Self-Learning Fuzzy Neural Networks and Computer Vision for Control of Pulsed GTAW

Neural network modeling and computer vision techniques are used to control the dynamics of the pulsed gas tungsten arc welding process

BY S. B. CHEN, L. WU, Q. L. WANG AND Y. C. LIU

ABSTRACT. The objective of this research is to apply intelligent control methodology to improve weld quality. Based on fuzzy logic and artificial neural network theory, a self-learning fuzzy and neural network control scheme has been developed for real-time control of pulsed gas tungsten arc welding (GTAW). Using an industrial TV camera as the sensor, the weld face width of the weld pool, i.e., the feedback signal in the closed loop system, is obtained by computer image processing techniques. The computer vision providing process status information in real-time is an integral part of a self-learning fuzzy neural control system. Such a system enables adaptive altering of welding parameters to compensate for changing environments. The experiments on the control of the pulsed GTAW process show that the scheme presented in this paper can be used to control complicated variables such as encountered in welding processes.

Introduction

Owing to the uncertainties of phenomena such as metallurgy, heat transfer, chemical reaction, arc physics and magnetization, the arc welding process is inherently variable, nonlinear, time-varying and strong coupling in its input/output relationships. As a result, it is very difficult to obtain a practical and controllable model of the arc welding process by classical modeling approaches. Until now, control of weld quality, such as weld joint penetration, weld bead formation, and control of the size and changes in the weld pool, is still a perplexing problem, whether for the control engineer or the welding technologist (Refs. 1-7). In recent years, fuzzy logic controllers, which do not require analytical or accurate models of the controlled process, have demonstrated a number of successful applications. Perhaps the most important application field where fuzzy logic plays a significant role is the control of complex industrial processes. In general, these processes may be conventionally adjusted by a human operator due to their complexity. Several industrial applications of fuzzy logic control have been reported (Refs. 8-13). These applications have mainly concentrated on simulating the performance of a skilled human operator in terms of linguistic rules. However, the process of learning and tuning linguistic rules to achieve the desired performance remains a difficult task. Starting with the self-organizing control (SOC) techniques of Mamdani, et al. (Refs. 14-17), the need for research to develop a fuzzy logic controller that can learn from experiences has been realized. The learning task may include the identification of the main control parameters or the development and tuning of the fuzzy memberships used in the control rules.

On the other hand, there has been an increasing interest in studying artificial neural networks (ANN). The ANN is a representation that attempts to mimic the functionality of the brain. The present research on ANN shows that the methodology has been used on a more modest scale to develop nonlinear models and controls. For example, recent studies (Refs. 18-20) demonstrated the utility and flexibility of the concept within the domain of process engineering. It has been proven that any continuous function can be approximated arbitrarily well on a compact set by a feedforward artificial neural network (Ref. 20). Generally, the ANN model and control system have the following characteristics: a strong robustness, fault tolerance, universality, parallel distributed processing, and learning and adaptive abilities. Moreover, it is noted that Ref. 21 presented neural networks and fuzzy theory from a unified engineering perspective. The combination of fuzzy logic and artificial neural networks obviously has great potential for various fields in industrial engineering.

The experiences of a skilled welder could be summarized as a set of fuzzy logic control rules that are described in terms of IF (caused or conditions) THEN (actions) rules. So we can develop a fuzzy logic controller to imitate a skilled welder's operation of the welding
Fig. 1 — Principle diagram for the self-learning fuzzy neural control of the arc welding process.

Fig. 2 — Fuzzy controller architecture.

Fig. 3 — Simulation of three-term and two-term controllers.

Fig. 4 — Simulation of the output of the fuzzy controller for a step change in input.
where \( \langle x \rangle \) means taking the minimum positive integer or maximum negative integer for \( x \). If we use the general equation

\[
U = \langle \alpha E + (1-\alpha) EC \rangle, \quad \alpha \in [0,1]
\]  

(2.2)

to replace a fuzzy rule table, then we have a fuzzy controller that can be regulated by tuning parameter \( \alpha \). Corresponding to \( \alpha = 0.2 \) and \( \alpha = 0.7 \), the fuzzy controllers are shown as Table 2 and Table 3, respectively. Obviously, Table 1 corresponds to tuning parameter \( \alpha = 0.5 \) in Equation 2.2. Noting the rotation of the zero-zone axis in Tables 1–3, one can modify fuzzy rules by changing \( \alpha \) easily. So it can be seen that tuning parameter \( \alpha \) means changing weights to error \( E \) and error change \( EC \), which is just an imitation of human thinking in regulating actions. For example, the weight to \( EC \) should be larger than that to \( E \) for a high-order system, and vice versa. Producing fuzzy rules or fuzzy controllers by this way, one can overcome difficulties of choosing fuzzy rules by human experiences or trial and error, and also avoid the jump or empty phenomenon in the definition of common fuzzy rules. Similarly, this paper develops the three-term analytical fuzzy controller as in Equation 2.1. Using the controller 2.1, simulation of the controlled object was investigated.

\[
G(s) = \frac{20 \exp(-0.5s)}{(2s + 1)(4s + 1)(2.2s + 1)}
\]

(4.5)

The simulation results are shown in Fig. 2 where the symbol • indicates the unit-step response of the controlled object \( G(s) \) regulated by three-term fuzzy controller (Equation 2.1) with the tuning factors \( a = 0.625 \) and \( b = 0.5625 \); and the symbol * indicates a two-term controller (Equation 2.2) with the tuning factor \( \alpha = 0.625 \). The other parameters in both controllers were identical, that is \( K_e = 30, K_c = 10, K_r = 10 \) and \( K_u = 50 \). Obviously, the simulation results show that the three-term fuzzy controller is more advanced for high-order systems than the common two-term one. More details were illustrated in Ref. 23.

In Fig. 1, the fuzzy controller (Equation 2.1) is adaptively regulated by modifying the tuning parameters \( a(t) \) and \( b(t) \) according to the neural network model of an uncertain system.

**The Neural Network Model PMN**

The model PMN for the arc welding process WP and the measurement part MS, which contains computer vision systems, can be realized by back-propagation networks with four layers and nodes N1-N2-N3-N4. The mapping relationship of the model is described as

\[
y_M(t+1) = f_M(u(t), u(t-1), \ldots, u(t-m); y_M(t), \ldots, y_M(t-n))
\]

(2.3)

where defining

\[
x^T = [x_1, \ldots, x_N]^T = [u(t), \ldots, u(t-m)]
\]

\[
y_M(t), \ldots, y_M(t-n)]^T
\]

(2.4)

where \( m \) and \( n \) denote orders of the arc welding dynamic process, which could be roughly estimated by experiments on the system. The network structure of PMN is shown in Fig. 1.

The network functions of the PMN are described as follows:

\[
f_{li} = \frac{1}{1 + \exp[-\sum_{j=1}^{N2} W_{lij} x_j + q_{li}]} \quad l = 1, \ldots, N3
\]

(2.4)

\[
f_{2k} = \frac{1}{1 + \exp[-\sum_{j=1}^{N3} W_{2kj} f_{1k} + q_{2k}]} \quad k = 1, \ldots, N3
\]

(2.5)

**Table 1 — Simple Fuzzy Control**

<table>
<thead>
<tr>
<th>E</th>
<th>-3</th>
<th>-2</th>
<th>-1</th>
<th>0</th>
<th>+1</th>
<th>+2</th>
<th>+3</th>
</tr>
</thead>
<tbody>
<tr>
<td>EC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
</tr>
<tr>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
</tr>
<tr>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>+1</td>
<td>+1</td>
<td>+1</td>
<td>+1</td>
<td>+1</td>
<td>+1</td>
<td>+1</td>
<td>+1</td>
</tr>
<tr>
<td>+2</td>
<td>+2</td>
<td>+2</td>
<td>+2</td>
<td>+2</td>
<td>+2</td>
<td>+2</td>
<td>+2</td>
</tr>
<tr>
<td>+3</td>
<td>+3</td>
<td>+3</td>
<td>+3</td>
<td>+3</td>
<td>+3</td>
<td>+3</td>
<td>+3</td>
</tr>
</tbody>
</table>

**Table 2 — Fuzzy Control**

<table>
<thead>
<tr>
<th>E</th>
<th>-3</th>
<th>-2</th>
<th>-1</th>
<th>0</th>
<th>+1</th>
<th>+2</th>
<th>+3</th>
</tr>
</thead>
<tbody>
<tr>
<td>EC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
</tr>
<tr>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
</tr>
<tr>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>+1</td>
<td>+1</td>
<td>+1</td>
<td>+1</td>
<td>+1</td>
<td>+1</td>
<td>+1</td>
<td>+1</td>
</tr>
<tr>
<td>+2</td>
<td>+2</td>
<td>+2</td>
<td>+2</td>
<td>+2</td>
<td>+2</td>
<td>+2</td>
<td>+2</td>
</tr>
<tr>
<td>+3</td>
<td>+3</td>
<td>+3</td>
<td>+3</td>
<td>+3</td>
<td>+3</td>
<td>+3</td>
<td>+3</td>
</tr>
</tbody>
</table>

**Table 3 — Fuzzy Control**

<table>
<thead>
<tr>
<th>E</th>
<th>-3</th>
<th>-2</th>
<th>-1</th>
<th>0</th>
<th>+1</th>
<th>+2</th>
<th>+3</th>
</tr>
</thead>
<tbody>
<tr>
<td>EC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
</tr>
<tr>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
</tr>
<tr>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>+1</td>
<td>+1</td>
<td>+1</td>
<td>+1</td>
<td>+1</td>
<td>+1</td>
<td>+1</td>
<td>+1</td>
</tr>
<tr>
<td>+2</td>
<td>+2</td>
<td>+2</td>
<td>+2</td>
<td>+2</td>
<td>+2</td>
<td>+2</td>
<td>+2</td>
</tr>
<tr>
<td>+3</td>
<td>+3</td>
<td>+3</td>
<td>+3</td>
<td>+3</td>
<td>+3</td>
<td>+3</td>
<td>+3</td>
</tr>
</tbody>
</table>
\[
\begin{align*}
\gamma_{M}(t+1) &= \left[1 + \exp\left(-\frac{N^2 \sum W_{jk}(t) f_{k}(t)}{2}\right)\right]^{-1} \\
= h_{M}\left(f_{k}(t)+1\right) \\
\end{align*}
\]

where \( W_{jk} \) and \( x_{i} \) denote connection weights and output functions for PMN first-layer network, \( W_{jk} \) and \( f_{k} \) for the second-layer, and \( W_{jk} \) and \( f_{k} \) for the third-layer, respectively.

**Self-Learning Algorithms of Fuzzy Neural Controller**

Self-learning algorithms for the fuzzy controller FC and neural network model PMN are developed as follows:

\(<1>\) The index of control error
\[
E_{c} = \sum_{t=1}^{N} \frac{[D - y(t+1)]^2}{2} \\
\]

\(<2>\) The index of model error
\[
E_{m} = \sum_{t=1}^{N} \frac{y_{m}(t+1) - y_{m}(t+1)^2}{2} \\
\]

\(<3>\) Learning algorithms of the model PMN.

Off-line and on-line learning algorithms can be used to modify parameters of the PMN network. We can obtain the initial weights of the PMN by batch sample data \( \{u(t),y(t)\} \). The off-line learning results of the PMN model can be used as a reference model for the uncertain controlled objects. Using the on-line learning, we can modify network weights of the PMN in real-time by the index (3.2) and the principle of error gradient descent, that is

\[
\begin{align*}
\Delta W(t+1) &= -\eta \frac{\partial W(t)}{\partial E(t)} \\
W(t+1) &= W(t) + \eta \Delta W(t) \\
\end{align*}
\]

The learning algorithms for the PMN are briefly described below.

**Defining**
\[
\begin{align*}
\eta_{i}(t) &= f_{i}(t)(1-f_{i}(t)) x_{i}(t) \\
\eta_{j}(t) &= f_{j}(t)(1-f_{j}(t)) \sum_{k=1}^{N^2} W_{jk}(t) f_{k}(t) \\
\end{align*}
\]

then weights modified as

\[
\begin{align*}
\Delta W_{jk}(t) &= h_{j}(t) \frac{\partial E(t)}{\partial W_{jk}(t)} \\
W_{jk}(t+1) &= W_{jk}(t) + \Delta W_{jk}(t) \\
\Delta W_{k}(t) &= h_{k}(t) \frac{\partial E(t)}{\partial W_{k}(t)} \\
W_{k}(t+1) &= W_{k}(t) + \Delta W_{k}(t) \\
\end{align*}
\]

where \( h_{j},h_{k} \in (0,1) \) are learning factors and momentum factors, respectively. The Equations 3.3-3.9 are one-step learning algorithms for PMN networks in a controlling period.

\(<4>\) Adaptive-modifying of tuning parameters \( a(t), b(t) \) of the FC.

Assuming the PMN network parameters are known variables that have been obtained by off-line last one-step learning, we have the following algorithms to modify tuning parameters \( a(t), b(t) \) of the fuzzy controller FC.

\[
\begin{align*}
a(t+1) &= a(t) + \Delta a(t) \\
b(t+1) &= b(t) + \Delta b(t) \\
\Delta a(t) &= -h_{a} \frac{\partial a(t)}{\partial a(t)} \\
\Delta b(t) &= -h_{b} \frac{\partial b(t)}{\partial b(t)} \\
\end{align*}
\]

learning factors \( h_{a},h_{b} \in (0,1) \)

\[
\begin{align*}
\frac{\partial a(t)}{\partial a(t)} &= \left[c - \frac{\partial}{\partial a(t)} \left(D - y_{M}(t+1) + e\right)\right] \\
\frac{\partial b(t)}{\partial b(t)} &= \left[c - \frac{\partial}{\partial b(t)} \left(D - y_{M}(t+1)\right)\right] \\
\frac{\partial y_{M}(t+1)}{\partial a(t)} &= \left[c - \frac{\partial}{\partial a(t)} \left(D - y_{M}(t+1)\right)\right] \\
\frac{\partial y_{M}(t+1)}{\partial b(t)} &= \left[c - \frac{\partial}{\partial b(t)} \left(D - y_{M}(t+1)\right)\right] \\
\frac{\partial y_{M}(t+1)}{\partial a(t)} &= \left[c - \frac{\partial}{\partial a(t)} \left(D - y_{M}(t+1)\right)\right] \\
\frac{\partial y_{M}(t+1)}{\partial b(t)} &= \left[c - \frac{\partial}{\partial b(t)} \left(D - y_{M}(t+1)\right)\right] \\
\frac{\partial y_{M}(t+1)}{\partial a(t)} &= \left[c - \frac{\partial}{\partial a(t)} \left(D - y_{M}(t+1)\right)\right] \\
\frac{\partial y_{M}(t+1)}{\partial b(t)} &= \left[c - \frac{\partial}{\partial b(t)} \left(D - y_{M}(t+1)\right)\right] \\
\end{align*}
\]
that is
\[a(t+1) = a(t) + h a (D - y(t+1))b(t)lE(t)
\]
\[-EC(t)l\partial u(t)] \tag{3.10}\]
where
\[
\delta_{a} / \partial u(t) = \frac{\partial a}{\partial b}(D - y(t+1))b(t)lE(t)
\]
\[-EC(t)l\partial u(t)] = \left\{ \begin{array}{l}
l_{0}(1-l_{0}) \sum_{i=1}^{N} \left[ \delta_{h} / \partial v_{i} \right] W_{h} h_{i} \left(1-l_{h} \right) \\
\phantom{l_{0}(1-l_{0}) \sum_{i=1}^{N} \left[ \delta_{h} / \partial v_{i} \right] W_{h} h_{i} \left(1-l_{h} \right)} \end{array} \right\}
\]
\[b(t+1) = b(t) + h b (D - y(t+1))lE(t)
\]
\[-EC(t)l\partial u(t)] \tag{3.11}\]

Equations 3.10–3.12 are one-step self-learning algorithms for the tuning parameters \(a(t), b(t)\) of the FC in a controlling period. Essentially, it is to regulate the fuzzy control rules as a human operator in real-time.

**Experimental Systems for Pulsed GTAW**

To investigate the feasibility of the above fuzzy neural control scheme for practical, uncertain objects, an experiment on control of weld bead width with the pulsed GTAW process was conducted. In this study, an attempt is made to combine the fuzzy logic and neural network techniques for the process dynamics.

**Components of Control Systems**

The block diagram for the control systems of the pulsed GTAW process is shown in Fig. 5. The systems are composed of an IBM-PC/AT 386 computer for self-learning control and image processing algorithms, an industrial TV camera used as a vision sensor to pick up the top image of the weld pool, an image interface unit, an image monitor and an alter/direct pulse arc welding power source. The photographs of the experiment system and sensor equipment are shown in Figs. 6 and 7, respectively. The TV camera was mounted on the back of the welding torch, at an angle of 30 deg with the horizon. The working platform with the workpiece holder could be regulated in a horizontal plane (500 x 200 mm), i.e., which is with two-degrees of freedom. The image of the weld pool was picked up by the TV camera and transformed to a digital image stored in the frame buffer of the image interface unit as a part of the computer memory by means of the segment mapping, which could be written/read randomly. The welding current was regulated by the interface unit of the welding power source, and regulation of welding travel speed was realized by a single chip computer system.

**Image Processing of the Face of the Weld Pool**

As is well known, the arc of the welding process makes illumination and irradiation of surfaces of the weld specimen and the weld pool complicated. Usually, the arc light is considered a strong interference while sensing surface changes on the weld pool. After investigating many experiments, the intensity of the arc light was found to change from strong to weak when the welding current transformed from the pulse peak value to the base value in pulsed GTAW. We chose the pulse ripple and time sequence of the welding current shown in Fig. 8 for the purpose of regulating and image processing of the welding process. At the very moment when the pulse peak value of the welding current was decreasing to

![Fig. 9 - Image of weld pool zone.](image1)

![Fig. 10 - The extracted bead width.](image2)

![Fig. 11 - A line gray distribution of image.](image3)

![Fig. 12 - Specimen size (mm).](image4)
At this moment, the weld pool began to cool down and solidify incompletely due to the heat inertia of the molten process. An image of the welding pool zone was shown in Fig. 9. The image consists of four parts from top to bottom: welding torch, tungsten electrode, weld pool and welded joint. A static digital image with various noises entered into the image interface. Using slide-median filtering algorithm of one-dimension, a line pixel gray distribution was obtained in a frame filtered image shown in Fig. 11. Searching the distributed zones of the pixel gray in the weld pool image, line by line, along the tungsten axis, the maximum width of the weld pool was obtained, which is indicated by a real-line crossing the weld pool in Fig. 10. To obtain a reliable weld bead width, the difference was used in the algorithm of one-order five-points to calculate the sudden change boundaries of the pixel gray distributions in the weld pool image and obtained the maximum weld bead width in real-time. The whole image processing period was approximate 0.7 s.

**Experiment Results**

**Modeling and Simulation of the Welding Process**

By analyzing the technology of the pulsed GTAW process and the test data from standard conditions, it is known that the main factors influencing weld pool changes are welding current and welding travel speed, with other parameters such as plate thickness and root opening being normal. Usually, a robotic welding system only regulates the welding current under a fixed welding travel speed and other parameters. For simplicity, we established a SISO model of the welding pool dynamics for the control of pulsed GTAW. The input and output of the model were the welding current and the weld bead width of the top of the welding pool, respectively. An experiment for testing the dynamics of the welding current/bead width process was completed under the following conditions: bead-on-plate welds on low-carbon steel plate of 2-mm thickness, 100 x 250 mm; tungsten electrode diameter of 3 mm; shielding gas of argon at 8 mL/min flow rate; constant travel speed of 9.5 cm/min; and arc voltage of 12 to 30 V DC. The parameters of welding current pulse series in Fig. 8 were designed as follows:

\[ T_v = 200 \text{ ms}, \quad I_v = 30 \text{ A}, \quad I_p = 350 \text{ ms}, \quad T_1 = 40 \text{ ms}, \quad T_2 = 650 \text{ ms}, \quad I_b = 20 \text{ A}, \quad T_g = 60 \text{ ms} \]

The pulse peak value of the welding current \( I_p \) is the regulating variable.

The test result for the process is shown in Fig. 13. Based on the above experimental data and using the batch testing data of input-output pairs and an off-line learning algorithm, a neural network model of the process as shown in Fig. 4 was obtained. Its nodes N1, N2, N3, and N4 are 5, 10, 10, and 1, respectively, which realizes the following mapping:

\[ Y_{W}(t+1) = f_{N_4}(u(t),u(t-1),u(t-2), Y_4(t), Y_4(t-1)) \]  
\[ Y_4(t+1) \]  

\[ (4.1) \]

The test result for the process is shown in Fig. 13. Based on the above experimental data and using the batch testing data of input-output pairs and an off-line learning algorithm, a neural network model of the process as shown in Fig. 4 was obtained. Its nodes N1, N2, N3, and N4 are 5, 10, 10, and 1, respectively, which realizes the following mapping:

\[ Y_{W}(t+1) = f_{N_4}(u(t),u(t-1),u(t-2), Y_4(t), Y_4(t-1)) \]  
\[ Y_4(t+1) \]  

\[ (4.1) \]

The test result for the process is shown in Fig. 13. Based on the above experimental data and using the batch testing data of input-output pairs and an off-line learning algorithm, a neural network model of the process as shown in Fig. 4 was obtained. Its nodes N1, N2, N3, and N4 are 5, 10, 10, and 1, respectively, which realizes the following mapping:

\[ Y_{W}(t+1) = f_{N_4}(u(t),u(t-1),u(t-2), Y_4(t), Y_4(t-1)) \]  
\[ Y_4(t+1) \]  

\[ (4.1) \]

\[ Y_4(t+1) \]  

\[ (4.1) \]

The test result for the process is shown in Fig. 13. Based on the above experimental data and using the batch testing data of input-output pairs and an off-line learning algorithm, a neural network model of the process as shown in Fig. 4 was obtained. Its nodes N1, N2, N3, and N4 are 5, 10, 10, and 1, respectively, which realizes the following mapping:

\[ Y_{W}(t+1) = f_{N_4}(u(t),u(t-1),u(t-2), Y_4(t), Y_4(t-1)) \]  
\[ Y_4(t+1) \]  

\[ (4.1) \]

The test result for the process is shown in Fig. 13. Based on the above experimental data and using the batch testing data of input-output pairs and an off-line learning algorithm, a neural network model of the process as shown in Fig. 4 was obtained. Its nodes N1, N2, N3, and N4 are 5, 10, 10, and 1, respectively, which realizes the following mapping:

\[ Y_{W}(t+1) = f_{N_4}(u(t),u(t-1),u(t-2), Y_4(t), Y_4(t-1)) \]  
\[ Y_4(t+1) \]  

\[ (4.1) \]

The test result for the process is shown in Fig. 13. Based on the above experimental data and using the batch testing data of input-output pairs and an off-line learning algorithm, a neural network model of the process as shown in Fig. 4 was obtained. Its nodes N1, N2, N3, and N4 are 5, 10, 10, and 1, respectively, which realizes the following mapping:

\[ Y_{W}(t+1) = f_{N_4}(u(t),u(t-1),u(t-2), Y_4(t), Y_4(t-1)) \]  
\[ Y_4(t+1) \]  

\[ (4.1) \]

The test result for the process is shown in Fig. 13. Based on the above experimental data and using the batch testing data of input-output pairs and an off-line learning algorithm, a neural network model of the process as shown in Fig. 4 was obtained. Its nodes N1, N2, N3, and N4 are 5, 10, 10, and 1, respectively, which realizes the following mapping:

\[ Y_{W}(t+1) = f_{N_4}(u(t),u(t-1),u(t-2), Y_4(t), Y_4(t-1)) \]  
\[ Y_4(t+1) \]  

\[ (4.1) \]

Experimental Results on the Control of Pulsed GTAW

Based on the system scheme in Fig. 5, experiments to control weld bead width with pulsed GTAW have been performed on the system shown in Fig. 6. The bead-on-plate experiments were conducted under the following conditions: The specimens for the test were low-carbon steel plate of 2 mm thickness shown in Fig. 12; a dumbbell specimen was used for simulating sudden changes in heat transfer or heat sink conditions during the weld process; the tungsten electrode diameter was 3 mm with an acute angle of 45 deg and the tip diameter not larger than 0.5 mm; the shielding gas of argon was at 8 mL/min flow rate; constant travel speed of 9.5 cm/min; and an arc voltage of 12 to 30 V DC. The welding current pulse series shown in Fig. 8 was chosen for the experiment. The parameters of the pulsed current were designed as:

\[ T_v = 200 \text{ ms}, \quad I_v = 30 \text{ A}, \quad I_p = 350 \text{ ms}, \quad T_1 = 40 \text{ ms}, \quad T_2 = 650 \text{ ms}, \quad I_b = 20 \text{ A}, \quad T_g = 60 \text{ ms} \]

and the pulse peak value of the welding current \( I_p \) is the regulating variable. In the control scheme for pulsed GTAW are shown in Fig. 14.

The simulation of the process shows that the control system is able to adaptively attain the optimized control and desired welding specification by the self-learning algorithm, i.e., welding current peak value is approximately 180 A, weld bead width \( y(t) \) to 6 mm, and tuning factors \( a(t) \) to 0.63 and \( b(t) \) to 0.67.
process by the self-learning fuzzy neural control scheme presented in this paper, Fig. 16A shows that the results of controlling weld bead width are satisfactory. The control error is maintained within 0.3 mm and the regulation of welding current (peak value) in Fig. 16B is identical with experiments with a human operator, i.e., the fuzzy controller is adaptively regulating welding current (peak value) in the narrower section of the specimen due to the adverse changes of scattered heat conditions, which is just similar to the behavior of a skilled welder. With real-time learning of the process, the tuning parameters $a(t)$ and $b(t)$ of the fuzzy controller FC are adaptively modified as seen in Fig. 16C and D. Figures 15B, 16E and 16F are photographs of the weld bead.

In this study, we have only investigated the controllability of welding current and weld face width, and only indirectly obtained a desired root surface width. If the requirement of the underside formation is critical, the design scheme shown in Fig. 17 is for the control of the root surface width. BWMN indicates the back weld width network model of pulsed GTAW combined with the infor-
Fig. 15 — Constant normal welding. A — bead face width; B — top of the specimen.

Fig. 16 — Welding results. A — bead face width; B — welding current; C — tuning factor a(t); D — tuning factor b(t); E — top of the specimen; F — underside of the specimen.
Formation of weld pool half-length $h(t)$.

If the welding travel speed is chosen as the regulating variable for control of desired weld bead width, the model of the process will appear as a time-delay due to the mechanical inertia of the regulating speed mechanism. Such a system with uncertainties and time delays is discussed in Ref. 25.

Discussion

The experimental results to control welds with changing heat transfer circumstances demonstrated that the self-learning fuzzy neural control scheme shown in Fig. 1 is feasible for controlling weld pool dynamics for pulsed GTAW. Comparing Fig. 15 with Fig. 16, the results show that the regulating actions of the control system are similar to the intelligent behavior of a skilled welder.

The results are mainly influenced by the time needed for completing the control algorithm and image processing, which could be improved by parallel processing of the neural networks and by enhancing computational speed.

Conclusions and Further Work

The following conclusions resulted from the above investigation on the application of fuzzy logic and artificial neural networks to the control of the welding process:

1) The self-learning fuzzy neural control scheme presented in this paper can be used for real-time control of the bead width with the pulsed GTAW.

2) The self-learning fuzzy neural control in the experimental system has exhibited satisfactory adaptive behavior to control a variable process such as arc welding. Artificial neural networks can also be used to model the dynamics of the welding process.

3) The arc can be utilized to illuminate the weld pool and the weld face during the transformation of the pulse peak value of the welding current to the base value.

4) Computer vision techniques can be used to provide dynamic information on the weld pool in feedback control of the arc welding process.

Further investigations will be made to perfect the fuzzy neural network to control weld quality such as fine formation of the weld bead and full joint penetration with pulsed GTAW. A multivariable feedback control system will be designed for the arc welding process. Action entries of the control system will contain welding current, arc voltage and travel speed. Image sensor and processing techniques will be improved correspondingly.

Acknowledgment

This work was supported by the National Natural Science Foundation of China, No. 59575057 and No. 59635160.

References


