Article ID: 1000 - 8152(2001)03 - 0401 - 08

Neural Fuzzy Logic Control for Gas Tungsten Arc Welding *

LI Di
(Mechatronic Engineering Department, South China University of Technology · Guangzhou, 510640, P.R.China)
T. Srikanthan
(School of Applied Science, Nanyang Technological University, Singapore)
Inoue Katsunori
(Joining and Welding Research Institute, Osaka University, Japan)

Abstract: A novel technique, that combines the FLC and neural network (NN) techniques, to control the gas tungsten arc welding (GTAW) process is presented. This technique overcomes the limitations such as the dependency on the experts for fuzzy rule generation, the fuzzy set that is non-adaptive, etc. The adaptation of membership function as well as the self-organizing of fuzzy rule are realized by the self-learning and competitiveness of the NN. This approach facilitates a mechanism for an automatic determination of the fuzzy rule and in-process adaptation of membership function for an advanced welding process control. This is because a fixed membership function cannot guarantee the required system performance, as the arc-welding process is a highly time-variable system. Taking GTAW process welds bead width that regulates the system as the controlled plant, the proposed algorithm has been verified to be highly effective for an arc-welding process. Computer simulations confirm that the characteristics of the system have improved notably when compared with a number of currently available methods.

Key words: fuzzy logic; neural network; membership function; gas tungsten arc welding
Document code: A

弧焊过程神经网络模糊控制

李　迪
(华南理工大学机电系·广州,510604)
T. Srikanthan　　井上胜境
(南洋理工大学应用科学学院·新加坡)　(大阪大学接合科学研究所·日本)

摘要: 提出一种将FLC与神经网络技术相结合的方法对钨极氩弧焊(GTAW)过程进行控制.它克服了模糊规则产生对专家的依赖及模糊集非自适应性的问题.隶属函数的自适应及模糊规则的自组织通过神经网络的自学习和竞争获得.该方法实现了弧焊过程中模糊规则的自动确定和隶属度函数的在线调节.以GTAW过程焊缝几何参数调节为对象,验证了算法的有效性.计算机仿真表明,采用该方法的系统性能有较大的提高.

关键词: 模糊逻辑; 神经网络; 隶属函数; 钨极保护气体氩弧焊

1 Introduction

Fuzzy Logic Control (FLC) is a knowledge-based control strategy that has proven its potential in industrial control applications in recent years. It can be used when a sufficiently accurate, yet not unreasonably complex model of the physical system to be controlled is unavailable or when a precise measure of performance is either not meaningful or practical. Fuzzy logic is much closer in spirit to human thinking and natural language than the traditional logic systems. The control design problem makes use of empirically acquired knowledge of the process operation instead of analytic framework. The core of the FLC is its linguistic or rule based form of knowledge expression.

FLC is especially suited for the ill defined and uncertain systems where conventional mathematical tools (e.g. differential equations) based on modeling and control fail. Arc welding is one such process that involves heat

* Foundation item: supported by National Natural Science Foundation of China (5005007, 59785004) and Guangdong Provincial Natural Science Foundations (940504, 990550) and SRF for ROCS, SEM (9927).
Received date: 2000 - 03 - 03; Revised date: 2000 - 12 - 26.

and metal transfer, phase transfomations and many unknown disturbances and thus difficult to model by conventional mathematical framework based approach. The fuzzy logic control is a well-qualified candidate for such a process. Langari et al[1] and Satoshi Yamane et al[2] have shown encouraging results with the control of weld pool by means of FLC. They used fuzzy IF-THEN rules to model the qualitative aspects of expert's knowledge and reasoning process of the controlled arc welding process. However, there are still some aspects of this research that need to be addressed. In fuzzy logic control strategy the membership functions for the fuzzy sets are determined by the expert prior to on-line control process begins and they normally remain unchanged during the control process. This may bring unsatisfactory results when the controlled plant is time-variable or disturbance exists in the process. In addition, the determination of fuzzy rule and the membership function is heavily dependent on the experts' judgment. In this approach, satisfactory results may not be possible if the level of experience is not adequate. Thus, an on-line automatic tuning of the membership function according to the varying plant characteristics and pre-defined performance requirements is necessary for the effective control of the welding process. It would also be beneficial to develop a common method for transforming human knowledge or experience into the rule base of FLC for this application. This requires that there must be a provision to detect the changes in the membership functions and to automatically determine the rule-base of the FLC using the data obtained directly from the arc welding system.

Among machine-intelligent adaptive control systems, neural network (NN) control is also a viable alternative to FLC. Neural networks can be trained to mimic biological neural systems in performing functions such as learning and pattern recognition. It has been successfully applied to a range of process controls including arc welding process control[3,4]. Although, neural network can automatically learn from the samples of data, it lacks the explanatory ability. While FLC can perform approximate reasoning, it is usually not self-adaptive. The desire for a learning ability in FLC encourages one to incorporate the learning ability of NN into FLC. This has prompted many researchers to search for ways to combine the two techniques[5-7].

In light of the above, the objective of this work is to embed NN into FLC to realize a fuzzy rule generation mechanism that is both self-organizing and self-adaptive. This should then provide for the fine-tuning of membership function for arc welding process control.

2 Neural network based fuzzy logic controller

2.1 Fuzzy logic control in arc welding process

Figure 1 shows the basic structure of fuzzy logic control of an arc welding system. In a control based application, the error (E) and change of error (CE) are chosen as the inputs to fuzzy logic controller, while the change of control input to the arc welding process is selected as the output of the FLC. In a system where weld pool is to be controlled, E may be defined as the error between the desired weld feature such as width and actual width of the weld pool. The control input to the process may be the welding current, travel speed etc[4]. This enables the formulation of simple linguistic rules based on observation or simple study about the process. In this work, the fuzzy controller with min as the AND operator, max as the OR operator and center of area defuzzification is used. A bell shape function is used as the membership function with the following form:

$$f(x) = e^{-(x-m)^2/\sigma^2}, \tag{1}$$

where m and σ are the center and the width of the membership function.

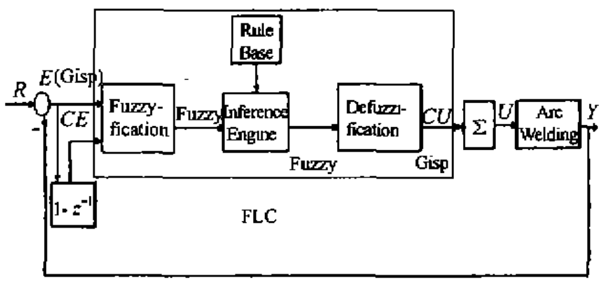

Fig. 1 Basic structure of fuzzy logic control are welding system

2.2 Structure of the neural network based on fuzzy logic controller

By incorporating NN into fuzzy logic controller, the fine tuning of membership function and automatic fuzzy logic rule generation can be realized. The feed forward network as illustrated in Fig. 2 is incorporated to realize the above fuzzification, inference engine and defuzzification function.

Fig. 2 also shows the structure of the FLC realized by a feed forward neural network that has a total of 5 layers with each layer performing different functions of the FLC. Node types are shown as squares and circles to distinguish them. The square-node means that the parameters of the node need to be regulated, while the circle-node means that the parameters of the node remain fixed during the learning process of the network. The arrows between two layers of nodes indicate the direction of the signal flow.

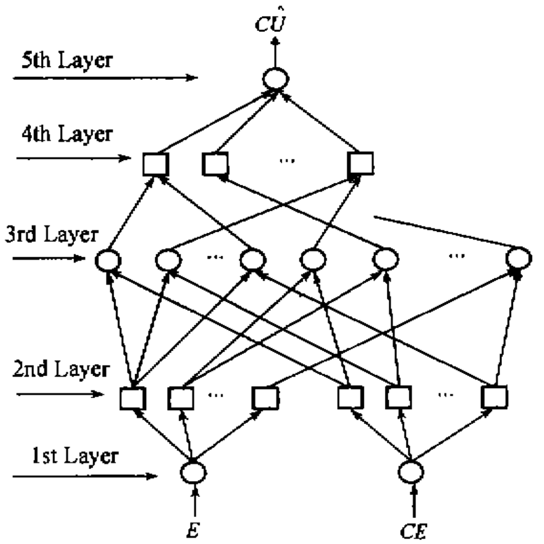

Fig. 2 Structure on NN based fuzzy logic controller

The structure of the FLC shown in Fig. 2 resembles a feed-forward neural network. The three components of FLC, namely: fuzzification, fuzzy inference and defuzzification, have been incorporated into this neural network. The different layers of this NN perform different functions of FLC. This neural network like FLC makes the adaptation of the membership function as well as the dynamic generation of self-organizing fuzzy logic rule possible. The brief description of the function of each layer is given below. For detail explanation please refer to [8].

Function of 1st layer:

Function of 1 st layer is to transfer the inputs to the 2nd layer.

$$u_i^{(1)} = \begin{cases} E, & (i = 1), \\ CE, & (i = 2), \end{cases} \tag{2}$$

$$o_i^{(1)} = a_i^{(1)} = u_i^{(1)},\ i = 1,2.$$

Function of 2nd layer:

The 2nd layer performs the fuzzification task. For the ith node in 2nd layer, the input $n_i^{(2)}$, activation $a_i^{(2)}$ and output $O_i^{(2)}$ are defined as follow:

$$\begin{cases} n_i^{(2)} = -\dfrac{(u_{ji}^{(2)} - m_{ji})^2}{\sigma_{ji}^2}, \\ o_i^{(2)} = a_i^{(2)} = e^{n_i^{(2)}}. \end{cases} \tag{3}$$

Here, m_{ji} are σ_{ji} the center and width of the ith membership function for jth input ($j = 1,2$) in the first layer.

Function of 3rd layer:

The 3rd layer performs fuzzy AND operation. For the ith node, the input $n_i^{(3)}$, activation $a_i^{(3)}$ and output $o_i^{(3)}$ are as follows:

$$\begin{cases} n_i^{(3)} = \min(u_{j_1 i}^{(3)}, u_{j_2 i}^{(3)}), \\ o_i^{(3)} = a_i^{(3)} = n_i^{(3)}, \end{cases} \tag{4}$$

where j_1 and j_2 are the indexes for nodes in the 2nd layer. These nodes respond to one of the membership functions of input 1 (E) and input 2 (CE), from which the ith node in 3rd layer receives the signal.

Function of 4th layer:

The 4th layer performs fuzzy OR operation:

$$\begin{cases} n_i^{(4)} = \max(u_{j_1 i}^{(4)}, u_{j_2 i}^{(4)}, \cdots, u_{j_p i}^{(4)}), \\ o_i^{(4)} = a_i^{(4)} = n_i^{(4)}, \end{cases} \tag{5}$$

where $j_1, j_2, \cdots, j_p$ are the indexes of the nodes in the 3rd layer from where the ith node in the 4th layer receives the signal.

Function of the 5th layer:

The 5th layer performs defuzzification by the center of area method. Thus we have:

$$\begin{cases} n_1^{(5)} = \dfrac{\sum_{i=1}^{q} m_i \sigma_i u_{i1}^{(5)}}{\sum_{i=1}^{q} \sigma_i u_{i1}^{(5)}}, \\ o_1^{(5)} = a_1^{(5)} = n_1^{(5)}, \end{cases} \tag{6}$$

where m_i, σ_i are the center and width of the ith membership function of the output (CU).

The learning process of the network, which uses the back-propagation algorithm, refines the parameters in the square nodes.

2.3 Automatic determination of the linkage between the nodes of 3rd and 4th layer

The purpose of the automatic determination of the linkage between the 3rd and 4th layer is to dynamically generate the 'THEN' part of the rule for observed sample data for a pair of input and output variable terms. The steps required for this task are specified below.

Ⅰ) Using the c-means clustering method and the samples obtained, generate the center value of m_{ij} of each cluster for each input/output variable. The cluster number of a variable, which equals to the term number of that variable, can then be chosen according to the complexity of the controlled process.

Ⅱ) The width of each membership function σ_{ij} can be determined according to the pre-defined overlap parameter.

Ⅲ) Construct a network in which the nodes in 3rd layer fully connect with the nodes in the 4th layer at start. This network is similar to the network in Fig.2, except that the CU (sample data, which is not the same as network output in Fig.2) is fed into the network according to the direction of flow. For each pair of sample data, the inputs (E and CE) are propagated to the 2nd and 3rd layer according to the direction of flow. Similarly, the output (CU) is then propagated to the 4th layer in order to generate the output of layer 4 with the following equation:

$$\delta_i^{(4)} = e^{-\frac{(CU_k - m_i)^2}{\sigma_i^2}}, \quad k = 1,2,\cdots,M, \tag{7}$$

where M is the number of the sample data.

Ⅳ) With the output in layers 3 and 4, perform the competitive learning in these 2 layers. Here, it is assumed that the output of the 4th layer shows the degree of win of the output data in layer 3 that corresponds to a pair of sample data. The weights for each link from ith node in 3rd layer to the jth node in 4th layer, w_{ij} can be obtained by the following competitive learning:

$$\dot{w}_{ij} = -\delta_j^{(4)}(w_{ij} - o_i^{(3)}). \tag{8}$$

For the whole set of sample data, perform Steps 1 ~ 4 to get the value of w_{ij}.

Ⅴ) For the ith node in 3rd layer, find the maximum weight:

$$w_{iJ} = \max(w_{i1}, w_{i2}, \cdots, w_{ip}), \tag{9}$$

where p is the number of the 4th nodes and J is the index of the 4th node with the maximum weight.

Ⅵ) The ith node in 3rd layer has a link only with the node in the 4th layer that have the maximum weight of the ith node. In this way, the linkage can be determined for every node in the 3rd layer.

2.4 Training algorithm for the neural network based fuzzy logic controller

The aim here is to give the sample data

$$(E_1, CE_1, CU_1), (E_2, CE_2, CU_2), \cdots, (E_M, CE_M, CU_M), \tag{10}$$

and the link between the 3rd and 4th layers so that a fine tuning of the parameters of square nodes could be made to minimize the following error.

$$TE = \sum_{l=1}^{M} TE_l, \tag{11}$$

where

$$TE_l = \frac{1}{2}(CU_l - \hat{CU}_l)^2. \tag{12}$$

$CU, \hat{CU}$ are the sample data and output of the neural network, respectively. The back-propagation algorithm is used to regulate the parameter. The error is back propagated from the output layer to the hidden layers in order to refine the parameters of square nodes. The training algorithm shown below starts from the output layer (5th layer) to the 2nd layer.

5th layer:

In this layer, the center and width of the membership

$$m_i = m_i + \eta(CU_l - \hat{CU}_l) \cdot \frac{\sigma_i u_{i1}^{(5)}}{\sum \sigma_i u_{i1}^{(5)}},$$

$$\sigma_i = \sigma_i + \eta(CU_l - \hat{CU}_l) \cdot \frac{m_i u_{i1}^{(5)}(\sum \sigma_i u_{i1}^{(5)}) - (\sum m_i \sigma_i u_{i1}^{(5)}) u_{i1}^{(5)}}{(\sum \sigma_i u_{i1}^{(5)})^2}, \tag{13}$$

$$\delta_1^{(5)} = -\frac{\partial TE_1}{\partial n_1^{(5)}} = \frac{\partial TE_l}{\partial a_1^{(5)}} \cdot \frac{\partial a_1^{(5)}}{\partial n_1^{(5)}} = CU_l - \hat{CU}_l. \tag{14}$$

4th layer:

In this layer, no parameter needs to be regulated. Only the error needs to be computed.

$$\delta_i^{(4)} = \delta_1^{(5)} \frac{m_i \sigma_i(\sum \sigma_i u_{i1}^{(5)}) - (\sum m_i \sigma_i u_{i1}^{(5)}) \sigma_i}{(\sum \sigma_i u_{i1}^{(5)})^2}. \tag{15}$$

3rd layer:

Only error needs to be propagated in this layer.

$$\delta_i^{(3)} = \delta_j^{(4)}. \tag{16}$$

Here j is the index of the nodes in 4th layer which connect with the ith node in 3rd layer.

2nd layer:

In this layer, the parameters for the membership function of the controller input need to be regulated.

According to Eq. (10)

$$m_{ij} = m_{ij} - \eta \frac{\partial TE_l}{\partial a_j^{(2)}} e^{n_j^{(2)}} \frac{2(u_{ij}^{(2)} - m_{ij})}{\sigma_{ij}^2}, \quad (17)$$

$$\sigma_{ij} = \sigma_{ij} - \eta \frac{\partial TE_1}{\partial a_j^{(2)}} e^{n_j^{(2)}} \frac{2(u_{ij}^{(2)} - m_{ij})^2}{\sigma_i^3}, \quad (18)$$

where

$$\frac{\partial TE_l}{\partial a_j^{(2)}} = \sum_k q_k,$$

$$q_k = \begin{cases} -\delta_k^{(3)}, & \text{if } a_j^{(2)} \text{ is the minimun of all} \\ & \text{input to the } k\text{th node in layer 3,} \\ 0, & \text{otherwise.} \end{cases} \quad (19)$$

3 Simulation results

Consider the gas tungsten arc welding (GTAW) as an example of a process to be controlled, where the direct welding parameter (DWP) is the welding pool width and the indirect welding parameter (IWP) (also, control variable) is the welding current. The main aim of this control problem is:

Given the sample data of GTAW process, design the neural network fuzzy controller which provides the $U(k)$ so as to obtain the output $Y(k+1)$, which approaches desired output.

3.1 Obtaining of the sample data

The aim of this step is to obtain the sample data to be used for training the fuzzy neural network (FNN). In this simulation the sample data is $\{(I_k, W_k), k = 1,2,\cdots,N\}$, where N is the number of the sample data and I_k, W_k is the welding current and width of welding pool at time steps of k, respectively. To obtain the sample data, the welding current at step k $I_k \in [I_{\min}, I_{\max}]$ is given, where $I_{\min}$ is the minimum required welding current to form the welding pool, while $I_{\max}$ is the maximum welding current to ensure that a burn through does not take place. At each step k, the welding width W_k corresponding to welding current I_k is calculated.

Assuming that the heat intensity q_k at time step k to the welding work piece is

$$q_k = \frac{3\eta I_k U}{h\pi r_a^2} \exp\left(-\frac{3r^2}{r_a^2}\right), \quad (20)$$

where η: efficiency of arc, I_k: welding current at time step k, U: arc voltage, h: thickness of work piece, r_a: efficient heat input radius, r: distance from the point in the work piece to the center of heat source.

The heat source used in these simulations follows Gaussian distribution and the weld pool width is the maximum width of melting area. By calculating the temperature distribution of step k according to the melting point of the work piece material, the welding pool width can be obtained.

The temperature distribution has been calculated with the help of ABAQUS software package[9]. The conditions for the calculation are shown in Table 1.

Table 1 Calculation conditions

material	1Cr18Ni9Ti
size of work-piece	250mm × 100mm × 1mm
η	0.65
U	10.0V
r_a	3.0mm
v (torch traveling speed)	1.5mm/s
I_k	20.0 ~ 60.0A
liquid conductivity	22.0W/(m·K)
solid conductivity	20.0W/(m·K)
specific heat(liquid and static)	735.0J/(kg·K)
liquidus temperature	1723K
solidus temperature	1523K
specific weight	7200kg/mm³
latent heat	2.47×10^5 J/kg

Fig.3 shows the sample data at every step k ($k = 1, 2, \cdots, 148$).

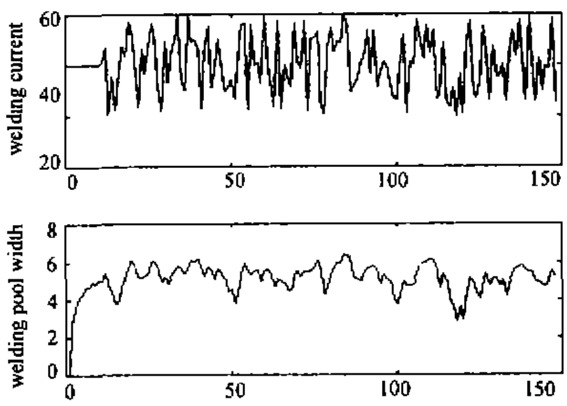

Fig. 3 Sample data for welding current and pool width

3.2 Approximation of process model

This step aims to configure the model between the welding current and pool width for simulation purposes. Although the ABAQUS s/w could have been used for this propose, it was decided to realize this calculation by other method mainly due to the fact that the calculation speed of ABAQUS is very low for our simulation purposes. This relationship between the welding current and pool width was set up using BP network based on the fact that BP network can approximate any function with any degree of similarity. This step is not required for the on-line control of a real process as the output of the process corresponding to input can be obtained directly from the plant.

The relationship between welding pool width and welding current can be expressed by the following equation:

$$W(k+1) = f(W(k), W(k-1), \cdots, W(k-n), I(k), I(k-1), \cdots, I(k-m)), \tag{21}$$

where the function f represents a memory-less nonlinear function.

Of the sample data used in Fig.3, data numbered 11 ~ 148 were chosen as the training sample data. The established model is used to replace the practical GTAW process during simulation.

3.3 Simulation results for FNN control system

The steps taken during the simulation are as follows:

Step A The sample data for training FLC were generated from Fig.3 using the following expression:

$$\{(E_k, CE_k), CU_k, k = 11, 12, \cdots, 148\},$$
$$\begin{cases} E_k = W(k+1) - W(k), \\ CE_k = E_k - E_{k-1}, \\ CU_k = I_k - I_{k-1}. \end{cases} \tag{22}$$

Step B Perform the clustering of E_k, CE_k, and CU_k in order to assign the center of each cluster as the center for each membership function. Choose the cluster number as 3, which corresponds to Negative, Zero and Positive respectively. Obtain the width corresponding to each membership function by choosing the overlap parameter as 1.5.

Step C Competitive learning is used to obtain the link between 3 and 4 and 'THEN' part of rule can be obtained. Table 2 is the rule base obtained by competitive learning using these sample data sets. From this, we can see that the link between 3rd and 4th layer has a strong explanatory ability. The above fuzzy rules are corresponding to those in [1].

Table 2 The rule base obtained by competitive learning with 138 sample data set shown in Fig.5 numbered 11 ~ 148th

CE	E: Negative	E: Zero	E: Positive
Negative	N	N	N
Zero	N	Z	Z
Positive	P	P	P

Step D Using the obtained structure of FNN and sample data sets above, perform an off-line tuning of the membership function by BP algorithm. The membership function after the training of BP network is shown in Fig.4.

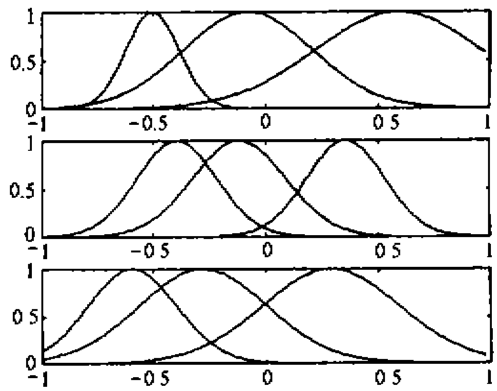

Fig. 4 Membership function of CU (upper), E (middle) and CE (lower) after the off-line fine tuning by BP algorithm

Step E Construct the close loop with the form of Fig.1 to realize the close loop control of GTAW process. In every control cycle, tune the membership function according to the real input and output of the GTAW process.

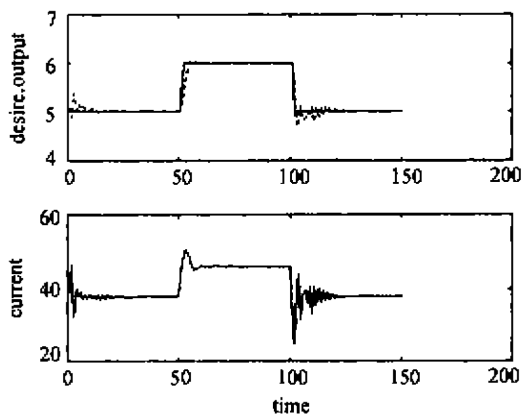

Fig. 5 Control results with the fine tuning of membership function

Fig.5 shows the control results and welding current at every control cycle that were obtained using the proposed neural based fuzzy logic controller. In this figure, the solid line represents the desired output while the dashed line represents the output of the control system. This method of interpretation applies to the remaining figures.

4 Comparison with other methods

4.1 FLC without on-line fine tuning

Fig.6 shows the result without the on-line fine-tuning of membership function. The membership function is obtained directly from the off-line learning by using the sample data shown in Fig.3. It can be seen that there exists certain degree of oscillation and steady state error.

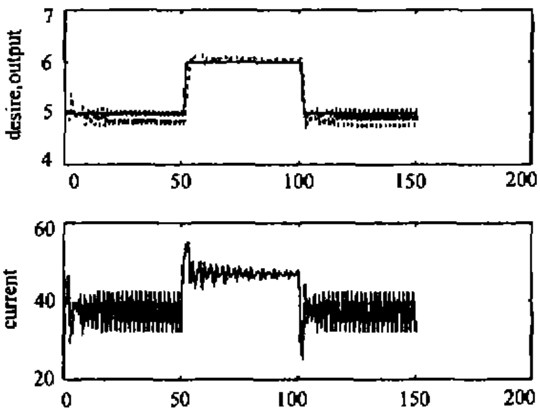

Fig. 6 Control results without the on-line tuning of nembership function

4.2 PI controller

According to Suzuki and Hardt et al[10], the dynamics for GTAW can be expressed as a first order transfer function between the welding pool width and welding current. This is expressed as follows:

$$G(s) = \frac{W(s)}{I(s)} = \frac{K_p}{\tau_p s + 1}. \tag{23}$$

It's ZOH equivalence form is

$$G(z) = \frac{b_p}{z + a_p}, \tag{24}$$

where

$$a_p = -e^{-\frac{T}{\tau_p}},\ b_p = K_p(1 + a_p)$$

and T = sample time.

By using the dynamic model realized by the BP algorithm described in 3.2, we can generate a sequence that can be used for identification of parameters a_p and b_p.

Parameters can be obtained by the least-squares method. The following standard structure of PI controller is used[11].

$$\frac{u(z)}{e(z)} = K_D + \frac{K_D T}{T_{ID}(z-1)}. \tag{25}$$

By using the transient-response method[11], the parameters of PI controller $K_D = 3.6873$ and $T_{ID} = 1.0642$ can be determined. The control result obtained using the PI controller is shown in Fig.7.

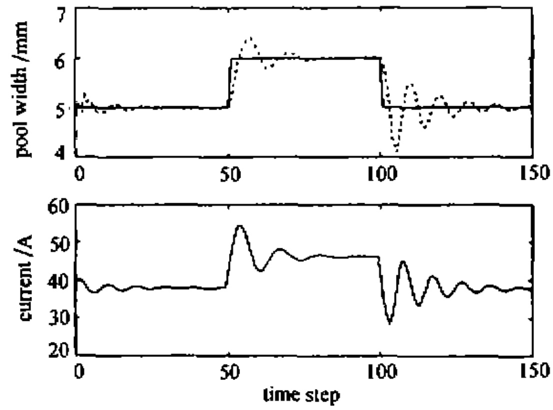

Fig. 7 Output results of PI control system

4.3 STC/PP method

The STP/PP method used here is the same as that used by Suzuki[11], the details of which are not detailed here. The closed loop model is the same as that given in [11], which is of the following form:

$$G_m(s) = \frac{1}{s+1}. \tag{26}$$

The parameters of GTA process a_p and b_p are updated in every control cycle with the following recursive least squares method.

$$\begin{cases} \theta_k = \theta_{k-1} + P_k \phi_k (y_k - \theta_{k-1}^{\mathrm{T}} \phi_k), \\ \theta^{\mathrm{T}} = [a_p \quad b_p], \\ \phi_k = [-y_{k-1} \quad u_{k-1}]^{\mathrm{T}}, \\ P_k = \frac{1}{\lambda} P_{k-1} - \frac{P_{k-1}\phi_k\phi_k^{\mathrm{T}}P_{k-1}}{\lambda + \phi_k^{\mathrm{T}}P_{k-1}\phi_k}. \end{cases} \tag{27}$$

The effect of control of the STC/PP method is shown in Fig.8.

From the above comparisons, it can be clearly seen that neural network based FLC has the better properties when compared with the FLC without on-line tuning, PI control method and STP/PP method for the control of the GTAW process for the case of welding current and welding pool width as the input and output respectively. It is also evident that the fine tuning of membership function can improve the characteristics of the system. The PI and STC/PP methods are based on the assumption that the GTAW is a linear system. This assumption

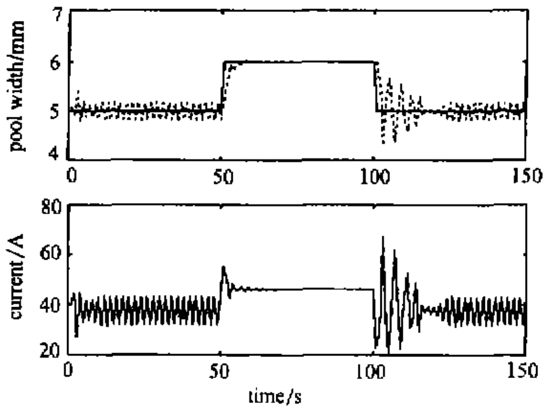

Fig. 8 Control result of STC/PP control system

leads to the neglecting of nonlinear properties of GTAW process, which results in a clear diversion between the established model and the real process. As the result of this, the control result is not as good. In STC/PP method the process model is updated according to the real input and output of GTAW process. This somewhat compensates for the non – linearity of the process by linearizing the model in a small space. This leads to better dynamic properties than those realized by the PI control system.

5 Conclusions and discussions

This work confirms that a neural network based fuzzy logic controller can be used effectively in GTAW process control without deriving a mathematical model of the process which is obtained with the help of computer simulation results. Unlike the traditional FLC, neural network based FLC incorporates a learning ability which can be used in the fine tuning of membership function to minimize the output error of the control system. This allows for the properties of the control system to be improved. It has been shown that the rule base can be generated automatically by the proposed method. This is very useful as the proposed technique does not rely heavily on the inputs from an expert. It has also been shown that the proposed neural network based FLC with on-line fine tuning has better characteristics than those obtained using traditional FLC without on-line fine-tuning, PI control and STC/PP based approaches. However, when the input of the system is subjected to an abrupt change (e.g. welding pool width from 6 ~ 5 mm in time step 100 in Fig. 6), the control variable near this time step consists of some oscillations. These oscillations can be reduced by incorporating the change in input variable in the performance index (equation (17)).

References

[1] Langari G and Tomizuka M. Self organizing fuzzy linguistic control with application to arc welding [A]. IEEE Int. Workshop on Intelligent Robots and Systems (IROS'90) [C], USA, 1990, 1007 – 1014

[2] Satoshi Yamane, Yasuyoshi Kaneko, Noriguki kitahara. Neural network and fuzzy control of weld pool with welding robot [A]. Manufacturing Systems Development and Applications Department Conference Records-IAS Annual Meeting [C], NJ, USA: IEEE, 1993, 2175 – 2180

[3] George E Cook, Robert Joel Barnett, Kristinn Andersen, et al. Weld modeling and control using artificial neural network [J]. IEEE Transactions on Industry Applications, 1995, 31(6): 1484 – 1491

[4] Huang Shisheng, Li Di and Song Yonglun. Welding quality control by neural networks [J]. Welding in The World, 1994, 34, 359 – 363

[5] Jyh-Shing Roger Jang. ANFIS: Adaptive-network-based fuzzy inference system [J]. IEEE Transactions on System, Man, and Cybernetics, 1993, 23(3): 665 – 685

[6] Lin C T. Neural Fuzzy Control System with structure and Parameter Learning [s]. Singapore: World Scientific, 1994

[7] Atsushi Aoyama, Francis J, Doyle III, et al. A fuzzy neural-network approach for nonlinear process control [J]. Engng. Applic. Artif. Intell., 1995, 8(5): 483 – 498

[8] Wang Lixin. Adaptive Fuzzy Systems and Control: Design and Stability Analysis [M]. N J: Prentice-Hall, 1994

[9] ABAQUS/Standard User's Manual [S]. Version5.5, logo: Hibbitt, Karlsson & Sorensen, 1995

[10] Suzuki A, Hardt D E and Valavenu L. Application of adaptive control theory to on-line GTA weld geometry regulation [J]. J. Dynamic Systems, Measurement, and Control, 1991, 113(1): 93 – 103

[11] Aström K J and Wittenmark B. Computer Controlled Systems [M]. N J: Prentice-Hall, 1984

本文作者简介

李 迪 1965年生.1993年华南理工大学自动化系毕业,获工学博士学位,现为机电系副教授.从事制造过程智能化监测及控制方面的研究工作,承担并完成省级及国家级项目多项并获国家教委科技进步二等奖一项.在国内外发表论文20余篇.

T. Srikathan 新加坡南洋理工大学副教授,高性能嵌入式系统中心主任.

井上胜填 日本大阪大学焊接科学研究所教授,所长.