2.3. Rules of sub-cube

Since a certain cube's bound range always fits in the fuzzy range [$vu_k^{\,L}$, $vu_k^{\,U}$] of Z^y_k, the fuzzy rule to decide the output of a certain cube *cube*(w*, Z_1Z_2 ... Z_k) can be described as "IF x_1 is assigned in Z_1 AND x_2 assigned in Z_2 AND … AND x_k assigned in Z_k then the output of the neuron falls in the range of Z^y_k".

2.4. Algorithm

Bound Decomposition Tree

Input: A neuron's weights and bias

Output: Extract rules of the neuron

```
Step 1. Set certain cube sets $c_1$~$c_p$ as empty
        Set uncertain cube set as cube(w*)
        Set rule sets $r_1$~$r_p$ as empty
Step 2. Select a cube(w*)$_a$ form uncertain cube set
Step 3. Divide the cube(w*)$_a$ into m sub-cubes:
        cube(w*)$_{a1}$ ~ cube(w*)$_{am}$
Step 4. Calculate the bound of cube(w*)$_{a1}$ ~
        cube(w*)$_{am}$ with Eq. (3)
Step 5. Check cube(w*)$_{a1}$~ cube(w*)$_{am}$ separately,
        If the cube is a certain cube of $Z^y_k$, add it to
        certain cube set $c_k$
        Else add it to uncertain cube set
Step 6. If the uncertain cube set is not empty go to
        Step 2.
Step 7. For each certain cube sets $c_k$
            Transform each cube in $c_k$ into a rule and
        insert it into the rule set $r_k$.
Step 8. Then, each rule set $r_k$ contains the rules
        which make the neuron's output fit in the
        range of $Z^y_k$.
```

3. Results and Discussion

3.1. Training data

Here the training data of the neural network, process parameters and membrane thickness, were collected from the CVDs' Quality control (QC) department of CPT. These data has 700 sets of records from several CVD machines. The raw records contains 27 kinds of parameters, include flow rates of 11 gases, the maximum, minimum, and average value of DC bias (DC_BIAS), deliver power (DELIVER_PWR), reactor temperature (REACTOR_TEMP), reflect power (REFLECT_PWR), and radio-frequency peak (RF_PEAK). Five values of membrane thickness including maximum, minimum, average, uniformity, and standard deviation are chosen to be the outputs of the training data.

The correlation values between each input and outputs were first determined, and only the parameters which have high correlation with outputs were chosen as the inputs of the neural network. Finally 10 inputs were chosen as following

§ x_1. Average of DELIVER_PWR
§ x_2. Average of DC_BIAS
§ x_3. Average of RF_PEAK
§ x_4. Average flow rate of GAS_SF6

- § x_5. Average flow rate of GAS_H2
- § x_6. Average flow rate of GAS_NH3
- § x_7. Average flow rate of GAS_H2_PH3
- § x_8. Average flow rate of GAS_O2
- § x_9. Average flow rate of GAS_HE
- § x_{10}. Average of REFLECT_PWR

3.2. Training with the neural network

As it was mentioned, a neural network with 10 inputs and 5 outputs was built and learn from the training data. With 700 sets of collected records, 500 sets were used to be the training data of the neural network and the other 200 sets were used to be the test data.

The neural network model was established with back-propagation network. After training with those 500 sets of training data, we use test data to test the trained neural network. As shown in Fig.1, the obtained average errors for the maximum, minimum, and average of the predicted membrane thickness are ±9.78%, ±9.96%, and ±9.25%, respectively. Corresponding to the real values of membrane thickness are ±21.52Å, ±22.81Å, and ±16.48Å, respectively. Figure 2 shows the predicted results of the standard deviation and the uniformity of the membrane, where the tested errors are ±9.65% and ±9.92%, respectively. Since all of the predicted accuracies were over 90%, this means the weighting vectors have been adjusted to an acceptable situation.

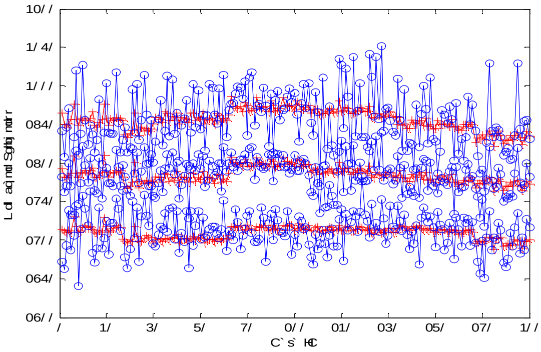
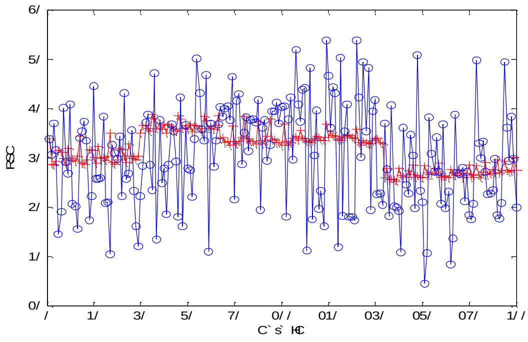
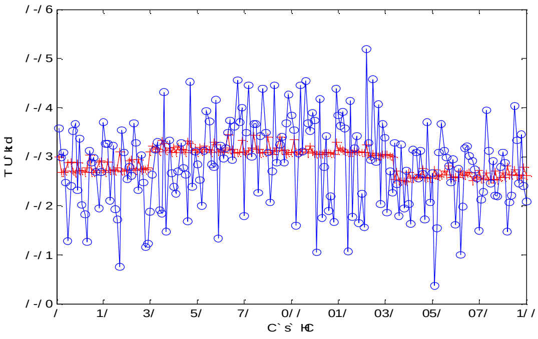


Figure 1: The predicted results of the trained neural network with the test data for maximum, minimum, and average value of the membrane thickness.



(a)



(b)

Figure 2: The predicted results for (a) the standard deviation and (b) the uniformity of the membrane thickness.

3.3. Extracted rules from the trained network

@esdq sgd mdtq`k mdsvnqj g`c addm sq`hmdc vhsg sgd sq`hmhmf c`s`+sgd etyyx qtkdr trdc enq sgd hmrodbshnm `mc bnmsqnk ne l`mte`bstqhmf oqnbdrr o`q`ldsdqr vdqd sgdm dwsq`bsdc eqnl sgd sq`hmdc mdtq`k mdsvnqj- Gdqd d`bg hmot sr v`r rdo`q`sdc hmsn 5 etyyx ynmdr $[Z_{i1}, Z_{i2}, \mathbf{L}, Z_{i6}]$ `bbnqchmf sn sgd qnv c`s` rdo`q`sdkx+ `mc d`bg nt sot sr v`r rdo`q`sdc hmsn 4 etyyx ynmdr $[Z_1^y, Z_2^y, \mathbf{L}, Z_5^y]$ `bbnqchmf sgd q`mfd ne l d`rtqdc l dl aq`md sghbjmdrr- Gdqd sgd nas`hmdc mtl adq ne dwsq`bsdc qtkdr enq d`bg nt sot s hr rgnvdc hm S`akd 0- A`rdc nm sgdrd dwsq`bsdc etyyx qtkdr+ `m dwodqs rxrsdl enq nm,khmd hmrodbshnm `mc bnmsqnk ne oqnbdrr o`q`l dsdqr b`m ad bnmrsqtbsdc vhsg ` bnl ot sdq-

Table 1 Numbers of extracted rules for each output.

	Max	Min	Avg	Std	U
Z^y_1	1633	0	6	51	32
Z^y_2	175234	7289	145226	12321	1225
Z^y_3	236717	30091	1053037	8578	710
Z^y_4	62322	11015	813515	611	127
Z^y_5	115	1	120547	0	2

3.4. Examples

As shown in Table 1, the extracted fuzzy rules were too complicate to be detail described; an example is chosen below to explain the application of these rules. Table 2 shows the first 10 extracted rules for the output of maximum of membrane thickness in the zone of Z^y_3. The number in the table is the fuzzy zone for each input and 0 means the state of "don't care". For example, rule R_1 indicates **"IF** x_2 is in the 4th fuzzy state Z_{24} **AND** x_5 in the 3rd fuzzy state Z_{53} **AND** x_6 in the 3rd fuzzy state Z_{63} **AND** x_7 in the fifth fuzzy state Z_{75} **AND** x_8 in the 2nd fuzzy state Z_{82} **AND** x_9 in the first fuzzy state Z_{91} **THEN** the maximum of membrane thickness will be in the 3rd fuzzy state of Z^y_3". That is **"IF** x_2 (Average of DC_BIAS) is between [-29.6 -15.76] **AND** x_5 (Average flow rate of GAS_H2) is between [29082 29099] **AND** x_6 (Average flow rate of GAS_NH3) is between [20165 20188] **AND** x_7 (Average flow rate of GAS_H2_PH3) is between [16.92 35.64] **AND** x_8 (Average flow rate of GAS_O2) is between [-3.36 0.92] **AND** x_9 (Average flow rate of GAS_HE) is between [-38.3 -29.18] **THEN** maximum membrane thickness will be between 1798~1842Å".

Table 2 The first 10 extracted rules for the control of maximum of membrane thickness in the Z^y_3 state.

	w_0	w_1	w_2	w_3	w_4	x_6	x_7	x_8	x_9	x_{10}
Q_0	/	3	/	/	2	2	4	1	0	/
R_2	4	4	/	/	3	1	4	1	0	/
R_3	5	2	/	/	3	2	5	0	1	/
R_4	4	4	/	/	4	1	4	1	1	/
R_5	3	3	/	/	3	0	4	0	0	/
R_6	3	5	/	/	2	1	4	0	0	/
R_7	3	3	/	/	3	2	5	1	0	/
R_8	/	3	/	/	4	0	2	2	0	/
R_9	2	3	/	/	4	0	4	1	0	/

3.5. Accuracy of the expert system

Based on these rules, a quantitative prediction of the membrane thickness was executed and compared to the true data. The obtained error rates for each output are shown in Table 3. As shown, very good results can be obtained in the control of maximum, minimum, and standard deviation of membrane thickness, but very poor for the output of uniformity. The main reason maybe caused by the tilt of substrate. Since this study was focused on the adjustment of process parameters, any problem induced by machinery will not be discussed here, and definitely the purpose of uniformity control can't fit in this system at present.

Table 3 Output errors based on the extracted rules.

	Max	Min	Avg	Std	U
Error Rate	3.76%	9.28%	18.61%	7.76%	34.88%

4. Conclusions

A fuzzy rule extraction algorithm based on neural networks for the inspection and control of membrane thickness in PECVD process is presented. In general the process is operated with complex and nonlinear reactions, the process parameters are always drifting and varying; so that an effective control of these parameters is important in improving the yield of product line. Experiments showed that the developed technique can present a control mechanism between those complicate manufacturing parameters and the deposited membrane thickness. The current accuracy for the control of membrane thickness and the standard deviation are roughly 90%, therefore the constructed knowledge base can provide a reference to the engineers for the work of recipe adjustments. At present the inspection work must be executed with the help of a computer since the extracted fuzzy rules are too complicate. A more effective pruning method should be expected in the future to reduce the number of rules and reach to an ability of real-time control.

5. Acknowledgement

The authors gratefully acknowledge the support of the National Science Council of Taiwan, the Republic of China, under project number NSC94-2218-E-033-007.

6. References

[1] Jansen, F., *AVS Short Course:PECVD*, American Vacuum Society, 1990

[2] Chang, M., Chen, J.C., Chang, J.W. and Heh, J.S., "Advanced Process Control Expert System of CVD Membrane Thickness Based on Neural Network", *Material Science Forum*, 505-507:313-318, 2006

[3] Chang, M., Chen, J.C. and Heh, J.S., "The Control of Membrane Thickness in PECVD Process Utilizing a Rule Extraction Technique of Neural Networks", *Lecture Notes in Computer Science*, 3973:1091-1098, 2006

[4] Buchnan, B. G. and Shortliffe, E. H., *Rule-based expert systems*, Addison-Wesley, New York, 1984

[5] Haykin, S.: Neural Netwroks. Prentice Hall, New Jersey, 1999

[6] Chen, J.C., Heh, J.S. and Chang, M., "Designing a Decompositional Rule Extraction Algorithm for Neural Networks", *Lecture Notes in Computer Science*, 3971:1305-1311, 2006

[7] Andrews, R., Diederich, J. and Tickle, A.B., "Survey and Critique of Techniques for Extracting rules from Trained Artificial Neural Networks', *Knowledge Base Systems*, 8(6):373-389, 1995

[8] Freeman, J.A. and Skapura, D.M., *Neural Networks*, Addison-Wesley, New York, Ch.3, 1992

[9] Freeman, J.A. and Skapura, D.M., *Neural Networks*, Addison-Wesley, New York, Ch.1, 1992

[10] Ross, T.J., *Fuzzy Logic with Engineering Application*, McGRAW-HILL, Singapore, Ch.2~4, 1997

[11] Klir, G. and Yuan, B., *Fuzzy Sets and Fuzzy Logic*, Prentice Hall, New Jersey, Ch.8, 1999

[12] Klir, G. and Yuan, B., *Fuzzy Sets and Fuzzy Logic*, Prentice Hall, New Jersey, Ch.12&14, 1999