Suck Cho, Hyung "Neural Network Applications to Manufacturing Processes: Monitoring and Control"
Computational Intelligence in Manufacturing Handbook
Edited by Jun Wang et al
Boca Raton: CRC Press LLC,2001

12

Neural Network Applications to Manufacturing Processes: Monitoring and Control

Hyung Suck Cho
Korea Advanced Institute of Science and Technology (KAIST)

12.1 Introduction

The nature of today's manufacturing systems is changing with greater speed than ever and is becoming tremendously sophisticated due to rapid changes in their environments that result from customer demand and reduced product life cycle. Accordingly, the systems have to be capable of responding to the rapid changes and solving the complex problems that occur in various manufacturing steps. The monitoring and control of manufacturing processes is one of the important manufacturing step that requires the capabilities described in the above.

Monitoring of the process state is comprised of three major steps carried out on-line: (i) the process is continuously monitored with a sensor or multiple sensor; (ii) the sensor signals are conditioned and preprocessed so that certain features and peaks sensitive to the process states can be obtained; (iii) by pattern recognition based on these, the process states are identified. Control of the process state is usually meant for feedback control, and is comprised of the following steps: (i) identifying the dynamic characteristics of the process, (ii) measuring the process state, (iii) correcting the process operation, observing the resulting product quality, and comparing the observed with the desired quality. It is noted that in the last step, the observed state needs to be related to product quality.

Normal operation of the above-mentioned steps should not be interrupted and needs to be carried out with little human intervention, in an unmanned manner if possible. To this end the process with this capability should be equipped with such functionalities as storing information, reasoning, decision making, learning, and integration of these into the process. In particular, the learning characteristic is a unique feature of the ANN. Neural networks are not programmed; they learn by example. Typically, a neural network is presented with a training set consisting of a group of examples that can occur during manufacturing processes and from which the neural network can learn. One typical example is to measure the quality-related variable of the process state and identify the product quality based on these measured data. The use of artificial neural networks (ANN) is apparently a good solution to make manufacturing processes truly intelligent and autonomous. The reason is that the networks possess most of the above functionalities along with massively computing power.

Utilizing such functionalities, ANNs have quite recently established themselves as the versatile algorithmic and information processing tool for use in monitoring and control of manufacturing process. In most manufacturing processes, the role of the artificial neural network is to perform signal processing, pattern recognition, mapping or approximation system identification and control, optimization and *multisensors data fusion*. In more detail, the ANNs being used for manufacturing process applications are able to exhibit the ability to

1. Generalize the results obtained from known situations to unforeseen situations.
2. Perform classification and pattern recognition from a given set of measured data.
3. Identify the uncertainties associated with the process dynamics.
4. Generate control signal based upon inverse model learning.
5. Predict the quality from the measured process state variables.

Due to such capabilities, there has been widespread recognition that the ANNs are an artificial intelligence (AI) technique that has the potential of improving the product quality, increasing the effect events in production, increasing autonomity and intelligence in manufacturing lines, reducing the reaction time of manufacturing systems, and improving system reliability. Therefore, in recent years, an explosion of interest that has occured in the application of ANNs to manufacturing process monitoring and control.

The purpose of this chapter is to provide the newest information and state-of-the-art technology in neural-network-based manufacturing process monitoring and control. Most applications are widely scattered over many different monitoring and control tasks but, in this chapter, those related to product quality will be highlighted. Section 12.2 reviews basic concept methodologies, and procedures of process monitoring and control. In this section the nature of the processes is discussed to give reasons and justification for applying the neural networks. Section 12.3 deals with the applications of neural networks in monitoring various manufacturing processes such as welding, laser heat treatment, and PCB solder joint inspection. Section 12.4 treats neural-network-based control and discusses the architecture of the control system and the role of the network within the system. Various manufacturing processes including machining, arc welding, semiconductor, and hydroforming processes are considered for networks applications. Finally, perspectives of future applications are briefly discussed and conclusions are made.

12.2 Manufacturing Process Monitoring and Control

In this chapter, we will treat the problems associated with monitoring and control of manufacturing processes but confine ourselves only to product quality monitoring and control problems. Furthermore, we will consider only on-line monitoring and control schemes.

12.2.1 Manufacturing Process Monitoring

Product quality of most processes cannot be measurable in an on-line manner. For instance, weld quality in the arc welding process depends on a number of factors such as the weld pool geometry, the presence of cracks and void, inclusions, oxide films, and the metallographic conditions. Among these factors, the weld pool geometry is of vital importance, since this is directly correlated to weld strength of the welded joint. The weld pool size representative of weld strength is very difficult to measure, since the weld pool formed underneath the weldment surface represents complex geometry and is not exposed from the outside. This makes it very difficult to assess the weld quality in an on-line manner. Due to this reason, direct quality monitoring is extremely difficult. Thus, one needs to resort to finding some process state variables that can represent the product quality. In the case of arc welding, the representative variable is the temperature spatially distributed over the weld pool surface, since formation of the weld pool geometry is directly affected by heat input. In this situation, the weld quality can be indirectly assessed by measuring the surface temperature.

Two methodologies of assessing product quality, are considered. One is the *direct method*, in which the quality variables are the monitoring variables. The other is the *indirect method*, which utilizes the measured state variable as measures of the quality variables. In this case, several prerequisite steps are required to design the monitoring system, since the relationship between product quality and process condition is not known *a priori*. In fact, it is very difficult to understand the physics involved with this issue. The prerequisite steps treat the issues, which include (i) relating the product quality with the process state variables, (ii) selection of sensors that accurately measure the state variables, (iii) appropriate instrumentation, and (iv) correlation of the obtained process state data to quality variables. The procedure stated here casts itself a heavy burden in monitoring of process condition problem. Once this relationship is clearly established, the quality monitoring problem can be replaced by a process state monitoring problem.

Figure 12.1 illustrates the general procedure of evaluating product quality from measurement of process variables and/or machine condition variables. This procedure requires a number of activities that are performed by the sensing element, signal interpretation elements, and quality evaluation unit. The sensors may include multiple types having different principles of measurement or multiples of one type. In using sensors of different types, sensing reliability becomes very important in synthesizing the information needed to estimate the process condition or product quality. The reliability may change relative to one another. This necessitates careful development of a synthesis method. In reality, in almost all processes whose quality cannot be measured directly, multisensor integration/fusion is vital to characterize the product quality; for instance weld pool geometry in arc welding, nugget geometry in resistance spot welding, hardened layer thickness in laser hardening, etc. This is because, under complex physical processing or varying process conditions, a single sensor alone may not adequately provide the information required to make reliable decisions on product quality or process condition. In this case, sensor fusion or integration is effective, since the confidence level of the information can be enhanced by fusion/integration of the multiple sensor domain. This multiple sensor approach is similar to the method a human would use to monitor a manufacturing process by using his own multiple senses, and processing the information about a variety of state variables that characterize the process. Since measurement of process variables is performed by several sensing devices, i.e., more sensor-based information is considered, the uncertainty and randomness involved with the process measurement may be drastically reduced.

The two typical methods used to evaluate product quality handle information differently. One makes use of the raw signal directly, the other uses features extracted from the raw signal. In the case of using the raw signal, indicated in a dotted arrow, the amount of data can be a burden on tasks for clustering and pattern recognition. On the other hand, the feature extraction method is very popular, since it allows analysis of data in lower dimensional space and provides efficiency and accuracy in monitoring. Usually, the features are composed of the compressed data due to the reduction of dimensionality, which is postulated to be much smaller than the dimensionality of the data space. The feature values used could be of entirely different properties, depending upon monitoring applications. For example, in most industrial inspection problems adopting machine vision technique, image features such as area, center of gravity, periphery, and moment of inertia of the object image are frequently used to characterize the shapes of the object under inspection. In some complicated problems, the number of features used has to be as many as 20 in order to achieve successful problem solution. On the contrary, in some simple problems one single feature may suffice to characterize the object. Monitoring the machine conditions frequently employs time and frequency domain features of the sensor signal such as mean variance, kurtosis, crest factor, skewness, and power in a specified frequency band.

The selection of features is often not an easy task and needs an expert to work with characteristics of the signal/data. Furthermore, computation of feature values may often constitute a rather cumbersome

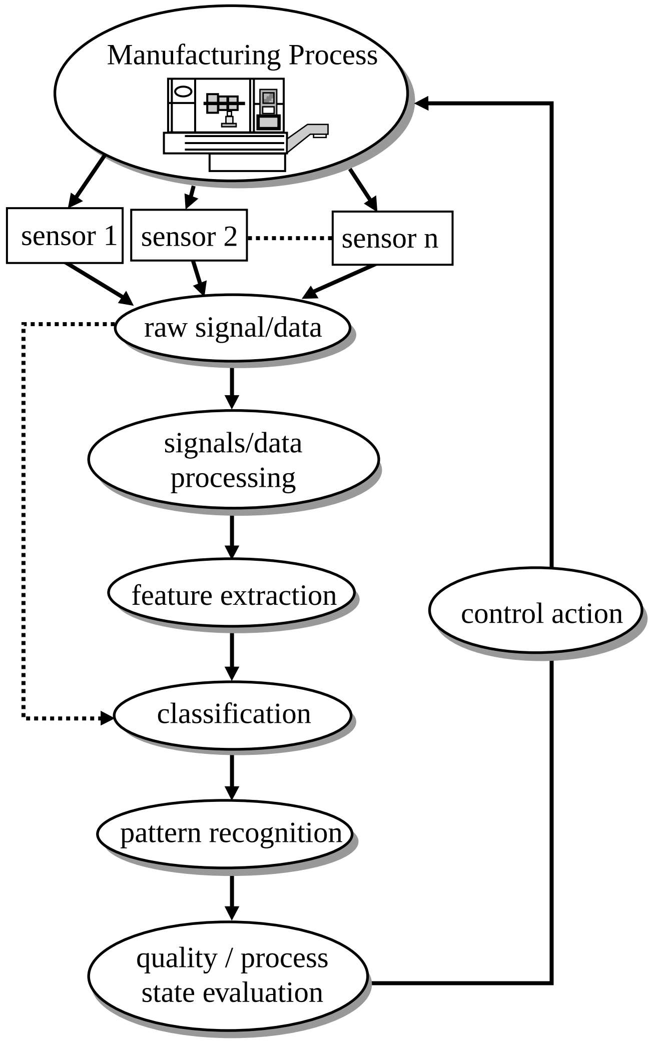


FIGURE 12.1 A general procedure for quality monitoring.

task. It is therefore important to obtain features that shows high sensitivity to product quality or a quality-related process variable and low sensitivity to process variation and uncertainty. It is equally important to obtain the fewest but the best combination of features in order to reduce the computational burden and increase efficiency in clustering. This can ensure better performance of the monitoring system, while reducing the monitoring cost.

When the choice of features is appropriately made, and their values are calculated, the next task is to find the similarity between the feature vector and the quality variables or process conditions, that is, to perform the classification task. If the feature vector is denoted by *x*, finding the similarity mathematically is to find the relationship *R*;

$$R{:}i(\underset{\sim}{x}) \;\rightarrow C \;(C = 1 \text{ or } 2, \text{ or } \ldots \text{ or, } m) \qquad \text{Equation (12.1)}$$

where *C* denotes the number assigned specifically to a class category and has *m* categories of classification. In the above equation, the category number *C* is assumed to be preassigned to represent the quality variables or process conditions. The operator i that yields the relationship expressed in Equation 12.1 is called the classifier.

A large number of classifiers have been developed for many classification problems. Depending upon the nature of the problem, the classifier needs to differ in its discriminating characteristics, since there is no universal classifier that can be effectively used for a large class of problems. In fact, it is observed from the literature that a specific method works for a specific application. Frequently used conventional classifiers include K-nearest mean, minimum distance, and the Bayes approach. This topic will be revisited in detail.

There are several important factors that affect classification accuracy, including the distribution characteristics of data in feature data space, and the degree of similarity between patterns. The set of extracted features yields the sets of pattern vectors to the classifier, and the vector components then are represented as the classifier input. The pattern vectors thus formed must be separable enough to discriminate each pattern that uniquely belongs to the corresponding category. This implies that we compute feature transformation such that the spread of each class in the output feature space is maximized. Therefore, the classifier should be designed in such a way that the designed methodology is insensitive to the influence of the above factors.

12.2.2 Manufacturing Process Control

Most of manufacturing processes suffer from the drawback that their operating parameters are usually preset with no provision for on-line adjustments. The preset values should be adjusted when process parameters are subject to change and external disturbances are present, as is usually the case in manufacturing process. As discussed previously, the manufacturing process is time-varying, highly nonlinear, complex, and of uncertain nature. Unlike nonlinearity and complexity, variability and uncertainty can be decreased if they are the result of some seemingly controllable factors such as incorrect machine setting, inconsistent material dimension and composition, miscalibration, and degradation of process machine equipment. Reducing the effect of these factors would improve process conditions, and therefore product quality. However, these controllable factors usually cannot be measurable in an on-line manner, and thus these effects cannot be easily estimated. This situation requires on-line adjustment or control of the operating parameters in response to the environment change, which in turn needs reliable, accurate models of the processes. This is due to the fact that, unless the process characteristics are exactly known, the performance of a control system that was designed based on such uncertainty may not be guaranteed to a satisfactory level.

A general feedback control system consists of a controller, an actuator, a sensor, and a feedback element that feeds the measured process signal to the controller. The role of the controller is to adjust its command signal depending upon the error characteristics. Therefore, performance of the controller significantly affects the overall performance of the control system for the manufacturing process. Equally important is the performance of the actuator and sensor to be used for control. Unless these are suitably designed or selected, the control performance would not be guaranteed, even though the controller was designed in a manner best reflecting the process characteristics.

For controller design of the manufacturing process, the greatest difficulty is that an accurate model of the process dynamics often does not exist. Lack of the physical models makes the design of a process controller difficult, and it is virtually impractical to use the conventional control methodologies. In this situation, these are two widely accepted methods of designing process controllers. One is to approximate the exact mathematical model dynamics by making some assumptions involved with the process mechanism and phenomena. As shown in Figure 12.2, the process model thus approximately obtained can be utilized for the design of the conventional controllers, which include all the model-based control schemes such as adaptive, optimal, predictive, robust, and time-delay control, etc. The advantage of the approach using the model dynamics is that the analytical method in design is possible by enabling us to investigate the effects of the design parameters. The disadvantage is that the control performance may not be satisfactory when compared with the desirable performance of the ideal case, since the controller is designed based upon an approximate model. Furthermore, when changes in the process characteristics occur with time, the designed controller may be further deteriorated.

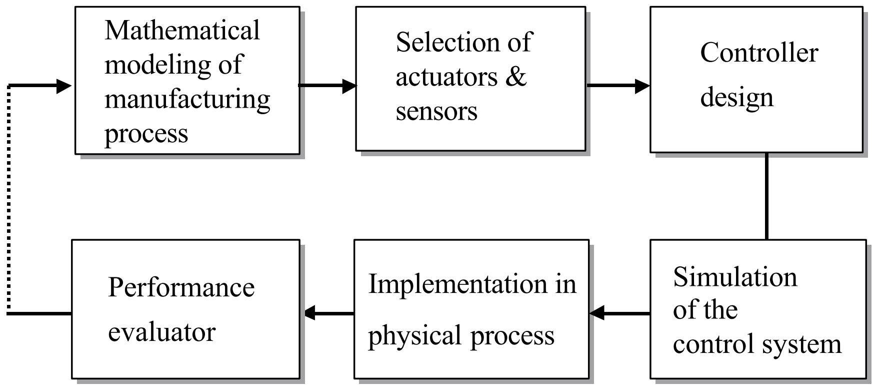


FIGURE 12.2 A feedback procedure for the design of a process controller.

The other widely accepted approach is based on an experimental trial-and-error method that uses heuristics of human operators rather than a mathematically based algorithm. In this case, human operators design the controller, making use of their own knowledge and past experience on the control action based upon observation of dynamic characteristics. The control actions of a human operator are generated from the inference of rules from which he formulates his knowledge. Accordingly, the performance of the control largely depends upon how broad and deep his knowledge of the process dynamic characteristic is and how well he can construct the appropriate rule base utilizing his knowledge and experience. As can be perceived, reliable control performance may not be guaranteed with a human operator's observation and experience alone, when the characteristics of manufacturing processes are uncertain and time-varying in nature.

12.3 Neural Network-Based Monitoring

In the previous sections we noted that monitoring requires identification or estimation of the characteristic changes of a manufacturing process based on the evaluation of a process signature without interrupting normal operations. In doing so, a series of tasks is performed, such as signal processing, feature extraction, feature selection and integration, classification, and pattern recognition. In some cases, a complete process model describing the functional relationship between process variables must be extracted. Some typical problems that arise in the conventional monitoring task may be listed as follows:

- Inability to learn and self-organize signals or data
- Inefficiency in solving complex problems
- Robustness problem in the presence of noise
- Inefficiency in handling the large amount of signals/data required

In any process, disturbances of some type arise during manufacturing. For example, in welding processes, there are usually some variation in incoming material and material thickness, variation in the wire feeder, variation in gas content, and variation in the operating conditions such as weld voltage and current. When some of these process parameter changes occur, the result is variation in monitoring signal. This situation requires adaptation of the monitoring strategy, signal processing, and feature extraction and selection by analyzing the changes in the signature of the incoming signal and incoming information on the observed phenomena. The conventional method, however, cannot effectively respond to these changing, real process variations. In contrast to this, a neural network has the capability of testing and selecting the best configuration of standard sensors and signal processing methods. In addition, it has a learning capability that can adapt and digest changes in the process.

Normally, it is not easy to directly measure product quality from sensors, as mentioned previously. Indirectly measuring a single measurement may suffice to give some correlation to the quality. However, the relationship between the quality variables and the measured variables is normally quite complex, being also subjected to the dependency of some other parameters. Furthermore, in some other cases, single sensor measurement may not provide a good solution, and thus multiple measurements may be required. This situation calls for a neural network role that has the capability to self-organize signals or data and fuse them together.

A robustness problem in the presence of signal noise and process noise is one of the major obstacles to achieving high quality in monitoring performance. In general, process noise has either long-term or short-term characteristics. For instance, in machining processes, if vibration from the ground is coming into the machine processing the materials, and lasts continuously for some time, it can be said to be a long-term noise. If it continues only for a short time and intermittently, it may be regarded as a short-term noise. A neural network can handle the short-term noise without difficulty due to its generalization characteristics; it provides monitoring performance that is almost immune to the process noise. Such a neural network easily takes the roles of association, mapping, and filtering of the incoming information on the observed phenomena.

Finally, a monitoring task requires a tremendous amount of signal/data to process. Handling this large volume of data is not a difficult task for the neural network, since it possesses the capability of a high-speed paralleled computation. And, if necessary, it has the ability to compress the data in an appropriate way.

The foregoing discussions imply that the role of networks is to provide generality, robustness, and reliability to the monitoring. When they are embedded in the monitoring system, the system is expected to work better, especially under operating conditions with uncertainty and noise. Even in such conditions the embedded system should be able to effectively extract feature of the measured signals, test and select the extracted features and, if necessary, integrate them to obtain better correlation to the quality-related variables. In addition, it should effectively classify the collected patterns and recognize each pattern to identify the quality variables.

The neural networks often used for monitoring and control purpose are shown in Figure 12.3. In this figure, the neural networks are classified in terms of the learning paradigm. These different types of the networks are used according to domain of problem characteristics and application area. Specifically, problem characteristics to be considered include ability of on-line monitoring, time limitation of classification and recognition robustness to uncertainty, and range of process operations. Even if one classifier works well in some problem and/or application area, it may not be effectively applied to some others because any single network does not process general functionality that can handle all types of complexity involved with the processes. For this reason, integration of two or more networks has become popular in monitoring and control of manufacturing processes.

12.3.1 Feature Selection Method

This issue concerns relating the feature vector to classification and recognition. Important input features can be selected in various ways within the neural network domain. The method introduced herein is based upon a multilayer perceptron with sigmoidal nonlinearity whose structure is shown in Figure 12.4(a).

The first method [Sokolowski and Kosmol, 1996] utilizes the concept of *weight pruning*, which can determine the importance of each input feature. The method starts with selection of a certain weight of an already trained network. This selected weight is then set to zero while the network processes a complete set of input feature vectors. Due to this change, the error will occur as follows:

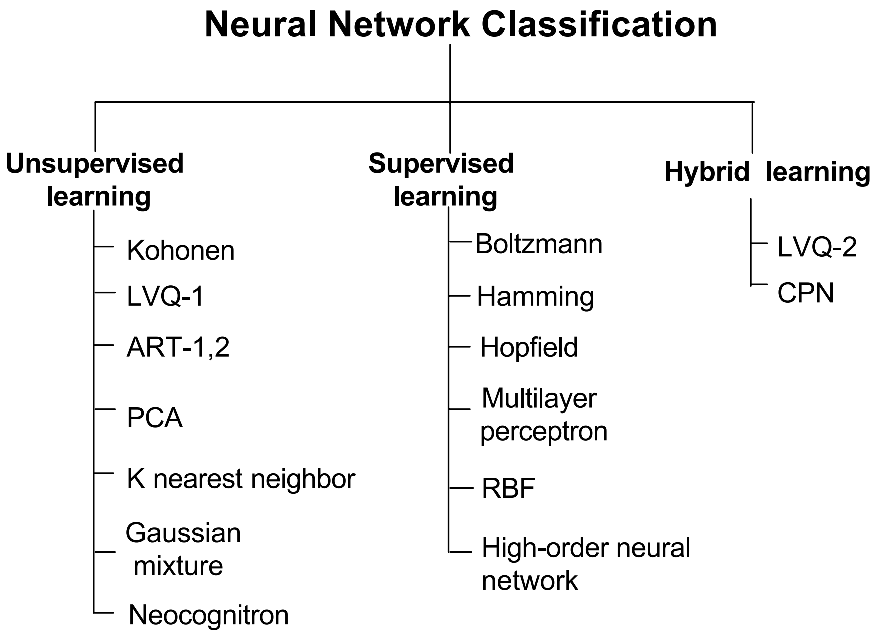


FIGURE 12.3 The neural networks frequently used for classifiers, identifiers, and controllers.

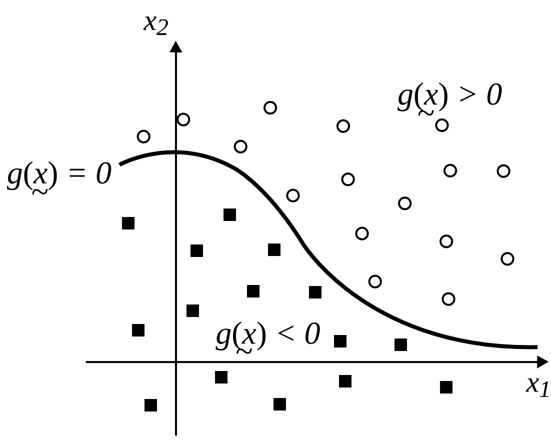


FIGURE 12.4 A discriminant function defined in a two-dimensional space.

$$E_s \max \| d_k^j - o_k^j \| \text{ for } k = 1, 2, \ldots M, j = 1, 2 \ldots, N \qquad \text{Equation (12.2)}$$

where the subscript j refers to the j^{th} input vector, d_k^j is the desired output of the k^{th} neuron in the output layer, is the actual output of the k^{th} neuron for the j^{th} input vector, M is the number of the output neurons, and N is the number of input vectors.

If the error does not exceed a prescribed maximum value E_s, the contribution of the weight omitted in the calculation to obtain the actual output is considered to be less important.

For each weight that satisfies E_k E_s the following total RMS error is calculated by

$$E_T = \frac{1}{NM}\sqrt{\sum_{k=1}^{M}\sum_{j=1}^{N}\left(d_k^j - o_k^j\right)^2} \qquad \text{Equation (12.3)}$$

After checking this weight its previous value is restored and another weight is tested. The procedure of weight pruning continues until the elimination of a weight leads to E_s error above the prescribed value.

The second method is referred to as weight sum method [Zurada, 1992]. In this method, the sensitivity of each input feature to total error is evaluated based on the sum of absolute values of the weight, which is defined by

$$\left\|w_{kj}\right\| = \sum_{k=1}^{M}\left|w_{kj}\right| \qquad (j=1, 2 \ldots, N) \qquad \text{Equation (12.4)}$$

where j is the j^{th} input feature, and w_{kj} is the weight parameter related to the j^{th} input. If the sum of the weight values $||w_{kj}||$ is below a prescribed value, the input can be discarded from further consideration, implying that the important input features can be removed.

12.3.2 Classification Method

With an appropriate set of input feature thus selected, the next task in monitoring is to perform classification and recognition. The goal of pattern classification is to assign input patterns to partition the multidimensional space spanned by the selected features into decision regions that indicate to which any belongs. Good classification performance therefore requires selection of an effective classifier, e.g., a type of neural network, in addition to selection of effective features. The network should be able to make good use of the selected features with limited training, memory, and computing power.

Figure 12.3 summarizes various types of neural networks popularly used for pattern classification. The Hopfield net, Hamming net, and Carpenter–Grossberg classifier have been developed for binary input classification, while the perceptron, Kohonen self-organizing feature maps, and radial basis function network have been developed for analog inputs. The training methods used with these neural networks include supervised learning, unsupervised learning, and hybrid learning (unsupervised + supervised). In the supervised classifier the desired class for given data is provided by a teacher. If an error in assigning correct classification occurs, the error can be used to adjust weight so that the error decreases. The multilayer perceptron and radial basis function classifiers are typical of this supervised learning.

In learning without supervision, the desired class is not known *a priori*, thus explicit error information cannot be used to adjust network behavior. This means that the network must discover for itself dissimilarity between patterns based upon observation of the characteristics of input patterns. The unsupervised learning classifiers include the Kohonen feature map, learning vector quantizer with a single layer and ART-1 and ART-2. Classifiers that employ unsupervised/supervised learning first form clusters by using unsupervised learning with unlabeled input patterns and then assign labels to the cluster using a small amount of training input patterns in the supervised manner. The supervised learning corrects the sizes and locations of the cluster to yield an accurate classification. The primary advantage of this classifier is that it can alleviate the effort needed to collect input data by requiring a small amount of training data. The classifiers that belong to this group are the learning vector quantizer (LVQ1 and 2) and feature map.

The role of the neural network classifiers is to characterize the decision boundaries by the computing elements or neurons. Lippmann [1989] divided various neural network classifiers into four broad groups according to the characteristics of decision boundaries made by neural network classifiers. The first group is based on *probabilistic distributions* such as probabilistic or Bayesian classifiers. These types of neural networks can learn to estimate probabilistic distributions such as Gaussian or Gaussian mixture distributions by using supervised learning. The second group is classifiers with *hyper-plane* decision boundaries. Nodes form a weighted sum of the inputs and pass this sum through a sigmoid nonlinearity. The group includes multilayer perceptrons, Boltzmann machines, and high-order nets. The third group has complex boundaries that are created from *kernel function* nodes that form overlapping receptive fields. Kernel function nodes use a kernel function, as shown in the figure, which provides the strongest output when the input is near a node's centroid. Kernel function indicates that the node output peaks when the input is near the centroid of the node and then falls off monotonically as the *Euclidean distance* between input and the centroid of a node increases. Classifications are made by high-level nodes that form functions from weighted sums of outputs of kernel function nodes. These type of neural network classifiers are based on the cerebellar model articulation controller (CMAC), radial basis function classifier. The fourth group is *exemplar classifiers*, which perform classification based on the identity of the training examples, or exemplars, that are nearest to the input, similar to the kernel function nodes. Exemplar nodes compute the weighted Euclidean distance between inputs and node centroids. Centroids correspond to previously given labeled training examples or to cluster center and called prototypes. These classifiers includes k-nearest neighbor classifiers, the feature map classifiers, the learning vector quantizer (LVQ), restricted coulomb energy (RCE) classifiers, and adaptive resonance theory (ART). These four group classifiers provide similar low error rate. But their characteristics for real world problems are different.

Let us illustrate the role of the neural network classifier in classification by illustrating a basic classification problem. Suppose that the input components of a classifier are denoted by an *n*-dimensional vector *x*. This then can be represented by a point in *n*-dimensional Euclidean space E'' called pattern space. An illustration is presented for the case of two-dimensional spaces, $n = 2$, in Figure 12.4.

In the figure, *g(x)* is called the *discriminant function*, which can discriminate the decision boundary. The function *g(x)* shown here is not a straight line dividing the pattern space, and represents an arbitrary curved line. This problem is called a n*onlinearly separable classification problem*. The pattern *x* belongs to the i^{th} category if and only if

$$g_i(\underset{\sim}{x}) > g_j(\underset{\sim}{x});\ i, j = 1, 2\ (i \neq j) \qquad \text{Equation (12.5)}$$

Therefore, within the region, the *i*th discriminant function will have the largest value. When a monitoring problem is complex and highly nonlinear, adaptive *nonparametric* neural network *classifiers* have an advantage over the conventional methodologies. They take role of determining the decision surface $g(\underset{\sim}{x})$ in multidimensional space defined by the input feature vectors.

Determining the function depends upon which classifier is used, and which domain of the training data is considered for classification. Depending upon the problem characteristics and domain, the classifier, its structure, and the learning algorithm need to be carefully chosen. Once these are chosen, the next task is to provide the network with the capability of good classifications. To design such a classifier, the development of neural network classifiers must go through two major phases: training phase and test phase.

12.4 Quality Monitoring Applications

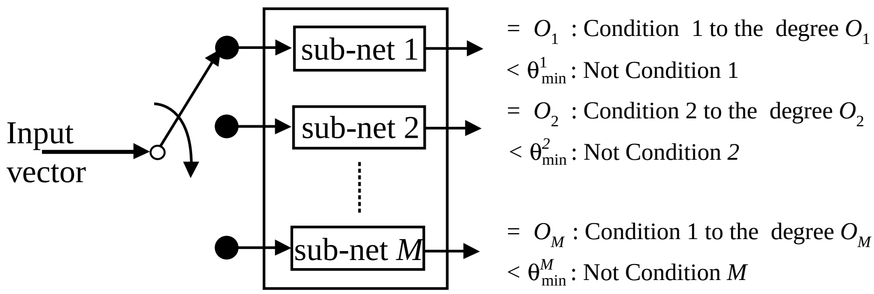
There have been tremendous research efforts in monitoring of manufacturing processes. A majority of the research deals with tool condition, machine process condition, and fault detection and diagnosis, which are not directly related to product quality. In contrast, not many studies deal with product quality monitoring. This is mainly due to the fact that direct quality monitoring is extremely difficult and that even correlating the process variables with it is not an easy task. Table 12.1 summarizes types of quality monitoring in various manufacturing processes, including types of sensor signals, neural networks, and quality variables used for monitoring. Some neural network applications will be summarized below for turning end milling, grinding, and spot welding processes.

TABLE 12.1 Types of Sensor Signals and Neural Networks in Monitoring

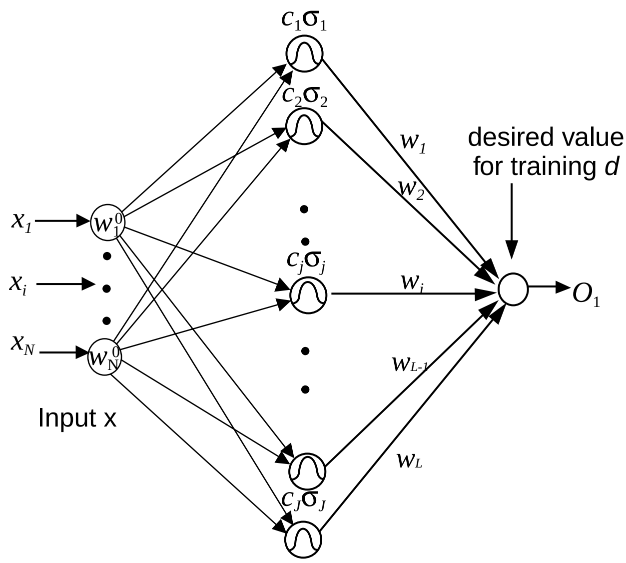
Process	Sensor Signal	Neural Network	Quality Variable
Turning	Force	Perceptron	Surface finish
	Diffraction image	Perceptron	Surface finish
Grinding	Image	RBF	Surface finish
	Wheel velocity grinding, depth	Perceptron	Grinding burn
	AE	RBF	Surface finish
Milling	Acoustic wave, spindle variation, cutting force	Perceptron	Surface finish, bore tolerance
Spot welding	Weld resistance	Perceptron	Weld nugget geometry
	Current	Perceptron	Quality factor
	Electrode force	LVQ	Strength, indentation
Arc welding	Weld current	Perceptron	Weld pool geometry
	Acoustic wave	Perceptron	Acceptability of weld
	Temperature	Perceptron	Weld pool geometry
	Vision image	Perceptron	Weld pool geometry
	Welding current, arc voltage	Perceptron	Weld pool geometry
	Ultrasonic sound	Perceptron	Weld defects (void, crack)
Pipe welding	Optical image	Kohonen	Weld pool geometry
PCB solder joint	CCD image	LVQ	Surface dimension
IC fabrication	Chamber pressure	Perceptron	Plasma etching fault detection
	DC bias, reflected RF power	Perceptron	Oxide thickness
	Etch time, gas flow rate, RF power pressure	Perceptron	Oxide thickness
Autoclave curing	Pressure, the 1st and 2nd holding temperature	Perceptron	Laminate thickness, void size
Web	CCD image	LVQ2	Surface roughness
Steel casting	Temperature	Time-series and spatial	Breakout
Steel types inspection	Vision (capture of spark)	Perceptron	Steel types
Metal forging	Ram load and velocity	Perceptron	Final shape, microscopic properties
Wire EDM	Pulse width, wire tension	Perceptron	Surface roughness
Color printing	Color	Perceptron	Desired color codes
Light wave inspection	CCD image	Perceptron, counter-propagation	Light ware defect
Tapping	Cutting force	Perceptron, RBF	Thread quality
Riveting	AE	Perceptron, Kohonen	Crack growth
Laser surface hardening	Temperature	Perceptron	Layer thickness

12.4.1 Tapping Process

Tapping is an important machining process that produces internal threads and requires relatively low cutting speed and effective cooling. Several malfunctions may occur in the process, including tap-hole misalignment, tap wear, tap-hole mismatch. With such malfunctions, the machine produces threads of undesirable quality such as hole undersize, hole oversize, eccentricity of the hole, and so on. To monitor



(a) The proposed neural architecture



(b) Information-gain-weighted RBF as the sub-net

FIGURE 12.5 A neural network schematic proposed for tapping process monitoring. (a) The proposed neural architecture. (b) Information-gain-weighted RBF as the sub-net.

these conditions, a network-based monitoring system [Chen et al., 1996] has been developed. This system utilizes a dynamometer that measures tapping torque, thrust force, and lateral force. The network used here is composed of *M* subnetworks, as shown in Figure 12.5(a), where *M* denotes the number of categories to be classified. Each subnetwork is essentially the information-gain-weighted radial basis function, which accepts an input vector ($x_1, x_2, \ldots, x_8$) and produces one output. The input vector composed of eight nodes of the input layer of the RBF is extracted from the dynamometer, including x_1 = peak of torque, x_2 = mean of torque, x_3 = variance of torque, x_4 = mean of torque in retraction stroke, x_5 = mean of thrust force, x_6 = covariance of torque and force, x_7 = correlation of torque and thrust force, x_8 = correlation of torque and thrust force in retraction stroke. The w_i° assigned to each node represents the information gain-weighted value, which is learned based on information available at the signal/index evaluation stage. The information gains can always be updated when new data are available. The weight parameters w_i° ($i = 1, 2, \ldots, N$) are calculated from entropy theory. According to the results given in the reference, the total information gain of the index x_k about the process is obtained by

$$G_\Omega(x_k) = \sum_{i=1}^{R} G_{c_i}(x_k) \qquad \text{Equation (12.6)}$$

In the above equation, $Gc_i(x_k)$ indicates the information gain of the index x_k about the class c_i, $\Omega = \{c_j;\ i = 1,2, \ldots, R\}$ is the class space with c_i representing the i^{th} class and R the total number of classes. Equation 12.6 essentially implies that the larger the $G_\Omega(x_k)$ is, the more significant the index x_k is to the tapping process.

The classification is obtained by the proposed RBF when the minimum threshold θ_{min} of each sub-network is set to 0.2 based on statistical distribution obtained during training. These results are compared with the conventional RBF. The results indicate that, in the case of undersize and misalignment classes, the proposed system discerns the patterns much more clearly than the conventional one.

12.4.2 Solder Joint Monitoring

Solder joints of printed circuit board have various shapes as soldering conditions (amount of solder paste cream, heating condition, heat profile, etc.) are changed. In the aspect of classification problem, even though solder joints belong to a set of the same soldering quality, the shapes are not identical to each other, but vary to a certain degree. This makes it difficult to define a quantitative reference of solder joint shape as a soldering quality. In recent years, artificial neural network (ANN) approaches have been applied to solder joint inspection due to learning capability and nonlinear classification performance. Among many neural network approaches, the LVQ neural network classifier [Kim and Cho, 1995] is one example of such neural network applications for solder joint inspection.

A three-color tiered illumination system to acquire the shape of a solder joint is shown in Figure 12.6, which consists of three colored circular lamps (red, green, blue), a CCD camera, a color image processing board, and a PC with a display monitor. The lamps are coaxially tiered in the sequence of green, red, and blue upward from the bottom of the inspection surface. The three color lamps illuminate the solder joint surface with different incident angles: the blue lamp to 20°, the red lamp to 40°, the green lamp to 70°. With the help of the three colors of lamp with different incident angles, we can acquire color patterns of three different slope surfaces in an image at a time. Figure 12.6(a) shows a typical image of solder joints on the PCB under the illumination system. The color patterns of the specular surface on solder joints are generated according to surface slope.

Utilizing the acquired image of the solder joint surface, a neural architecture for solder joint inspection is adopted as shown in Figure 12.6(b). The color intensities at each pixel in the stored color image are used as the input data of the LVQ-1 neural network. An input of LVQ-1 neural network is represented by

$$\underset{\sim}{x}_c = \{x(l, i, j) \mid l = 1, 2, 3, \quad i = 1, 2, \ldots, n, \quad j = 1, 2, \ldots, m\} \qquad \text{Equation (12.7)}$$

where the subscript c indicates the class of the input data, $\underset{\sim}{x}_c\,(l, i, j)$ means an intensity value at the (i, j) pixel in the l color frame, the l is an index to represent each color frame (1 = the red, 2 = the green, 3 = blue), the (i, j) indicates a pixel position, and the (n, m) indicates the size of a window image. The dimension of input nodes is the same as that of the input image.

Output nodes are fully connected to all input nodes by the weight vectors. The number of output nodes in a competitive layer is set to 10. A set of the weight vectors between the k^{th} output node and the input nodes is defined as

$$\underset{\sim}{w}_k = \{w_k(l, i, j) \mid l = 1, 2, 3, \quad i = 1, 2, \ldots, n, \quad j = 1, 2, \ldots, m\} \qquad \text{Equation (12.8)}$$

where all subscripts are the same as those of the input image. The input value of the k^{th} output node is the Euclidean distance between the input data and the weights, which is expressed by

$$o_k = \underset{\sim}{w}_k - \underset{\sim}{x}_c = \sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{l=1}^{3}\left\| w_k(l,i,j) - x_c(l,i,j) \right\|. \qquad \text{Equation (12.9)}$$

In the self-clustering module, the input value of the output node is a similarity measure for clustering that indicates the degree of resemblance between an input image and the weight vectors. The update of weight vectors are based on competitive learning, called *winner-take-all learning*. Only one node of the nearest weights to the current images is selected as the winner and has a chance to update its weight vector by

$$\underset{\sim}{w}_{win}(t+1) = \underset{\sim}{w}_{win}(t) + \eta(t)\left(\underset{\sim}{x}_c(t) - \underset{\sim}{w}_{win}(t)\right) \qquad \text{Equation (12.10)}$$

where $\eta(t)$ is the learning rate, and is initially set to 0.25 and decreases to 0.05 as t increases. During the training procedure, the weights of each output node are to be updated toward resembling the members in its own cluster. After the training procedure, a supervised learning technique is adopted to increase classification performance. Only if an input data is misclassified into a different class should the weight vector of the class be updated. For example, consider an input data, x_q belonging to q^{th} class. If it is misclassified into the cluster labeled as the c^{th} class, then update the synaptic weights of the cluster as follows:

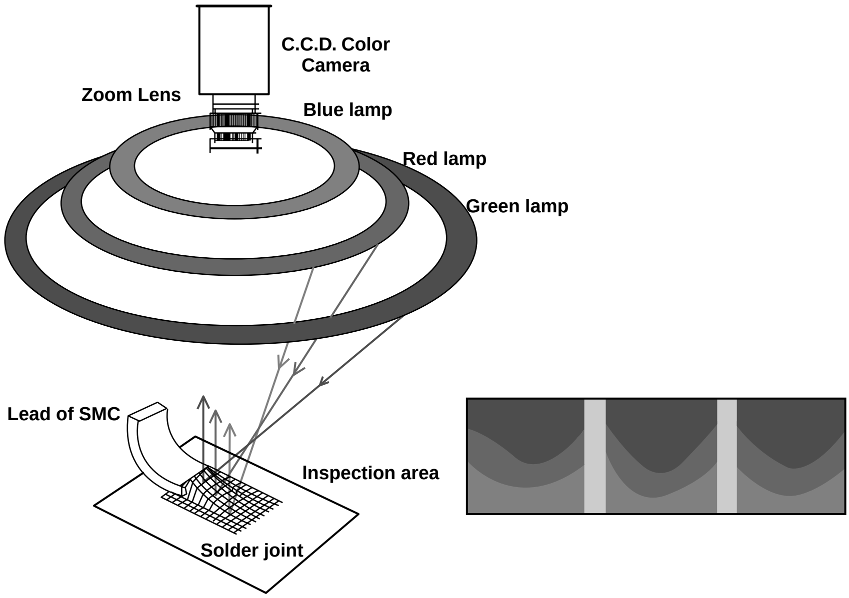
$$\underset{\sim}{w}_{win}(t+1) = \underset{\sim}{w}_{win}(t) + \eta(t)\left(\underset{\sim}{x}_q(t) - \underset{\sim}{w}_{win}(t)\right). \qquad \text{Equation (12.11)}$$

To evaluate the performance of the proposed neural network classifier, the classification is made for the test data that are not used for the training. The clustering result is compared to that of the expert inspector. The total classification success rate is found to be 93.1%. The proposed neural network has a simple structure that facilitates the use of raw intensity data of each pixel as its input data. It relieves the burden of performing a large number of experiments to find optimal visual features, and will also help to initially find the input feature space before designing a suitable classifier.

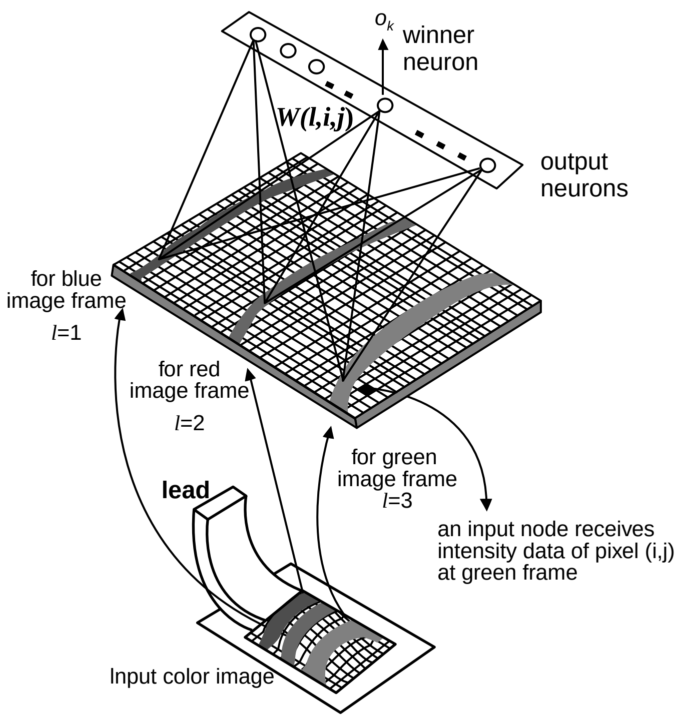
12.4.3 Pipe Welding Process

In a high-frequency electric resistance pipe welding process, the hot roll coils are progressively formed into cylinder shapes in several stages while high-frequency current is applied with contact tips to both edges of the formed metal to be joined. Applied current flowing between the adjacent surfaces of the edges metal heats and melts a small volume of metal along the edge by making the best use of the skin effect and proximity effects of high-frequency currents. When the molten metal from both adjacent edges runs together by the action of the squeeze rolls and cools down by cold water, a weld is produced.

The three typical shapes of bead in high-frequency electric resistance welding (HERW) are shown in Figure 12.7(a). Under insufficient heat input, the concave shape appears on the top bead while the slope of the bead is steep. The concave shape is produced when the molten metal diminishes between the base metal due to the squeezing force. A cold weld is a main defect from the lack of heat input. Under an optimum heat input condition, the concave shape disappears. In this case, the molten steel appears on the top of the bead and the smooth shape of the bead can be achieved. Under an excessive heat condition, the molten steel hangs over the top of the bead and the height of the bead becomes unstable while the bottom width of the bead increases with heat input. Penetration is one of the main defects from excessive heat input due to the inclusion of impure particles in the weld pool.

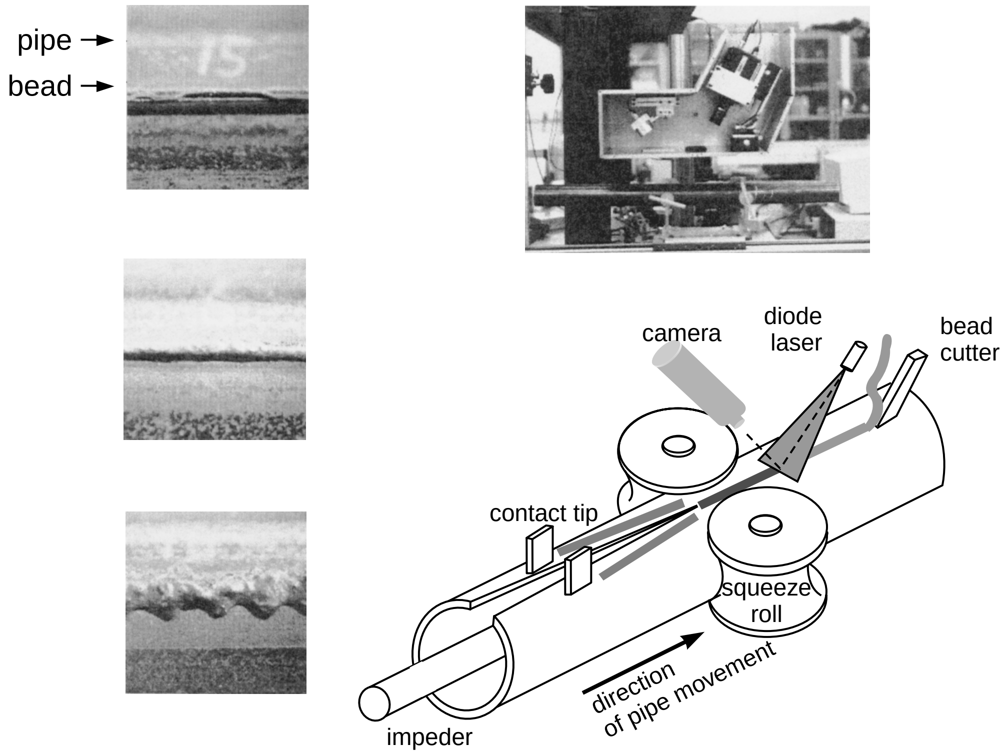


(a) Images of solder joint and 3-color ring illumination system



(b) The neural structure of LVQ for solder joint inspections

FIGURE 12.6 The solder joint image and LVQ architecture. (a) Images of solder joint and three-color ring illumination system. (b) The neural structure of LVQ for solder joint inspections.



(a) Bead shape images (b) Visual monitoring system

FIGURE 12.7 Bead shape image and a visual monitoring system in a high-frequency electric resistance welding process. (a) Bead shape images. (b) Visual monitoring system.

A visual bead shape monitoring system [Ko et al., 1994], as shown in Figure 12.7(b), was designed to acquire an image of the bead shape. It consists of a CID (charge injection device) camera, a laser with a cylindrical lens, fine mechanical alignment stage sets, filters, air spray, and cooling system. The monitoring system operates on a principle of triangulation that has been used to obtain visual information on the layout of bead shape. The Kohonen SOFM for bead shape classification is shown in Figure 12.8. The input layer and competitive layer consist of 180 nodes which corresponds to horizontal pixel and 12 nodes in the form of one dimensional array, respectively.

During the training procedure, the weight vectors of the network form a model of the input pattern space in terms of so-called prototypical feature patterns. A Kohonen network can segment an input bead image similar to the training pattern by comparing it with all trained weight vectors. For this classification procedure, a winning weight vector $\underset{\sim}{w}_{win}$ that is close to a given input pattern x is selected by

$$\|\underset{\sim}{w}_{win} - \underset{\sim}{x}\| = \min_{k} \|\underset{\sim}{w}_{k} - \underset{\sim}{x}\|, \quad k = 1, 2, \ldots, N \qquad \text{Equation (12.12)}$$

where N is the number of output node.

The weights of winning node and its neighborhood nodes are updated by

$$\underset{\sim}{w}_{win}(t+1) = \underset{\sim}{w}_{win}(t) + \Lambda_{win}(t)\eta(t)\left(\underset{\sim}{x}(t) - \underset{\sim}{w}_{win}(t)\right) \qquad \text{Equation (12.13)}$$

where $\Lambda_{win}(t)$ is the neighborhood relation and is a learning rate. The initial value of the learning rate is 0.5 and gradually decreases to 0.01 with time. The extent of the neighborhood relation is set initially

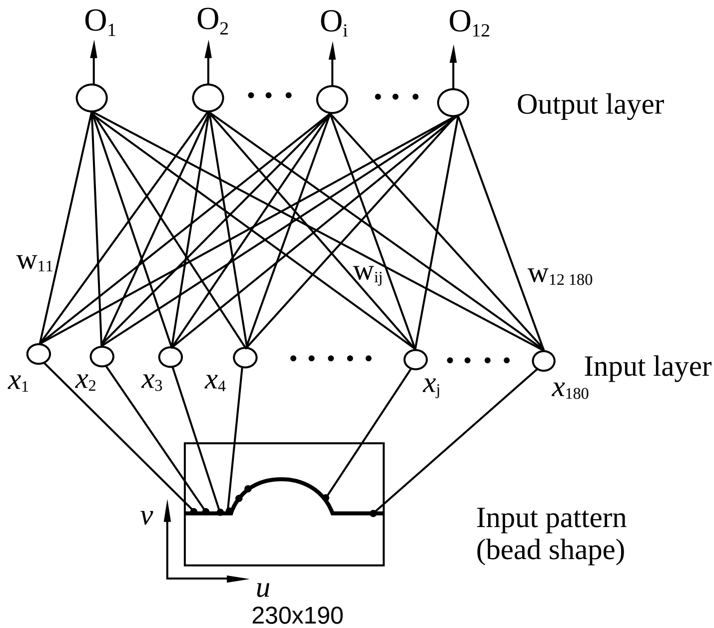


FIGURE 12.8 The NN architecture for bead shape classification.

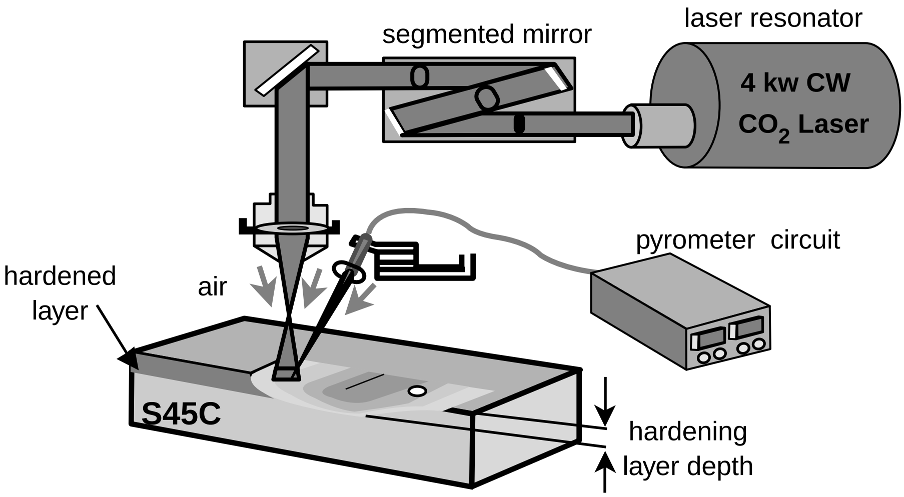
at half the output node and is made to decrease in every 1000 iterations. Total number of iterations is 10,000.

To examine classification performance of the neural network, the clustering results are compared to those of the expert operator. The classification success rate for the whole image is found to be 98.05%. These results reveal that the relationship between heat input condition and bead shape has been successfully implemented on the neural network. If the welding condition is changed, a similar neural bead shape classifier can be designed and trained using the experimentally obtained bead shape. This monitoring method faciliates adjustment of welding condition according to the current state of the bead shape. This SOFM neural network approach is more efficient for monitoring current welding conditions than the geometrical feature-based method for real HERW process, considering accuracy, experimental cost, and implementation time.

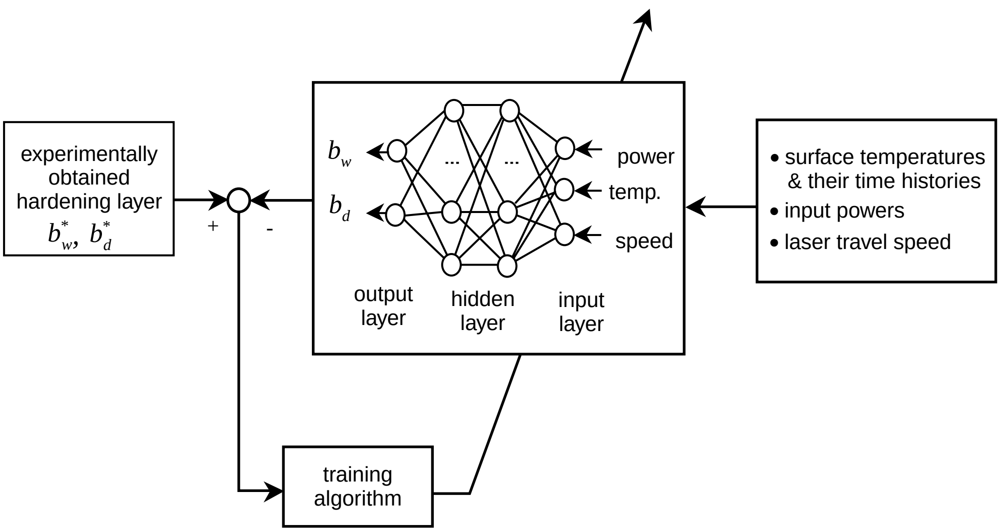
12.4.4 Laser Surface Hardening Process

A laser surface hardening process is shown in Figure 12.9(a). The surface layer is heated above its austenite transformation temperature but below the melting temperature by high-power laser beam energy traveling along a specified direction. As a result, the microstructure previously composed of ferrite and pearlite is transformed into the austenite phase. At this high temperature, the microstructure of the substrate material is homogenized by carbon diffusion. After being heated, the material is rapidly cooled, resulting in a hard martensite phase due to the self-quenching effect caused by the rapid thermal conduction from the surface into the bulk substrate material.

Due to its high energy density and power controllability, high-power laser material hardening has been widely used in industrial processes. The difficulty involved with this process is that the hardening quality, in this case layer thickness, cannot be easily assessed during the process. To assess the coating or hardening thickness during the processing, a surface temperature measurement is used, which can represent the corresponding layer thickness. Figure 12.9(b) gives a schematic of the measurement arrangement and a neural network training procedure [Woo and Cho, 1998]. The neural network used is a multilayer perceptron and it adopts the error backpropagation algorithm. The input data used for training includes surface temperatures and operating conditions such as input powers and laser travel speed, and its outputs



(a) A laser surface hardening process



(b) The layer geometry estimation using a neural network model

FIGURE 12.9 The neural network architecture. (a) A laser surface hardening process. (b) The layer geometry estimation using a neural network model.

generate the hardening depth (b_d) and width (b_w). In the error backpropagation algorithm, the weight of weights is given in an iterative manner as

$$
\begin{aligned}
w_{lk}(t+1) &= w_{lk}(t) + \eta\delta_l(t)o_l(t) + \alpha\Delta w_{lk}(t) \\
\delta_l(t) &= \left(o_l^d - o_l\right)\dot{f}_l\left(net_l(t)\right)
\end{aligned}
\qquad \text{Equation (12.14)}
$$

where w_{lk} is the weight linking the l^{th} node in output layer and k^{th} node in the hidden layer; o_l and $o_l^{\,d}$ are the output and the desired output of l^{th} node, respectively, is the activation function; $(\cdot)$ derivative of ( ); and t, η, and α are the number of training iterations, the learning rate concerned with the speed of convergence of the error, and the momentum rate to avoid the oscillating phenomena, respectively. In the above, $net_l\,(t)$ is given by

$$net_l\left(t\right) = \sum_{k=0}^{K} w_{lk}\left(t\right) o_k\left(t\right) \quad \left(l = 1,\, 2,\, \ldots,\, L\right) \qquad \text{Equation (12.15)}$$

where L is the total number of neurons in the last layer. Hidden layer weights are adjusted according to

$$\begin{aligned} &\delta_k\left(t\right) \dot{f}\left(net_k\left(t\right)\right) \sum_{l=0}^{L} \delta_l\left(t\right) \cdot w_{lk}\left(t\right) \quad \left(k = 1,\, 2,\, \ldots,\, K\right) \\ &w_{kj}\left(t+1\right) w_{kj}\left(t\right) + \eta \cdot \delta_k\left(t\right) \cdot o_j\left(t\right) \quad \left(j = 1,\, 2,\, \ldots,\, J\right) \end{aligned} \qquad \text{Equation (12.16)}$$

where J is the number of the neuron of the j^{th} preceding hidden layer, w_{kj} is the weights linking the k^{th} node of the last hidden layer and j^{th} node of the proceeding layer. In this application, learning and momentum gains are chosen to be 0.7 and 0.5, respectively. The network estimation is proven to yield fairy accurate results; the largest deviation is 10%.

12.5 Neural Network-Based Control

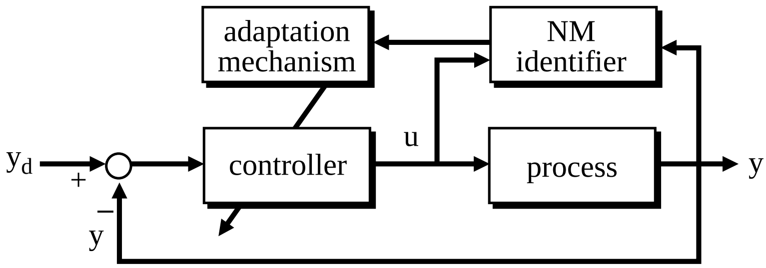
The conventional control approach lacks the ability to learn the property of the unknown system to self-organize the features of the system response, and to make an appropriate decision, based upon the learned process state, regarding how to generate the control signal so as to drive the control system to reach a designed state. The neural networks overcome the deficiency of the conventional control methodologies with their learning, self-organization, and decision-making abilities. This is why the network-based control is called one of *intelligent control*. It should be noted that "intelligent" here does not imply better system performance by intelligent control than by conventional control. In fact, for a system with its dynamic characteristics completely known there may be no need to utilize an intelligent control technique. This is because the conventional technique in this case provides better control performance and reliability in implementation.

In reality, the dynamic characteristics of most of the manufacturing processes are not exactly known. For this reason, neural networks come into play to learn the process characteristics and generate appropriate controller output based upon this learning to meet a certain specification of control performance. In carrying out these tasks the networks are embedded into the system in several forms.

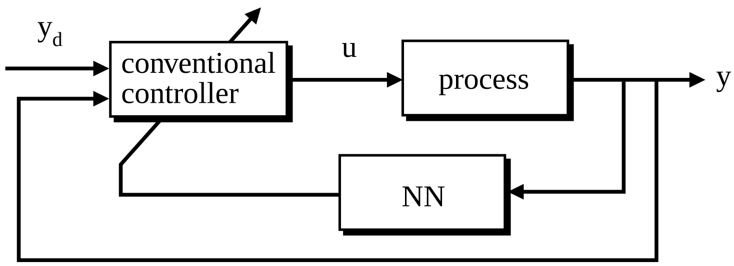
The first form is to use the networks as an aid to the conventional controller. For example, the process modeled under certain assumptions can be controlled by one of the conventional controls such as the optimal control, adaptive, and so on, as shown in Figure 12.10(a). The objective of the control is to make y follow the desired value y_d by aid of a neural network that estimates the uncertain part of the process. In this case, an approximately modeled process has to be identified on-line. The network's role here is to identify uncertain process parameters and provide the estimated parameters to the controller.

The second form is to use a neural network as a direct tuner to obtain the desired gains of the conventional controllers such as PI, PD, and PID, as shown in Figure 12.10(b). The control objective here is to self-tune the controller gain in such a way that the output y follows the desired value y. The network is trained here to minimize the error function with respect to the weight value w_{ij},

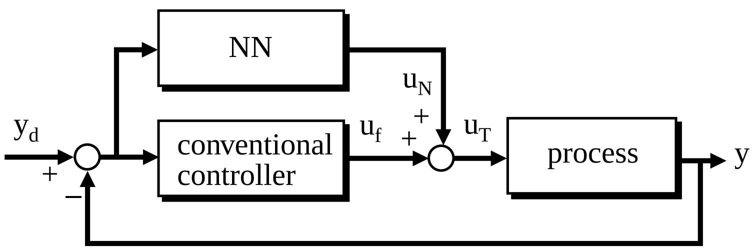
$$\frac{\partial E}{\partial w_{ij}} = f\left(\frac{\partial K}{\partial w_{ij}}\right) \qquad \text{Equation (12.17)}$$



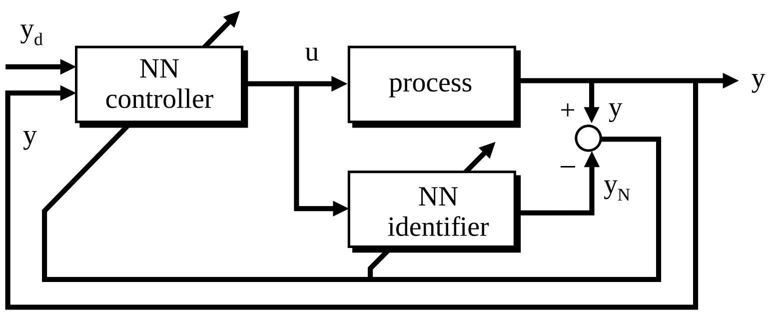
(a) A neural identifier combined with an adaptive controller



(b) A gain-tuning neural network controller



(c) A feed forward neural controller combined with a conventional feedback controller



(d) A neural controller combined with a neural identifier

Figure 12.10 Various neural network based monitoring and control schemes. (a) A neural identifier combined with an adaptive controller. (b) A gain-tuning neural network controller. (c) A feedforward neural controller combined with a conventional feedback controller. (d) A neural controller combined with a neural identifier.

where E is the squared error function defined by

$$E = \frac{1}{2}(y_d - y)^2. \qquad \text{Equation (12.18)}$$

In the above, the K consists of the gain values if a PID controller is used,

$$K = K_p, K_i, K_d \qquad \text{Equation (12.19)}$$

and the function f indicates that the gain values are the explicit functions of the network weight values w_{ij}.

The third form is a form of the neural network controller combined with a simple conventional controller, shown in Figure 12.10(c). In this control structure, the conventional controller is used to provide the network controller with stability of the system response, which may be needed at a stage of initial learning. The network is used here to learn the *inverse model* of the process, that is, the relationship that relates the desired output y_d to the corresponding control action u. In this control scheme, the neural network is so trained to minimize the squared error function E defined by

$$E = \frac{1}{2}\left(u_T - u_N\right)^2. \qquad \text{Equation (12.20)}$$

In Equation 12.20, the resultant control effort u_T consists of u_N, the control part generated by the network, and u_f, that of the feedback signal.

$$\frac{\partial E}{\partial w_{ij}} = f\left(\frac{\partial u_N}{\partial w_{ij}}, u_f\right). \qquad \text{Equation (12.21)}$$

Equation 12.21 shows the gradient of the error with respect to the weights; $\partial E/\partial w_{ij}$ is a function of the u_N/w_{ij} and u_f.

The final form, shown in Figure 12.10(d), is an efficient usage of the multiple networks in control as well as identification of the process, a neural controller and a neural identifier. In this case, learning signal and convergence are the most important factors for speed and robustness of the system response. Often, finding learning signals can be a difficult task. In the block diagram, the neural network identifier estimates the unknown process model based upon the measured output y and control input u. The network's role herein is to make the estimated process model M_N follow the actual process model M as close as possible by carrying out on-line identification of the process.

$$M_N \rightarrow M \ \ \forall\ u \in U, y \in Y \qquad \text{Equation (12.22)}$$

where u belongs to admissible operating input range U and Y is the range of the corresponding output y.

Thus, the network is trained to minimize the error function E defined by

$$E = \frac{1}{2}\left(y_N - y\right)^2$$

and $\qquad$ Equation (12.23)

$$\frac{\partial E}{\partial w_{ij}} = f\left(\frac{\partial y}{\partial w_{ij}}\right).$$

The other block representing the neural network controller is designed based upon the inverse model approach, as explained in Figure 12.10(c). That is to say, upon utilization of the identification result, the control input in the forward loop is generated such that the actual output y follows the desired value y_d as closely as possible. Thus, the network is so trained that the control error is modified in order to generate the desired output y_d,

$$E = \frac{1}{2}\left(u - u_N\right)^2 \qquad \text{Equation (12.24)}$$

and

$$\frac{\partial E}{\partial w_{ij}} = f\left(\frac{\partial u_N}{\partial w_{ij}}, \frac{\partial y}{\partial u}\right) \qquad \text{Equation (12.25)}$$

where the partial derivative y/u is included to account for generating the network training signal.

There have been tremendous research efforts in control of manufacturing processes. Table 12.2 summarizes the types of neural network control in various manufacturing processes. Some neural network applications to machining, arc welding, semiconductor fabrication, hydroforming, and hot plate rolling processes will be summarized in the following sections.

12.6 Process Control Applications

12.6.1 Machining Process

In machining, adaptive control has been viewed as a very promising strategy to adapt on-line the process parameters to widely varying process conditions. To effectively achieve this, a process model is needed for the feedback control of the process. Figure 12.11 shows a neural network based control system developed for this purpose [Azouzi and Guillot, 1996]. Two network models are used here for estimation and control of the quality variables, the dimensional deviation *DD* and the surface finish, *Ra*. The process neural model here is to provide a mathematical relationship for parameters needed to the optimizer, which minimizes the machining cost. This model estimates the *DD* and *Ra*, which implies that $\hat{x}(t) = [\hat{DD}, \hat{Ra}]$ learns the quality model output $\underset{\sim}{x}(t) = [DD, Ra]$.

In this model, a hybrid network model is adopted, which consists of a Kohonen feature map and a multilayer perceptron. The motivation of using this type of the network structure is to avoid the memory degradation due to distributed learning and process in most existing feedforward neural networks. Often, the feedforward neural model correctly represents the process behavior only in the vicinity of the most operational process input, partly forgetting the process behavior in other regions. As shown in Figure 12.12, a Kohonen network is used as a two-dimensional network tuned to a variety of input patterns through unsupervised learning. This network divides the multilayer perceptron network (MLP) into N specified clusters. Based upon this network, information flow and storage in the MLP is directed to a specified cluster. The neurons of the output layer in *P*-network accept their incoming activation signal weighted by the *K*-network. Let P_j be the weighting parameter of the j^{th} cluster (node) in the *P*-network. Then, P_j is essentially the percent contribution of the j^{th} cluster to the activation signal of the output neurons. The input to the k^{th} output neuron in the l^{th} layer is given by

$$net_k = \sum_{j=1}^{h} P_j \left\{ w_{kj}^{l-1} o_j^{l-1} \right\} \quad \left(k = 1, 2, \ldots, m\right) \qquad \text{Equation (12.26)}$$

TABLE 12.2 Types of Neural Networks in Control

Process	Control Input	Controller	Network	Quality Variable
Extrusion	Ram velocity	NN controller	Perceptron	Micro structure of the material
Semiconductor (etching)	Oxygen flow, pressure	NN controller	Perceptron	Etch rate, anisotropy
Arc welding	Heat power	NN controller	Perceptron	Weld bead width, weld depth
Steel making (galvanizing)	Temperature		RBF + perceptron	Percentage of iron at the surface
Hot plate mill	Servo valve current	NN controller + PD	RBF, perceptron	Slab width
Turning	Feed rate Cutting speed	NN controller	Perceptron + Kohonen ART 2	Surface finish, dimensional accuracy
	Feed	NN based tuning	Perceptron	Machining efficiency, surface finish
	Feed	NN controller + PD	Perceptron	Machining efficiency
	Cutting Force	NN controller		Surface finish
Plastic injection molding	Inlet flow rate	NN controller	Perceptron	Product defect
Hydroforming	Servo valve current	NN controler	CMAC	Forming dimension, wrinkling

where h is the number of the neurons in layer l-1, w_{kj}^{l-1} is the weight linking the k^{th} node in the output layer with the j^{th} node in layer l-1, and o_j^{l-1} is the output of the j^{th} node in layer l-1. The normalized P_j is estimated by

$$P_j = \frac{\exp(\lambda_j)}{\sum_{j=1}^{h} \exp(\lambda_j)} \quad (j = 1, 2, \ldots, h) \qquad \text{Equation (12.27)}$$

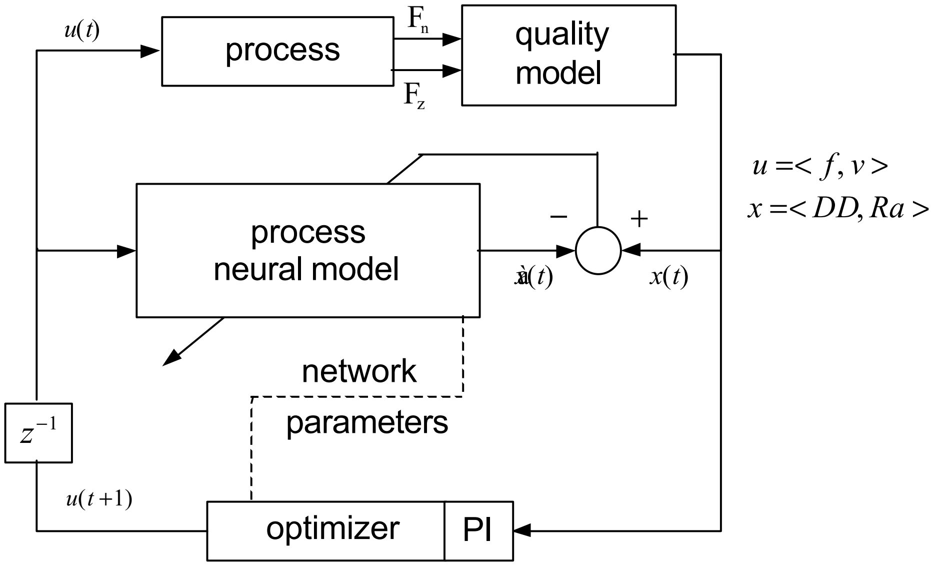
where the λ_j is given by

$$\lambda_j = \frac{\mu_1}{1 + \exp\left(\mu_2 d_j^2\right)} \qquad \text{Equation (12.28)}$$

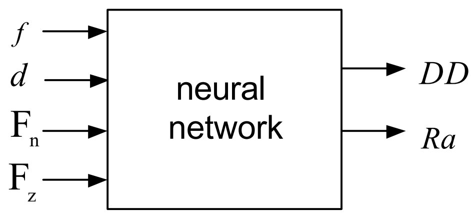
In Equation 12.28 above, d_j is the Euclidean distance between the input pattern and the weight w_j linking j^{th} node in the hidden layer with input nodes in input layer, so that

$$d_j \,\|\underset{\sim}{w}_j - \underset{\sim}{x}\| \quad j = 1, 2, \ldots, h \qquad \text{Equation (12.29)}$$

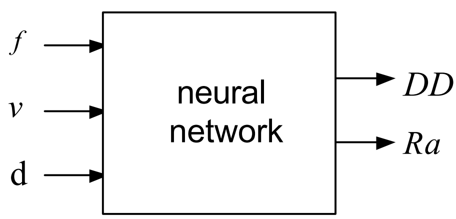
And μ_1 and μ_2 are positive constants that are used to control the relative importance of the cluster. It is noted that the winning cluster min $\|\underset{\sim}{w}_i - \underset{\sim}{x}\|$ and its neighbors are allowed to contribute significantly to the activation of the output neuron. The training of each network in this hybrid architecture is carried out using a set of special data. The Kohonen network is trained independently using a winner-take-all learning method, while the quasi-Newton method is used to determine the weights and thresholds of the MLP. Using the network architecture and the learning method described in the above, a series of simulation works was conducted for various conditions for turning operation of AISI 108 steel parts.



(a) The proposed synthesis scheme



(b) The K-P network-based process model



(c) The MLP-based quality model

FIGURE 12.11 The proposed neurocontrol scheme. (a) The proposed synthesis scheme. (b) The K-P network-based process model. (c) The MLP-based quality model.

The results are given in terms of the machining cost *L*, not in terms of dimensional deviation (*DD*) and surface finish (*Ra*). According to this work the optimized cost indicative of the *DD* and *Ra* shows 35% improvement in process performance by use of this proposed controller as compared to the well-known adaptive control with constraints (ACC).

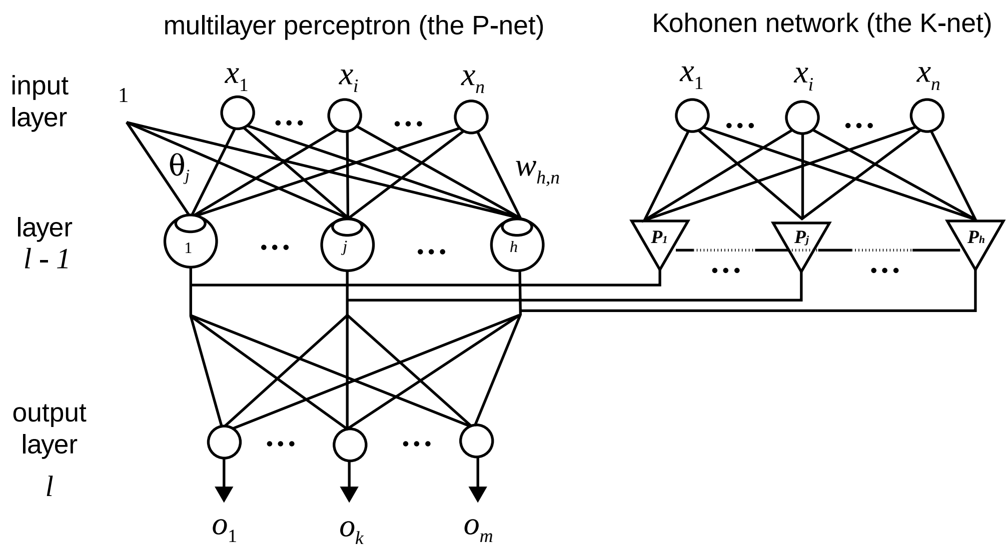


FIGURE 12.12 Schematic of the proposed K-P network.

12.6.2 Arc Welding Process

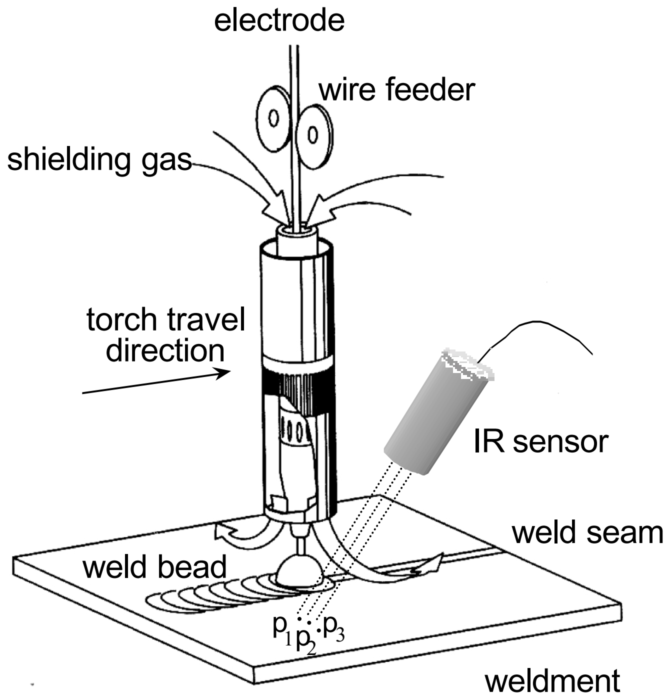
The objective of automated welding is to produce welds of high strength. To achieve this a number of research efforts have been made; one such effort is on-line control of the weld geometry such as width and penetration. The weld strength, indicative of quality, is usually represented by geometry of the weld pool, its width and penetration. These geometrical variables are not readily measurable during welding, and therefore an alternative that measures surface temperatures near the torch area has been used, since surface temperature distribution is indicative of the weld pool geometry. As shown in Figure 12.13(a), this temperature measurement is utilized to estimate an instantaneous weld pool size [Lim and Cho, 1993], which otherwise is not attainable. The control system to regulate the weld pool size is shown in Figure 12.13(b). In the figure, *PS* indicates the weld pool size and T_i denotes the temperature of the i^{th} location. The system is basically a feedback error learning adopting two neural networks, one for estimation of the weld pool size and one for a feedback forward controller. The networks are a multilayer perceptron and the error back propagation method is utilized for training these. This architecture essentially utilizes the inverse dynamics of the welding process, and therefore, the total input u_T is given by

$$u_T(t) = u_N(t) + u_f(t) \qquad \text{Equation (12.30)}$$

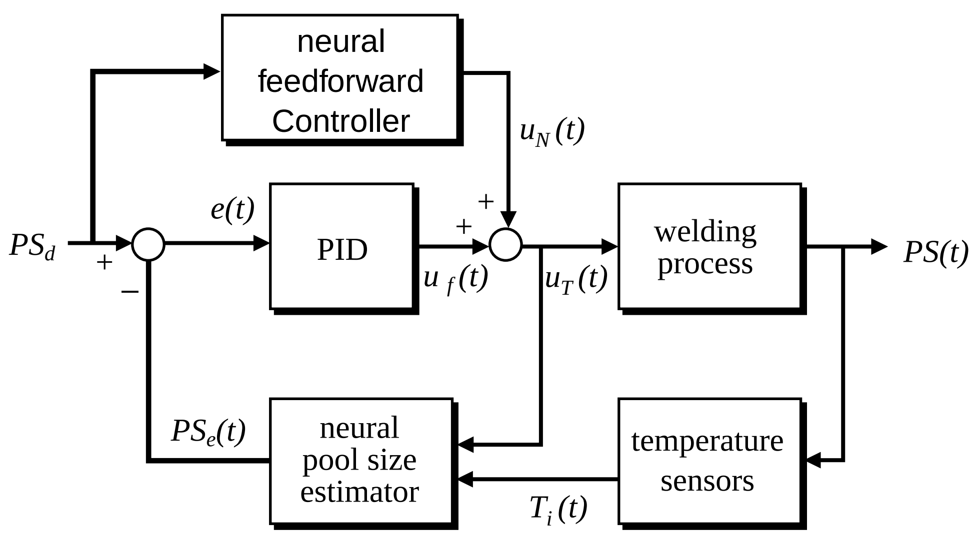
where u_N is the network generated control signal and u_f is the feedback control signal. In fact, u_f can be any of the conventional controllers. The weights of the neural network controller are corrected according to the following weight adaptation equations:

$$\begin{aligned} w_{lk}(t+1) &= w_{lk}(t) + \eta\delta_1(t)o_k(t) \\ \delta_1(t) &= u_f(t)\dot{f}(net_l(t)) \end{aligned} \qquad \text{Equation (12.31)}$$

where w_{lk} is the weight linking the l^{th} node in the output layer and the k^{th} node in the last hidden layer adjacent to the output layer, f is the activation function, $(\cdot)$ derivative of ( ). In the above, $net_l\,(t)$ is given by Equation 12.15, and hidden layer weights are adjusted by Equation 12.16.



(a) Schematic description of GMA welding process



(b) A weld pool control system

FIGURE 12.13 Temperature sensing and control system for the GMA welding process. (a) Schematic description of GMA welding process. (b) A weld pool control system.

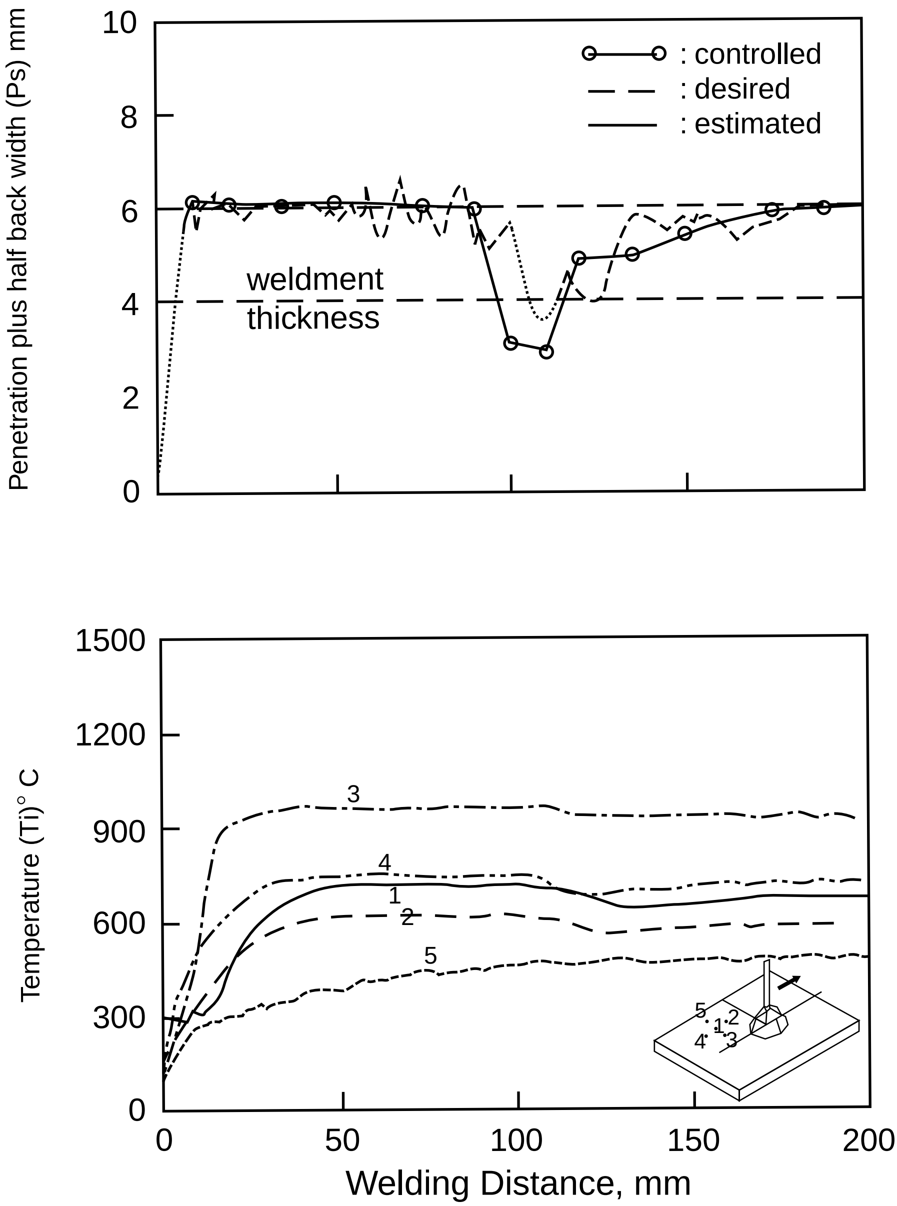


FIGURE 12.14 Neural control of a weld pool size with a neural pool.

To validate the capability of this control architecture, a series of welding experiments is performed with consideration of external disturbances. Two different network architectures are considered: $30 \times 50 \times 50 \times 1$ and $3 \times 25 \times 25 \times 1$ for the estimator and controller, respectively. An external disturbance torch travel speed is increased from 4 mm/s to 6 mm/s while the other input parameters are kept unchanged. In Figure 12.14, the experimental results of the welding control are shown together with responses of surfaces of surface temperatures at five locations. It can be seen that the pool size obtained by the neural control converges to a desired value, 6 mm. The controlled value, however, exhibits small fluctuations due to those of the estimation. The estimation results, however, denoted with a dotted line, indicate that the neural network estimates the actual pool size with satisfactory accuracy, which in this case is the controlled one.

12.6.3 Semiconductor Manufacturing

Semiconductor manufacturing processes typically exhibit complex interactions between multiple operating input signals and multiple output variables. In particular, reactive ion etching (RIE), as shown in Figure 12.15(a), is a highly nonlinear low-pressure form of plasma etching and remains a poorly understood process. The neural network control system [Stokes and May, 1996] employed here basically consists of an emulator (identifier) and an inverse neural model controller. The control system is illustrated in Figure 12.15(b). For simulation purpose, the process is assumed to be governed by a q-step ahead model described by

$$y(t+q) = f_N\left(\underset{\sim}{y}_t, \underset{\sim}{u}_{t+q-1}, \underset{\sim}{u}_{t-1}, \underset{\sim}{d}_t\right) \qquad \text{Equation (12.32)}$$

where $\underset{\sim}{y}_t, \underset{\sim}{u}_{t+q-1}, \underset{\sim}{u}_{t-1}, \underset{\sim}{d}_t$ are the sampled output, input, and disturbance vectors, respectively. To generate the control input $u(t)$, the emulator is generated for the function in the q-step ahead predictive model given by Equation 12.32. The output of the process model and the resulting difference signal is used to tune an inverted neural model contained in the neural controller, which then generates the control input to drive the process. A series of simulations of the etch rate control are conducted with the emulator neural network 4-7-1 multilayer perceptron and the controller network a 1-8-4 structure. The etch rate was observed to converge quickly to the desired values, which are changed every 30 seconds.

12.6.4 Hydroforming Process

As shown in Figure 12.16(a), the hydroforming process does not require a die to form a sheet metal product of desired shape. Instead, a forming chamber takes this role by generating a hydraulic pressure against the sheet metal to be formed according to a prescribed schedule. The objective of the control is accurate tracking of the curve, pressure vs. the punch stroke, which ensures forming of high-quality products with no defects. A pressure control system based on cerebellar model articulation control (CMAC) [Park and Cho, 1990] is shown in Figure 12.16(b). In the diagram, p_d is the desired forming pressure, p_f is the actual forming pressure of the chamber, e is the differential signal between these two, u_T is the total control input consisting of the neural controller, u_N and the feedback controller PID, u_f.

The CMAC controller output is essentially the inverse dynamics of the forming process, namely,

$$u(t) = g\left(\underset{\sim}{p}_f(t)\right) \qquad \text{Equation (12.33)}$$

where g represents the functional relationship between the u and p_f, and the chamber pressure vector at time step t consists of

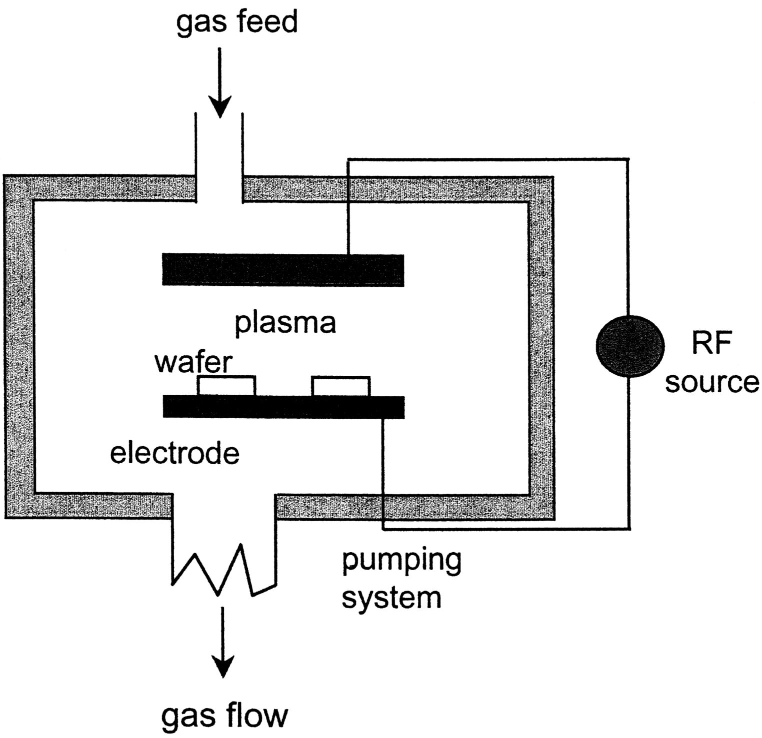
$$p_f(t) = \{p_f(k+t), \ldots, p_f(1+t), p_f(t)\}. \qquad \text{Equation (12.34)}$$

The g function is expressed by

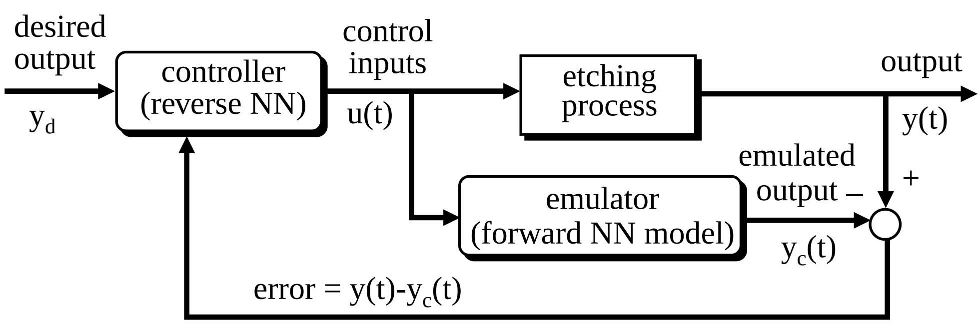
$$g\left(\underset{\sim}{p}_f\right) = \sum_{i=1}^{N} a_i \cdot w_i. \qquad \text{Equation (12.35)}$$

In Equation 12.35, a_i is the unipolar binary value and the weight w_i is given by

$$w_i(t+1) = w_i(t) + \Delta w_i \quad i = 1, 2, \ldots, N \qquad \text{Equation (12.36)}$$

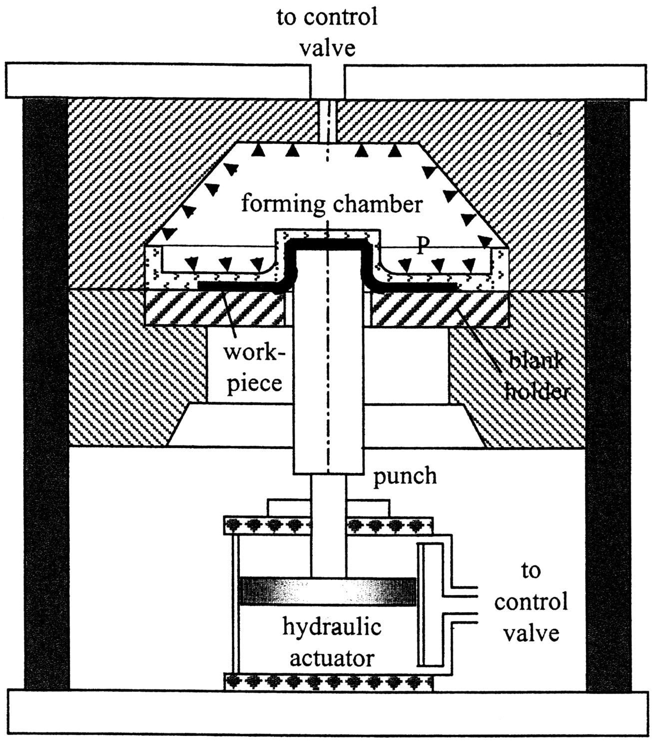


(a) Reactive ion etching process

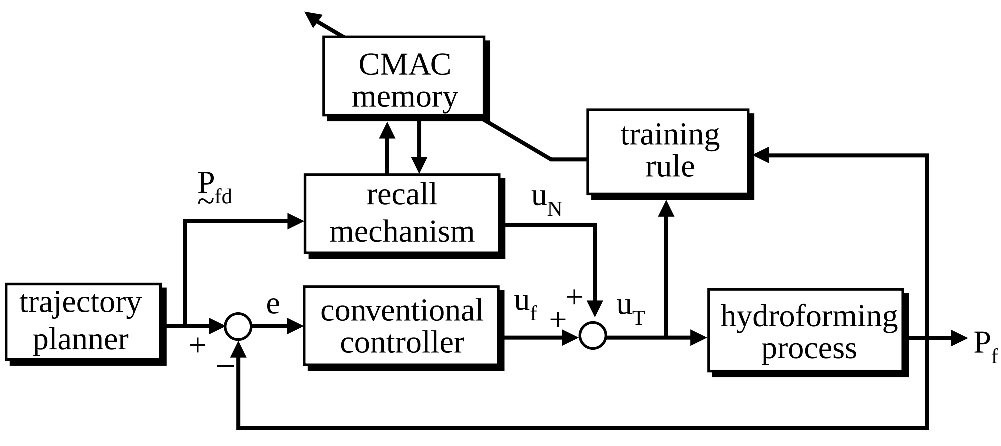


(b) Illustration of simple control scheme

FIGURE 12.15 Reactive ion etching process and the neural controller. (a) Reactive ion etching process. (b) Illustration of a simple control scheme.



(a) A schematic of hydroforming process



(b) The CMAC-based control system

FIGURE 12.16 The hydroforming process and the proposed pressure control system. (a) A schematic of the hydroforming process. (b) The CMAC-based control system.

where i is the memory address. The update rule for w_i is given by

$$\Delta w_i = \eta \cdot \frac{u(t) - g\left(\tilde{p}_f(t)\right)}{\gamma} \qquad \text{Equation (12.37)}$$

where γ is the generalization factor.

Utilizing the hydroforming machine, a series of experiments was carried out for various product shapes. The parameters of the CMAC used here such as learning rate η and generalization factor γ were set within a certain range. The controlled chamber pressure was found to approach the desired curve as the number of trials increased. In the long run, at the tenth trial, almost no difference was observed between the actual and desired chamber pressures. Several products produced by this control method show that control of the product quality in this manner can be used effectively for actual hydroforming process. The neural network application here is effective and highly recommendable, since the process is characterized by a highly nonlinear, uncertain, complex system, which may not achieve desired accuracy by conventional control techniques.

12.7 Conclusions

The current intense competition in markets requires manufacturers to produce high-quality products with lower cost and higher productivity. This stringent situation has created an issue of growing importance in the manufacturing community and evoked a necessity for reliable and robust monitoring and control systems with high performance that do not require any justification. Recently, artificial neural networks have emerged as an intelligent tool to approach this problem and are shown to offer great promise, especially in monitoring and control of manufacturing processes.

In this chapter the characteristics of the manufacturing processes were analyzed. According to this, the monitoring and control problems were identified and the use of artificial neural networks to solve them was justified. Types of sensor signals, network structures, and output variables for monitoring and control were surveyed for application domains covering a variety of processes. It has been pointed out that due to inherent functionalities of neural networks, they provide monitoring systems and network-based control systems with capabilities of handling time-varying parameters and uncertainty, and suppressing process noise and complexity involved with process phenomena better than the conventional techniques. These abilities will take the monitoring and control systems closer to a truly intelligent manufacturing system. However, there are problem domains that still need to further enhance the capabilities of the neural-based systems network. Current limitations or shortcomings of the neural networks are given in the following:

1. In identifying the processes, the networks are utilized like a transfer function which maps the input data into the output variables. The pitfall of this is that this method is just a black box approach, since the networks do not know what is going on with regard to the physical phenomena of the processes.
2. In pattern recognition and clustering, there still remains a limitation to the use of neural networks. Their performance here somewhat depends upon the statistical properties of sampled data. Accuracy clustering data located at the proximity of border lines of each group still need to be improved, since most manufacturing processes require nearly 100% correct judgment.
3. In process control, the networks assume roles in identification and/or generation of control signals based upon sensor data. As in many other problems, robustness and accuracy are the key issues that promote their implementation in real processes.

There are several other shortcomings not listed here, but all of these may be gradually solved when the networks are equipped with functionalities that accommodate the manufacturing-specific physical nature by enhancing the structure, learning algorithm, and convergency properties of the currently used networks. To approach and solve this, one must understand the characteristics of the manufacturing processes first and then design appropriate networks.

Defining Terms

Multisensor fusion: A technique for using a number of sensors to improve sensing accuracy by reducing the uncertainty of each sensor.

Weight pruning: A technique to prune an initially large structured network by weakening or eliminating certain synaptic weights in a selective and orderly fashion.

Hyper-plane: A plane is defined in a fictitious multidimensional space.

Kernel function: A simple function to estimate the probabilistic density.

Euclidean distance: A similarity measure expressed by the root-squared sum of the difference between two vectors.

Nonparametric classifier: Nonparametric methods do not need to use the parameter of a predefined model for classification, compared with parametric classifiers based on a model that can be completely described by the chosen mathematical function using the small and fixed number of parameters.

Intelligent control: A control method that does not rely on mathematical models of processes or plants but uses biologically inspired techniques and procedures.

Inverse model: A mathematical model of processes or plants to be controlled that yields the corresponding control/operating inputs for the given desired outputs.

References

Azouzi, R., and Guillot, M. 1996. Control and optimization of the turning process using a neural network, *Japan/USA Symposium on Flexible Automation ASME,* vol. 2, pp. 1437-1444.

Chen, Y., Li, X., and Orady, E. 1996. Integrated diagnosis using information-gain-weighted radial basis function network, *Computers and Ind. Engineering,* vol. 30, no. 2, pp. 243-255.

Chryssolouris, G., and Guillot, M. 1990. A comparison of statistical and AI approaches to the selection of process parameters in intelligent machining, *ASME Trans. J. Engineering for Industry,* vol. 112, pp. 122-131.

Grabec, I., and Kuljanic, E. 1994. Characterization of manufacturing processes based upon acoustic emission analysis by neural networks, *Annals of the CIRP,* vol. 43, pp. 77-80.

Khanchustambhan, R.G., and Zhang, G.M. 1992. A neural network approach to on-line monitoring of a turning process, *IEEE International Joint Conference on Neural Networks,* vol. 2, pp. 889-894.

Kim, J.H., and Cho, H.S. 1995. Neural network-based inspection of solder joints using a circular illumination, *Image and Vision Computing,* vol. 13, no. 6, pp. 479-490.

Ko, K.W., Cho, H.S., Kim, J.H., and Kong, W.I. 1998. A bead shape classification method using neural network in high frequency electric resistance weld, *Proc. of World Automation Congress.*

Javed, M.A., and Sanders, S.A.C. 1991. Neural networks based learning and adaptive control for manufacturing systems, *IEEE/RSJ International Workshop on Intelligent Robots and Systems,* pp. 242-246.

Lim, T.G., and Cho, H.S. 1993, A study on the estimation and control of weld pool sizes in GMA welding processes using multilayer perceptrons, Ph.D. thesis, *Korea Institute of Science and Technology.*

Lippmann, R.P. 1989. Pattern classification using neural networks, *IEEE Communication Magazine,* November, pp. 47-64.

Okafor, C., and Adetona, O. 1995. Predicting quality characteristic of end-milled parts based on multisensor integration using neural networks: individual effects of learning parameters and rules, *Journal of Intelligent Manufacturing*, vol. 6, pp. 389-400.

Park H.J., and Cho, H.S. 1990. A CMAC-based learning controller for pressure tracking control of hydroforming processes, *ASME Winter Annual Meeting*, Dollars, Texas.

Quero, J.M, Millan, R.L., and Franquelo, L.G. 1994. Neural network approach to weld quality monitoring, *International Conference on Industrial Electronics, Control and Instruments*, vol. 2, pp. 1287-1291.

Sokolowski, A., and Kosmol, J. 1996. Intelligent monitoring system designer, *Japan/USA Symposium on Flexible Automation*, vol. 2, pp. 1461-1468.

Stokes, D., and May, G. 1997. Real-time control of reactive ion etching using neural networks, *Proc. American Control Conference*, pp. 1575-1578.

Woo, H.G., and Cho, H.S. 1988. Estimation of hardening layer sizes in laser surface hardening processes with variations of coating thickness, *Surface and Coatings Technology*, vol. 102, pp. 205-217.

Zurada, J.M. 1992, *Introduction to Artificial Neural Systems*, West Publishing, St. Paul, MN.