'92 KACC 1992. 10. 19~21

Special Session 4

Applications of Neural Networks in Manufacturing Process Monitoring and Control

Hyung Suck Cho

Department of Automation & Design Engineering
Korea Advanced Institute of Science and Technology, Seoul Campus
P.O. Box 201, Cheongryangri, Seoul, Korea

ABSTRACT

Modern manufacturing process requires machine intelligence to meet the demands for high technology products as well as intelligence-based operating skills to lessen human worker's intervene. To meet this trend there has been wide spread interest in applying artificial neural network(ANN) to the areas of manufacturing process monitoring and control. This paper addresses application problems in such processes as welding, assembly, hydroforming process and inspection of solder joints.

1. Introduction

Manufacturing processes are becoming extremely sophisticated to meet the demands for high technology products. The sophisticated process characteristics demand sophisticated operating skills. Most of manufacturing processes suffer from the drawback that their operating parameters are usually preset with no provision for on-line adjustments. The on-line adjustments are ultimately needed to assure that production and quality levels are satisfactorily maintained, when processes suffer from variation of material properties, variation of machine conditions and changes in processing environments.

One solution is to utilize modern control techniques for on-line adjusting operating parameters based upon measurement of those parameters. This appears to be a typical feedback control method which regulates instantaneous values of the operating parameters to the preset ones. Current manufacturing process control technique developed up to now adopt this solution. Examples which employ such control techniques are current or voltage control in welding processes, punch stroke control in metal forming processes and forging processes, ram speed control in injection molding processes, voltage control for laser power variation in laser heat treatment processes, tool position or feedrate control in machining processes and roll position control in rolling processes. The critical operating parameters and product quality of each process are summarized in Table 1. This control strategy has drawback that, although the operating parameters are well controlled, good product quality cannot be guaranteed. This is due to the fact that the processes are often subject to variations of processing environments, while operated with those controlled operating parameters. This indicates that the process control requires the knowledge of instantaneous process operating states to cope with the changing environments.

Table 1. Operating Parameters and Product Quality in Various Manufacturing Processes.

Process	Operating Parameters	Product Quality /Defect
Welding	welding current/voltage wire feedrate torch speed gas content	weld pool size void
Metal forming	punch stroke punch speed	product geometrical dimension surface wrinkling fracture, necking
Hydroforming	hydraulic forming pressure punch stroke	product geometrical dimension surface wrinkling rupture, necking
Metal cutting	feedrate cutting speed	machining dimension surface roughness
Ring rolling	pressure roll feedrate conical roll feedrate	ring dimension surface quality
Laser heat treatment	laser power torch speed	surface hardening depth
Injection molding	ram speed vs. stroke, temperature	molecular orientation short size product geometry

The solution to overcome this problem is to control the operating parameters based upon direct measurement of product quality. This is extremely difficult because it requires specific knowledge of process dynamics as well as on-line quality measurement.

In recent years there has been wide spread interest in applying artificial neural networks (ANN) to such problem areas as machine diagnosis inspection of solder points on electronic circuit boards, machine vision system, robot control and so on. The neural networks mimics the computational architecture of human brain to achieve the intelligent capabilities such as learning and pattern recognition. The characteristics possessed by ANN appear to provide powerful solutions to inherent problems with which most of manufacturing processes seriously face, as discussed earlier. This paper addresses application problems in manufacturing processes such as welding process, assembly process, hydroforming process, and inspection of solder joints. The research works addressed here is confined to those conducted in Laboratory for Control Systems and Automation in KAIST.

2. Arc Welding Process

In GMA welding processes, the electric arc is generated by the flow of an electric current and maintained between the consumable wire electrode and the weldment, as shown in Fig.1. The consumable electrode is automatically fed by a wire feeding device and provides additional filler metal. To prohibit oxidation of the weld metal resulted from its exposure to the atmosphere at high temperatures, shielding gases are provided. The geometry of the resulting weld bead can be represented by the top bead width, the depth of penetration and the back bead width as shown in Fig.2. In a partial penetration welding, the size of the penetration depth is an important parameter to assess the integrity of the weld quality. While, in a full penetration welding the size of the back bead width is a significant factor. Because a good quality weld is characterized by the relatively high depth to width ratio of the weld bead, the integrity of the weld quality can be assessed by monitoring the penetration depth or the back bead width. To monitor and control weld geometry, the surface temperatures that are strongly related with the formation of the weld pool are utilized.

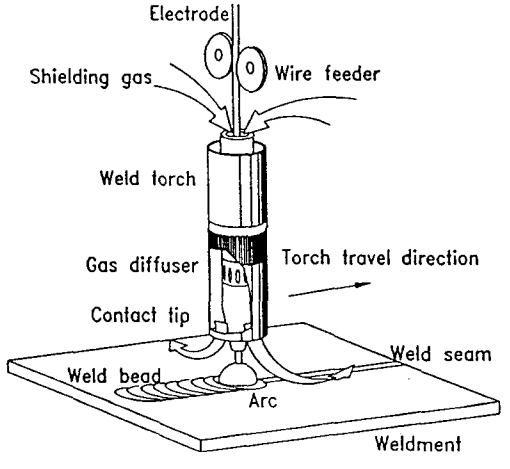

Fig.1 Schematic description of GMA welding process.

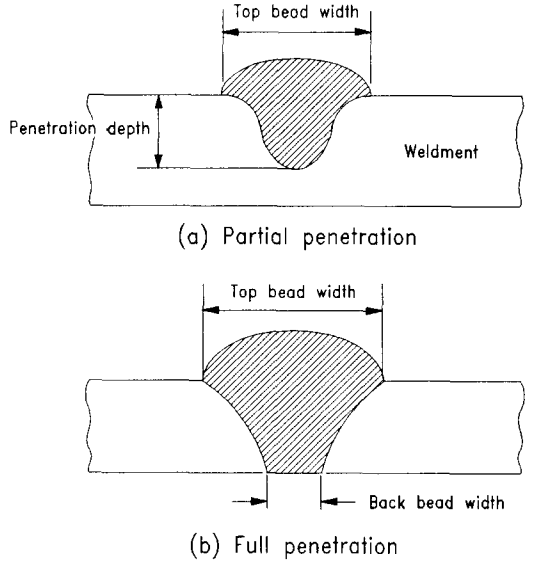

Fig.2 Cross sectional view of a weld pool.

Experimental System

Fig.3 shows the experimental apparatus used for quality monitoring and control. Temperatures on the top surface of the weldment are measured by noncontact infrared temperature sensing system. This temperature sensing system[1] is made of an objective lens, three apertures, three PbSe detectors and a signal processing circuit.

2.1 Weld Pool Size Estimation

Fig.4 shows the training and estimation procedures for the weld pool sizes from the surface temperatures

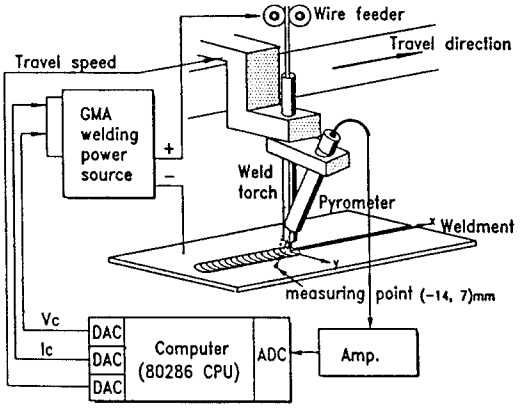

Fig.3 Schematic diagram of the experimental apparatus.

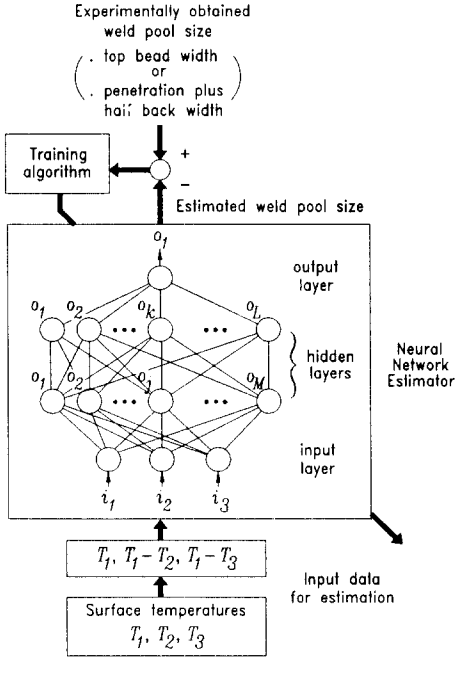

Fig.4 Schematic diagram of the training and estimation procedures for weld pool sizes using a neural network.

by using neural networks. Before the actual estimation is conducted using a neural network estimator, training is performed to determine the parameters necessary to construct the appropriate neural network estimator. The training data are obtained from the experiments conducted for various welding conditions. They include the point temperatures measured at the top surface of the weldment and the resulting weld pool sizes. The temperature information such as T_1, T_1-T_2 and T_1-T_3 and the corresponding weld pool size indicators, such as the top bead width and the penetration plus half back width are taken as the input and output parameters of the neural network estimator, respectively, as shown in Fig.4. In the beginning of the training, the estimated weld pool sizes may be largely different from the experimental ones, because construction of the appropriate neural network estimator is not completed. To reduce these differences the set of the network parameters are corrected according to the error back propagation algorithm. The weld pool sizes are recalculated based upon the recorrected network parameters and compared again with the experimental ones. This iterative training is performed for the entire training data set until the estimation errors fall within the tolerant values.

After the neural network estimator is completely trained, the actual estimation is performed using the newly measured temperature data which must not be included in the training data set.

2.2 Estimation Results and Discussions

Fig.5 shows the estimation results of the data set for the welding speed of 4 mm/sec when the neural network architectures are 3-4-2-1 nodes for the top bead width and 3-4-6-1 nodes for the penetration plus half back width. In Fig.5(a), the estimated weld pool sizes and the experimentally obtained ones are depicted in solid and hollow square boxes, respectively. As shown in this figure, the estimated weld pool sizes match very well up with the experimentally ones. The estimation errors range from -0.24 to 0.24 mm for the top bead width and from -0.24 to 0.41 mm for the

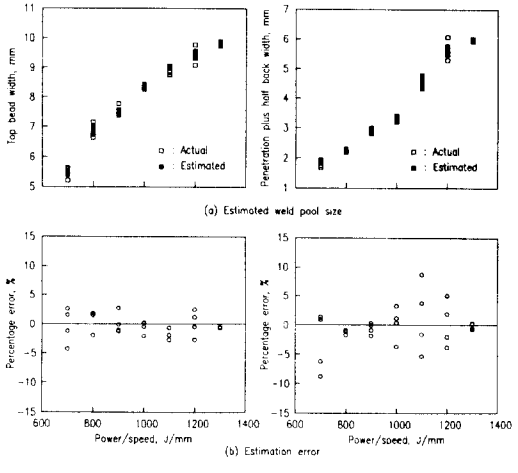

Fig.5 Estimation result of the weld pool sizes using a neural estimator: welding speed = 4 mm/sec.

penetration depth plus half back width. These results reveal that the relationship between the temperature information and the resulting weld pool sizes were successfully implemented in the neural network estimator, although the relationships are very complex and nonlinear. In Fig.5(b), the percentage estimation errors are presented. From this figure, it is shown that the estimation errors are within 5% for the top bead width, while 10% for the penetration depth plus half back width. The estimation errors for the penetration depth plus half back width are slightly larger than those for the top bead width. This is because the penetration depth plus half back width varies more nonlinearly with the welding conditions, thus deteriorating the characterization of the temperature - weld pool size relationship.

3. Assembly Process

Present assembly task requires an extremely high position accuracy and good knowledge of mating parts. However, actual assembly task suffers from the nonlinearities and uncertainties such as: 1)the nonlinear relationship between the corrective motion and sensor signals, 2)an imperfect knowledge due to the mechanical constraints of mating parts and their environments, 3)limitations of the devices performing the assembly, and 4) friction conditions. These nonlinearities and uncertainties results in several problems: 1)an exact model for recognizing the misalignment can not be constructed, 2)the relationship between sensor signals and misalignment is vague and inexact. To overcome above problems, a fuzzy set theory can be applied to assembly task[2]. However, in the fuzzy rule-based approach, the performance of assembly task depends heavily upon the fuzzy inference and rule base that can be constructed in various ways. Moreover, it is often difficult to gather effective rules sufficient to acquire good assembly performance because assembly strategy usually differs from geometrical conditions of the mating parts and their positions accuracy.

To cope with the difficulty associated the construction of rule base, we utilize two neuron-like elements to develop a self-learning rule-based assembly scheme whose rule base is improved iteratively until the assembly process is effectively performed.

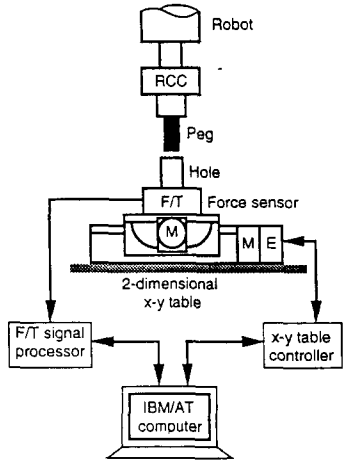

Fig.6 Schematic diagram of the experimental set-up.

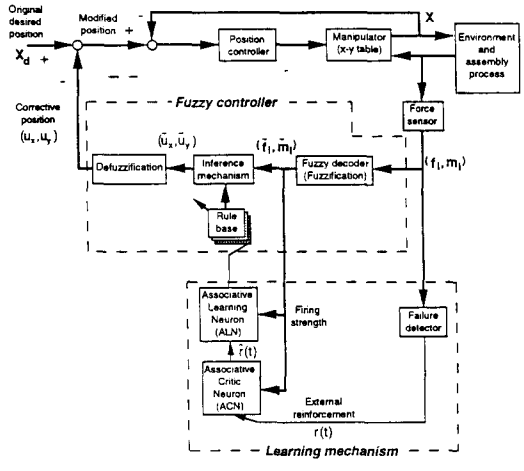

Fig.7 Block diagram of the proposed self-learning rule- based assembly algorithm.

3.1 Assembly System Configuration

The presented algorithm is described for the assembly system as shown in Fig.6. Namely, the assembly system consists of a 6-axis force sensor(Barry Wright), a x-y fine motion table, a SCARA robot, a RCC wrist with z-axis compliance(Barry Wright), and IBM/AT computer. With this assembly system, the block diagram of the presented assembly scheme, as shown in Fig.7, consists of two hierarchical levels. The lower one is a rule-based fuzzy controller which is composed of four modules: fuzzy decoder, rule base, fuzzy reasoning, and defuzzification. The higher one is the rule learning mechanism which is based upon two neuron-like elements[3], and learns control

rules iteratively until the assembly task can be successfully performed so that no further changers in rule base are necessary.

3.2 Rule Learning Mechanism

The force sensor-based rule consists of rules which connect the force elements($\tilde{f}_x$ and $\tilde{f}_y$) and moment elements ($\tilde{m}_x$ and $\tilde{m}_y$) to corrective motion($\tilde{u}_x$ and $\tilde{u}_y$) that has to be applied to the manipulator. The tild sign(~) represents fuzzy variable. The rule base takes the form:

IF $\tilde{f}_x$ is F_x^k, $\tilde{f}_y$ is F_y^k, $\tilde{m}_x$ is M_x^k, $\tilde{m}_y$ is M_y^k

THEN $\tilde{u}_x$ is U_x^k and $\tilde{u}_y$ is U_y^k

where k denotes the kth rule in the rule base. For the fuzzy rule, the objective of learning mechanism is to learn the linguistic fuzzy subsets, U_i^k, which is accomplished by two neuron-like elements: the associative critic neuron(ACN) and the associative learning neuron(ALN). The learning procedures, as shown in Fig.8, is as follows; at first, if force signals are measured, the fuzzy decoder investigates the current contact state of the mating parts, and generates outputs which are the activities of the fuzzy rules corresponding to input signals. The activity of each rule is summed, and sent to ACN and ALN. Then, in ACN, using the weighted sum of activities, p(t), and external evaluation signal, r(t), which evaluates whether rule learning is good (r(t)= 0) or bad (r(t) = -1); an internal evaluation signal, $\hat{r}(t)$, is generated and is served as an input of the ALN. The signal $\hat{r}(t)$ updates the memory of each rule by using corrective motion and associative trace relating to each rule. Finally, the value, $u_i^k(t)$, of fuzzy variable of control action part for fuzzy rule is generated by following equation:

$$u_i^k(t) = H(w_i^k(t)), \qquad i = x, y,\ k = 1,2,3,\ldots,n$$

$$w_i^k(t+1) = w_i^k(t) + \bar{\alpha}(t)\hat{r}(t)e_i^k(t), \quad \bar{\alpha} = \frac{\alpha\kappa}{\kappa + t}$$

where H is a bipolar continuous sigmoid function which takes the range from -1 to 1, $w_i^k(t)$ represents the i-axis memory of kth rule, $e_i^k(t)$ is the associative trace of the i-axis memory of kth rule, $\bar{\alpha}(t)$ is a positive learning parameter, and α and κ are initial constant values of $\bar{\alpha}(t)$.

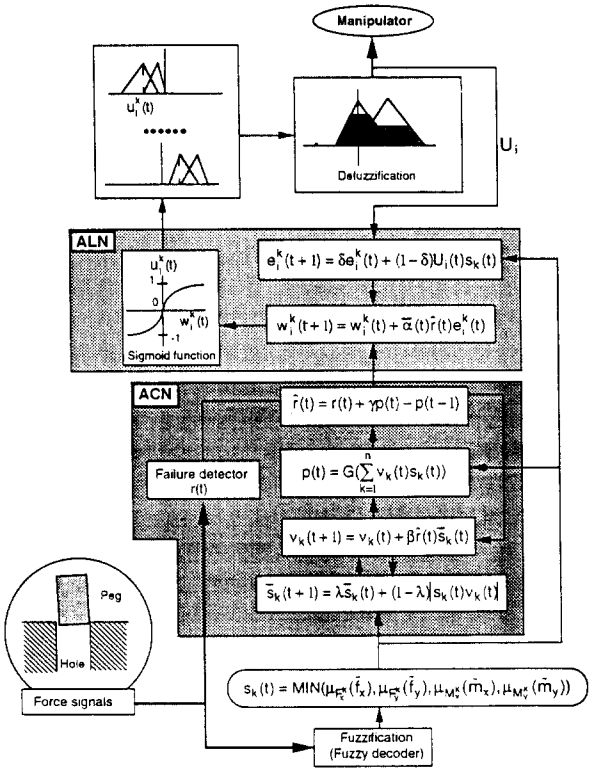

Fig.8 Signal flow of the proposed assembly scheme.

3.3 Experimental Results and Discussions

Fig.9 shows the learning performance of the

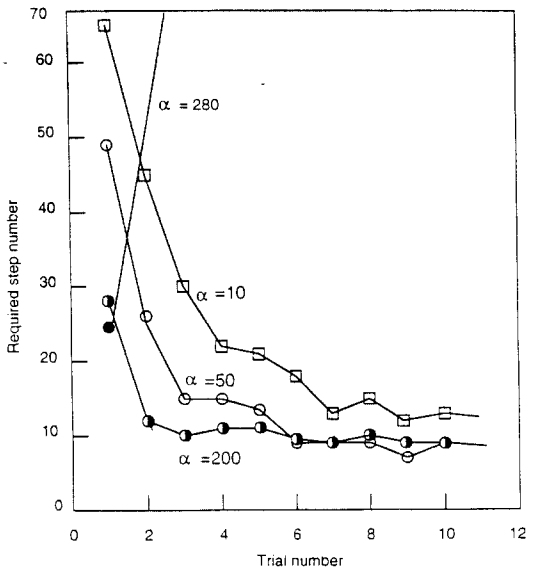

Fig.9 The effect for various values of learning parameter, α
($\beta = 0.8,\ \gamma = 0.8,\ \delta = 0.9,\ \lambda = 0.7,\ \kappa = 2500$)

proposed assembly method. As shown in the figure, the learning performance depends upon the values of initial value, α. In this experiment, other learning parameters were fixed as follows; $\beta = 0.8$, $\gamma = 0.8$, $\delta = 0.9$, $\lambda = 0.7$, $\kappa = 2500$. As can be seen from the figure, the larger the value is, the higher the convergency rate is. However, the α value must be limited because the response for the value larger than $\alpha = 280$ is shown to be unstable by this learning. On the other hand, other parameters were found to have a little effect on learning performance, and, thus, the results are not displayed here.

4. Hydroforming Process

The hydroforming operation is performed between the high pressure chamber controlled by a pressure control valve and the punch moving at a constant speed as shown in Fig 10. Since the pressure in the forming chamber is critical to the quality of the forming product, it is important to design an optimal curve for forming pressure vs. punch stroke. This curve is usually determined with consideration on the material properties, material thickness and geometrical shape of parts to be formed[4]. Currently, this operating curve is obtained by trial and error methods, thus demanding tremendous efforts and high cost.

The use of the neural network here is to replace the conventional trial and error method and automatically generates appropriate operating curves whenever any changes in part material properties and forming geometries necessitate new operating pressure curves.

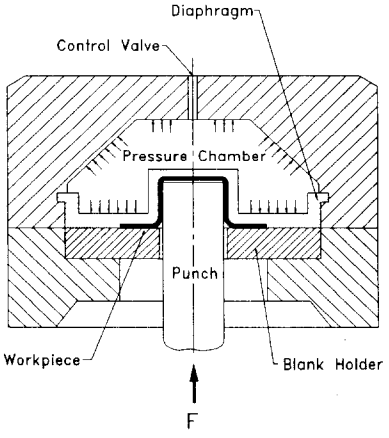

Fig.10 Schematic diagram for hydroforming process.

4.1 Neural Network-Based Pressure Curve Design

The variation of the chamber pressure with punch stroke determines instantaneous forming shape of a sheet metal part. The instantaneous forming shape is characterized by the geometric parameters such as blank area(A_b), flange area(A_f), contact angle between die and blank and projected area of the punch(A_p). These parameters are defined in Fig.11. As the punch goes upwards through the chamber, the pressure variation in the chamber alters the instantaneous shape of the part and, therefore, these geometric parameters change with punch stroke.

The problem is then how we can determine the pressure vs. punch stroke curve to obtain a formed part of high and uniform quality. To achieve this, a multi-layer perceptron is utilized to generate such curve for various punch shapes and sizes and drawing ratios. This implies that the network maps the nonlinear relationship between the geometric parameters and the

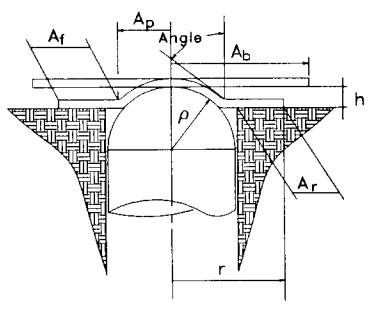

Fig.11 Geometric state variables for hydroforming process.

forming pressure throughout the punch stroke. The input variables(s) to the network are geometric parameters, while the network output(p) is the forming pressure value. Collecting all of the output pressure values yield the desired pressure curve during forming. As shown in Fig.12, the network has one input layer consisting of 5 nodes, two hidden layers consisting of 20 and 15 nodes each and one output layer. To train the network the error back propagation method is used. In this algorithm, the error, E , is defined by

$$E = \sum_k (\dot{p_k} - p_k)(\dot{p_k} - p_k)^T$$

where $\dot{p_k}$ denotes the desired pressure value for the k-th training sample and p_k is the corresponding

estimated pressure of the network.

Various punch shape, sizes and drawing ratios were used to train the network. For a given parameter condition a series of forming experiment were conducted to obtain the desired pressure vs. stroke. The desired pressure values $\dot{p}$ were so determined that thickness variation of the formed part is within 10% of the original part thickness. This choice is made because large thickness variation is detrimental to product quality.

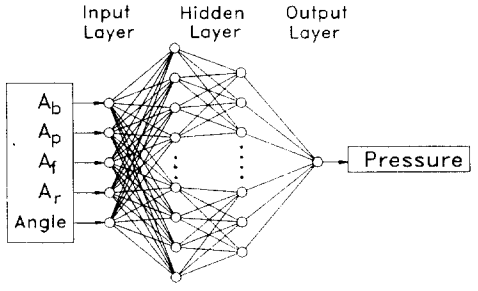

Fig.12 Multi-layered feedforward neural network.

4.2 Results and Discussions

Fig. 13 illustrates a typical pressure curve along the punch stroke. The solid line indicates the experimentally obtained curve, while the dotted line denotes that estimated by the neural network. As can be seen from the figure two curves are almost identical. This result indicates that the network exactly predicts the pressure curve for the given punch diameter and drawing ratio.

The performance of the trained network was tested using samples which have not been used for training. Fig. 14 shows the estimation error vs. number of sample. The result shows that maximum estimation error is within ± 10 bar. This is only less than 3% of the maximum value of the forming pressure which is shown to be approximately 350 bar as can be seen from Fig.13. The estimation error is distributed, but large number of samples show error occurrence at near zero bar. The conclusion obtained from these results is that the proposed network well estimates the forming pressure thus exhibiting its ability of designing an operating curve the pressure vs. punch stroke.

5. Inspection of Solder Joints

In printed circuit board assembly, quality inspection

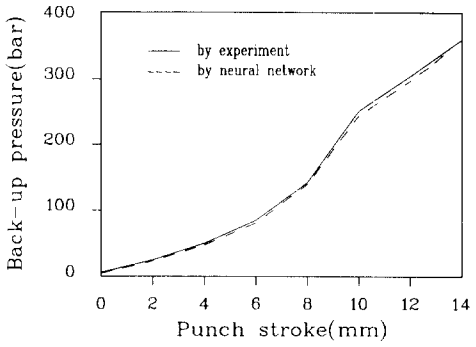

Fig.13 Typical pressure curve along punch stroke.

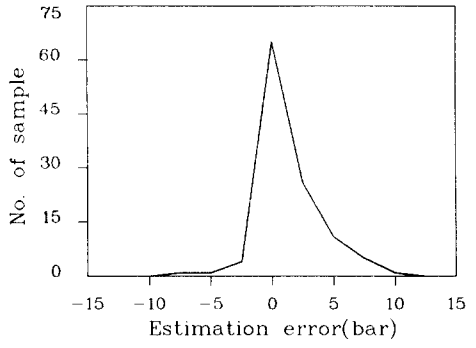

Fig.14 Number of sample v.s. estimation error.

is one of the most important process, because quality control is crucial to ensure functional reliability in the end product,. Quality inspection includes detection and classification of defects including no solder, insufficient or excess solder, power welting of component leads or solder pad. The inspection of such defects, however, requires not only the tedious joint by joint detection but also these skillful accurate test. Therefore, manual inspection used to date is gradually being replaced by the automated machine inspection which adopts image processing technology[5][6].

In this section, we describe a neural network based pattern classifier which identifiers each defect class.

5.1 Inspection System Configuration

Fig.15 shows the experimental test stand for inspection PCB solder joints. This system consists of solid state CCD color camera, illumination sources, a color image digitizer and IBM PC with a display monitor. The camera system has a 1:7 zoom lens giving a minimum field of view of approximately 3mm(H)x2mm(V) and a maximum field of view of approximately 21mm(H)x14mm(V). The illumination

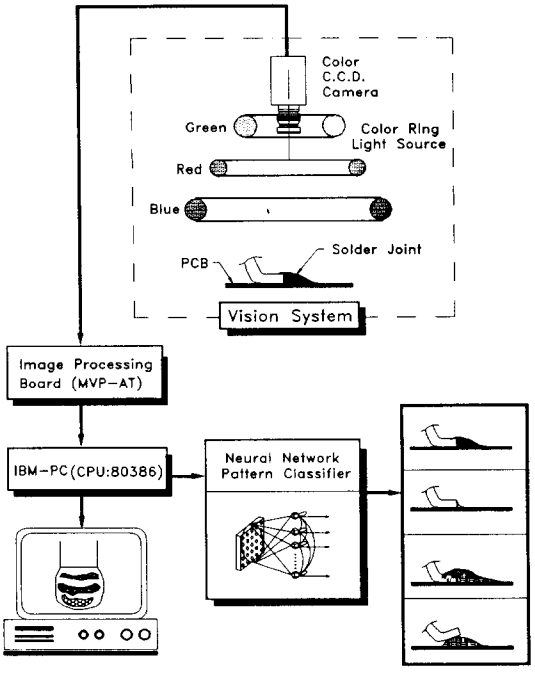

Fig.15 The schematic diagram of overall illumination and vision system.

sources are composed of three circular neon lamps mounted coaxially; high angle green light is positioned at the top, middle angle blue light is positioned next to the green , and low angle red light is placed above the PCB. The lighting arrangement enables us to correlate the solder joint profile and color planes. Depending upon the profile of the joint fillet, different color planes are obtained according to the incident angles of the color rays. The color images board digitizes images into 512(H)x512(V) pixels with 256 possible colors per pixel and stores it at a video frame rate of 1/30 seconds.

The monitor displays images directly from the camera or stored images from the digitizer board in which the IBM PC processes images to recognize each image pattern. In this study artificial neural network is used for pattern classifier. Once the image processings are completed, the recognition results are then, displayed in the monitor. The monitor indicates that the solder joints are good or bad; Among the bad no solder, insufficient or excess joint, bridge of components leads, separation of the leads from the board, and etc.

5.2 Neural Piecewise Linear Classifier

In general, classifiers use the features extracted from the real image data as inputs. However, It is time -consuming to extract features from image and difficult to determine which feature has a good separating ability. This directed us to use the raw image data as inputs to a classifier. When the raw images are used as the input to a classifier, its input dimension becomes very high. In this study, in order to overcome the above problem, a neural piecewise linear classifier[7] with parallel processing capability is used. This classifier is shown in Fig.16 which directly uses the raw solder joints images in order to classify the shape of solder joints.

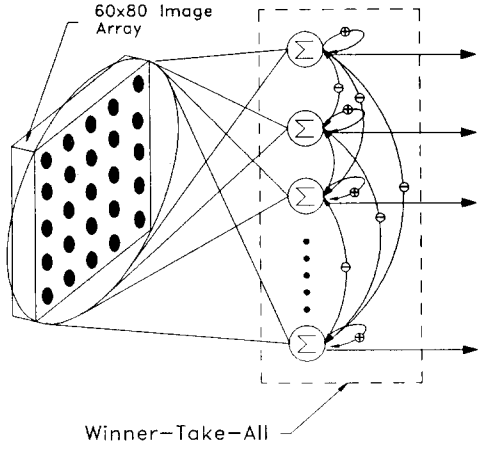

Fig.16 The structure of the neural network classifier.

Inputs to the neurons consisting of the output layer are determined by the input vector composed of the raw image data and the synaptic weights which correspond to the prototype vectors. The outputs of the neural network are expressed by the correlation of the input vector and the prototype vectors. The output of each neuron is positive-feedbacked to itself and negative-feedbacked to the other neurons. Then only one neuron with the maximum value converges to high positive value and the others converge to zero. In other words, the only one neuron of the nearest prototype vector to the current input vector which is called "winner" is selected and input image is classified into the class of the winner. Therefore, the position of the prototype vectors representing the classes determine the decision surfaces which separate the vectors into classes. Now, we are interested in how the prototype vectors are located in optimal positions in which the classification error is minimized.

The process of learning to locate the prototype in optimal positions are composed of two stages. In the first stage, the competitive learning vector quantization

(LVQ1) is used to find approximate positions of the prototype vectors for each class which is given by

$$m_i^w(t_{k+1}) = m_i^w(t_k) + \alpha(t_k)\,(x_i(t_k)-m_i^w(t_k))$$

where m_i^w is the prototype vector which has the minimum $x_i(t_k)^T m_i(t_k)$ value among the prototype vectors in ith class. This learning algorithm makes the prototype vectors located in the local means of the class distribution, but does not minimize the classification error because the variance of each class may differ.

In the second stage, LVQ2 is used to fine tune the position of the approximate prototype vectors learned in the first stage. The adaptation of the position of a prototype vector is carried out only if a misclassification occurs as follows;

$$m_i^w(t_{k+1})=m_i^w(t_k) - \alpha(t_k)\,(x_j(t_k)-m_i^w(t_k)) \quad \text{only if } i\neq j$$

where m_i^w is the prototype vector which has the minimum $x_j(t_k)^T m_i(t_k)$ value among all the prototype vectors. $x_j(t_k)$ belongs to jth class.

5.3 Results and Discussions

We implemented the neural classifier in C program in a 386 PC and a image processing board. Implementation in a parallel processing unit will be further realized. Fig.17 and Fig.18 show the solder joints images of 6 classes which are used in learning process. The numbers at the left corners represent the classes to which the input images within the boxes belong. First, we selected a prototype image representing each of 6 classes among 7 training images and then learned the prototype with 7 training images according to the competitive learning vector quantization algorithm(LVQ1). After this stage each prototype images converges to the mean of the training images of each classes. Second, the 6 prototype images was learned with 7x6 training images according to LVQ2, that is, only if the input image is selected randomly from 7x6 training images and the class of the nearest prototype to the input image is different from the class of the input image, the winner prototype position is updated backward from the input image

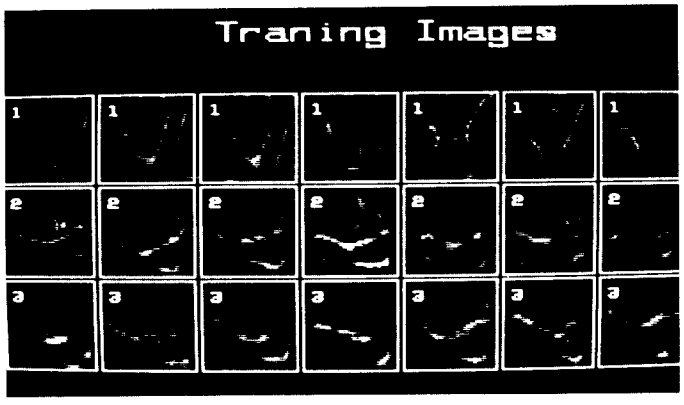

Fig.17 Training images of class 1 to 3.

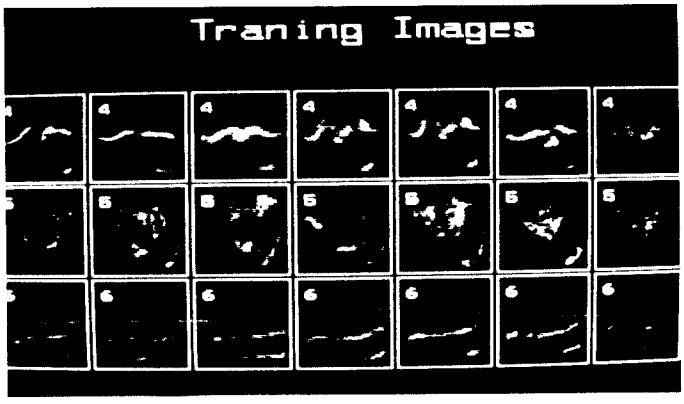

Fig.18 Training images of class 4 to 6.

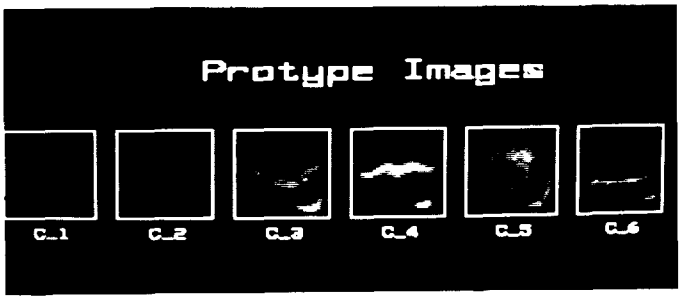

Fig.19 Prototype images of class 1 to 6 after learning.

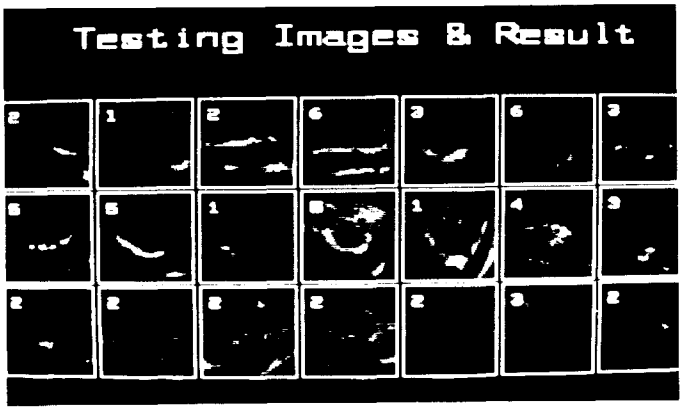

Fig.20 Testing images and results of classification by the neural network.

position. Fig.19 shows the prototype images representing each class after this two state of learning. The prototype image intensity values correspond to the synaptic weight vectors of neurons representing the classes shown in Fig.16.

The testing images shown Fig.20 were classified by the neural network classifier. The number in a left corner is the class of which the prototype image is the nearest to that input image. Although the input images in the third row is corrupted with noise, the neural classifier assigned the classes which we expected.

In this experiment, we used only one prototype image in each class. But if more prototype images in a class are used, more accurate decision surfaces can be obtained. From this experiment we can find that the classifier which has the raw image intensities as inputs directly has merit that it is not necessary to extract features. But it cannot be denied that the input dimension is so high that the computation burden is very heavy. To solve the problem we will further implement the algorithm using a parallel processing unit. It must be also noted that this algorithm is sensitive to position, rotation, distortion variances so that it is not appropriate to the case such as the recognition of handwritten characters in which pattern variance is very high. But this neural classifier has some insensitiveness inherently to those variances to some degree.

6. Conclusion

In this paper, several application problems based upon neural networks have been addressed for manufacturing process monitoring and control. These include on-line estimation of weld pool size in arc welding processes, self-learning adaptation of assembly rule base in assembly processes, automatic generation of operating curves for producing sheet metal products in hydroforming processes and image pattern recognition in PCB inspection processes. The application results show that the proposed neural network approaches can be effectively utilized for estimation, control and pattern recognition needed for product quality improvement in other manufacturing processes.

References

[1] K. S. Boo and H. S. Cho, "On the determination of

a temperature sensor location for monitoring weld pool sizes in GMAW", Submitted to Welding Journal, 1992.

[2] Y. K. Park and H. S. Cho, "A fuzzy rule-based assembly algorithm for precision parts mating," Mechatronics, 1992, (to appear).

[3] C. C. Lee, "Modelling behavioral substracts of associative learning and memory: adaptive neural models," IEEE Trans. on Systems, Man, and Cybernectics, pp.510-520, 1991.

[4] H. J. Park and H. S. Cho, "A fuzzy rule-based learning control method with application to hydroforming processes," Mechatronics, 1992, (to appear).

[5] David W, Capson and Sai-Kit Eng, "A tiered-color illumination approach for machine inspection of solder joints," IEEE Trans. on PAMI, vol.10,no.3, pp.387-394, May, 1988.

[6] Yuuji Takagi,Seiji Hata, and Whilhelm Beutel, "Visual inspection machine for solder joints using tiered illumination," SPIE vol.1386, Machine Vision Systems Integration in Industry, pp.21-29, 1990.

[7] Zhen-Ping Lo, and B. Bavarian, "A piecewise linear neural classifier," Proceedings of IJCNN, Seattle Washington, vol.1, pp.264-268, 1991.