Fuzzy Control of Temperature in a Semiconductor Processing Furnace

Charles L. Ramiller  
Member, IEEE  
chuck@ramiller.com

Mo-Yuen Chow  
Senior Member, IEEE  
chow@ncsu.edu

Richard T. Kuehn  
Member, IEEE  
rkuehn@ncsu.edu

North Carolina State University, Box 7911, Raleigh, NC 27695

Abstract — The ability to control temperature in semiconductor processing furnaces is critical to the manufacturing of semiconductor devices. In this paper, a fuzzy PI controller is described that takes into account some of the unique characteristics of such a furnace. Entries in the rule base are used to prevent integrator windup, and a fuzzy gain scheduler allows the controller to be tuned once and used over the whole operating temperature range of the system. Substantial improvements are shown for settling times following both large and small step changes in reference setpoint.

I. INTRODUCTION

Multi-zone batch furnace systems are widely used in the fabrication of integrated circuits. Typical silicon wafer processing applications include diffusion, oxidation, chemical vapor deposition (CVD), and annealing [1]. Processing temperatures for these applications start at about 300 °C and go as high as 1300 °C, but the primary range of interest is currently 500-1000 °C. It is extremely important to minimize the temperature variations within each wafer, wafer-to-wafer, and batch-to-batch. Most of the chemical processes used in device processing are strongly temperature dependent and some are purposely driven into this mode. Variations in temperature produce film thickness nonuniformities which can directly affect device performance and reliability. In some instances, such as during diffusion or any process that occurs late in the fabrication sequence, it is also desirable to keep the thermal histories of all the individual devices in an entire batch of wafers as nearly identical as possible.

The furnace element and process chamber of a vertical furnace system are shown schematically in Figure 1. The product wafer load consists of 100-150 silicon wafers placed horizontally into a slotted quartz boat or tower. To reduce heat losses, the lower third of the tower contains additional filler wafers that function as radiation shields. The furnace element is divided into zones in order to compensate for end losses and control the temperature profile over the length of the load. A “spike” thermocouple is located in each zone and produces a reading that is representative of the furnace wall temperature. Three or more “profile” thermocouples protected by a quartz sheath are positioned adjacent to the tower where they can be used to monitor the temperature of the wafer load.

The most commonly used method for controlling furnace temperatures is PID control applied on a zone by zone basis. If the steady state offsets between the thermocouples and the wafer edge temperatures are characterized over the operating range of the system, either set of thermocouples can be used to indirectly control the wafer temperatures. Alternatively, the furnace is a good application for dual loop cascade control where a faster inner control loop is nested inside a slower outer one. When cascade control is used, the profile thermocouple is used for primary control and the output of the profile loop modifies or replaces the spike loop setpoint. The individual loops can be tuned on-line using the Ziegler-Nichols (Z-N) frequency response method [2,3].

If the PID controllers are properly set up and tuned, steady state furnace temperatures can be controlled to within 0.1 °C, but PID control has a number of shortcomings. The thermal behavior of the furnace is nonlinear and its dynamic characteristics change with operating temperature. This makes it necessary to repeat the tuning process for each control loop at several temperatures and usually to implement a crude form of gain scheduling in the process recipes. In addition, Ziegler-Nichols tuning does not yield a desired step response. Instead, it produces an underdamped response with a degree of overshoot that would be excessive if it were not reduced by actuator saturation.

Efforts to develop improved control strategies have focused on precise thermal modeling and the application of optimal and model-based control methods [4-7]. These approaches suffer from similar drawbacks. While steady
state performance is satisfactory, it is necessary to use state space systems that have been linearized at a specific operating temperature. The whole design process then has to be repeated at other operating points. A more practical solution would take the nonlinear nature of the system into account and could be used hands-off over the whole operating temperature range of the system.

II. FUZZY CONTROL

Fuzzy sets were first introduced by Zadeh in 1965 as a way to extend ordinary set theory to include real world objects that do not meet precise membership criteria [8]. This concept was later expanded to include fuzzy algorithms or rule bases consisting of conditional statements of the form “IF A THEN B” where A and B contain linguistic variables such as “small” or “very large” [9]. Finally, a systematic methodology was introduced for using linguistic variables and fuzzy algorithms to solve a wide range of complex problems that are difficult or impossible to solve with precise quantitative methods [10].

Mamdani reported the successful use of fuzzy logic for computer control of a laboratory steam engine in 1974 [11]. The results were at least as good as those obtained with classical controllers, and the control system was easier to implement and understand. Since then, fuzzy control has been used in a wide range of applications and has proven to be well suited for complex nonlinear systems, particularly when human knowledge or experience can be incorporated into the control strategy [12].

DC motors, for example, exhibit nonlinear characteristics such as backlash, dead zones, saturation, and nonlinear friction. Significant advancements have been made in DC motor control by combining fuzzy logic with linear PI control in order to compensate for some of these effects [13]. Fuzzy logic has also been paired with PID controllers to implement fuzzy gain scheduling and self-tuning schemes [14-16].

A commonly used example of a discrete time fuzzy logic controller is shown in block diagram form in Fig. 2. The controller’s inputs are the error, E, which is defined as the reference setpoint minus the measured value of the process variable, and the change in that error, CE. The output is the change in control effort, CU. The inputs are crisp numerical values and therefore must be converted to linguistic terms or fuzzified before the rule base can be executed. Similarly, the output must be converted back to a crisp value or defuzzified before it can be applied to the process. The rule base consists of a series of statements in the antecedent-consequent form

\[ \text{IF } E \text{ is } A \text{ AND } CE \text{ is } B \text{ THEN } CU \text{ is } C \]  

where A, B and C are linguistic or fuzzy quantities such as “positive small” or “negative large”. Each rule is evaluated to determine its contribution to the output and an inference method is used to combine those contributions to form the total output.

An input is fuzzified using a set of membership functions spanning its expected range or “universe of discourse”. Each membership function describes the degree to which a crisp input belongs to a class by assigning it a membership value between zero and one. As an example, Fig. 3 illustrates a simple case where the linguistic variables zero (ZE) and positive small (PS) are represented by triangular membership functions. As shown, an input value of 0.25 would have the following membership values in ZE and PS:

\[ \mu_{ZE}(0.25) = 0.75 \]
\[ \mu_{PS}(0.25) = 0.25 \]

If a scaling or normalizing factor is used, any expected input value can be fuzzified using a set of membership functions that span the interval [-1, 1] as shown in Fig. 4. In this example, each membership function corresponds to one of the following linguistic variables:

NL: negative large
NM: negative medium
NS: negative small
ZE: zero
PS: positive small
PM: positive medium
PL: positive large

While triangular membership functions are shown in the figure, other shapes may be used, and trapezoidal and

![Fig. 2. Block diagram of a fuzzy controller.](image)

![Fig. 3. Input fuzzification example using two triangular membership functions.](image)

![Fig. 4. A set of seven triangular membership functions spanning a normalized input range of -1 to 1.](image)
gaussian functions are commonly encountered.

A simple rule base contains a rule structured like (1) for each possible combination of the inputs. For a two input controller, the rules are usually summarized in the tabular form shown in Fig. 5. The contribution of each rule to the output is most often determined using max-min or “Mamdani” inference. This process is illustrated graphically in Fig. 6 for the two rules represented by the shaded boxes in Fig. 5:

IF E is ZE AND CE is PS THEN CU is PS.
IF E is PS AND CE is PS THEN CU is PM.

For each input, a membership value is computed for the corresponding linguistic variable used in the rule. The membership function for the output class specified by the rule is then truncated by the minimum of the two membership values. This operation is the fuzzy equivalent of the logical AND connective used in the rule. The contributions from all of the rules are then aggregated by taking the maximum value of the individual contributions at each value of the output’s universe of discourse. This forms an envelope from all of the overlapping individual contributions as shown at the bottom of Fig. 6 and is the fuzzy equivalent of using a logical OR to connect the rules. The output is then defuzzified by computing the center of area or centroid of the envelope using

\[ CU_{OUT} = \frac{\int \mu(CU) \cdot CU \cdot dCU}{\int \mu(CU) \cdot dCU} \quad (2) \]

The controller just described is one example selected from an unlimited number of ways to use fuzzy logic in control applications. There are many different membership functions, inference methods and defuzzification techniques available. In addition, the rule base may be one-dimensional or multi-dimensional and may express a set of linguistic rules or a desired mapping of inputs to outputs.

III. APPLYING FUZZY CONTROL TO THE FURNACE

A continuous time linear PI controller is described by

\[ u(t) = \frac{1}{PB} \begin{bmatrix} e(t) + \frac{1}{T_I} \int_0^t e(\tau) d\tau \end{bmatrix} \quad (3) \]

where

- \( e(t) = \) error (reference – measured process variable),
- \( u(t) = \) controller output,
- \( PB = \) proportional band (the amount of error that will cause the output to go to its maximum value),
- \( T_I = \) integral time.

A discrete time equivalent of this equation is

\[ U(k) = \frac{1}{PB} \begin{bmatrix} E(k) + T_S \frac{1}{T_I} E(k) + I(k-1) \end{bmatrix} \quad (4) \]

or, in terms of the changes in error and output,

\[ CU(k) = \frac{1}{PB} \begin{bmatrix} CE(k) + T_S \frac{1}{T_I} E(k) \end{bmatrix} \quad (5) \]

where

- \( E = \) error,
- \( CE = \) change in error = \( E(k) - E(k-1) \),
- \( CU = \) change in output = \( U(k) - U(k-1) \),
- \( I(k-1) = \) previous value of the integral term,
- \( T_S = \) sampling interval,
- \( kT_S = \) current time,
- \( k = 0, 1, 2, \ldots \)

It can be seen from (5) that the example fuzzy controller described in the previous section is essentially a discrete time PI controller. The proportional contribution to \( CU \) is derived from \( CE \) and the integral contribution is derived from \( E \), which may seem counterintuitive when compared to the continuous time expression. It has been shown in detail that linear PI controllers are actually a special case of the example fuzzy controller [17]. This means that some of the problems experienced with PI control, such as integrator windup, will also occur with fuzzy control [18]. On the other hand, a well designed fuzzy controller should provide unique capabilities while still allowing the use of
practical knowledge gained from applying linear controllers to the furnace.

Fig. 7 is a block diagram of a fuzzy PI temperature controller that includes gain scheduling. The proportional term is derived directly from the error signal as in (4). The integral term is computed from the error as in (5) and produces a change in output that is accumulated and then summed with the proportional output. An "integral band", IB, is used as a scaling factor to define the range of error within which the integrator is active in the same way that the proportional band, PB, defines the region within which the proportional controller functions. Entries in the rule base prevent windup by causing the incremental output of the integrator to go to zero when the error is outside the integral band.

For the proportional controller, evenly distributed triangular membership functions are used to span the normalized input interval -1 to 1 as shown in Fig. 8 (a). Singleton are used for the output membership functions as shown in Fig. 8 (c). The inference method is the same as previously described in the example except that the outputs are defuzzified using the weighted averages,

\[ U = \frac{\sum \mu(U) \cdot U}{\sum \mu(U)} \quad (6) \]

rather than the centroid in (2). A one-dimensional rule base containing nine rules was used to generate each of the linear and nonlinear control response curves shown in Fig 9. This approach was taken in order to determine the effect of changing the sensitivity of the controller near the origin. With minor changes to the rules and the membership functions, any desired response curve could be implemented.

For the integral controller, an additional trapezoidal membership function was added at each end of the normalized input interval as shown in Fig. 8 (b). A very abrupt transition was created and the new fuzzy variables were mapped to zero in the rule base. Any input to the integral controller that is outside the integral band produces an incremental output of zero as shown by the response curves in Fig. 10.

The desired response for the gain scheduler was derived from the critical gains and periods obtained by applying the Ziegler-Nichols tuning technique at 200 °C increments over the range 200-1000 °C. The input and output...
output membership functions shown in Fig. 11 were used with the following rules:

IF $T$ is: 200 400 600 800 1000
THEN $GS$ is: 2.09 1.51 1.00 0.57 0.19

In the controller, the gain scheduling factor, $GS$, is applied to both the proportional band, $PB$, and the integral time, $TI$, as can be seen in Fig. 7. This was based on the observation that the critical values of $PB$ and $TI$ scaled almost identically as a function of temperature.

IV. RESULTS AND DISCUSSION

The fuzzy PI controller was evaluated using a lumped heat capacitance model for the dominant middle zone of the furnace. The model was derived from measured results obtained from an actual furnace and includes a temperature dependent delay that causes the dynamic behaviour of the simulation to closely approximate that of the real system.

Two criteria were used to determine the overall effectiveness of the controller. A 0-620 °C step was used to approximate the large initial transient that occurs when a cold wafer load is initially inserted into the furnace and heated to the process temperature. A 10 °C step near the process temperature was used to assess the capability of the controller to respond to small disturbances. In both cases, the controlled variable was the spike temperature, and the measure of performance was the time required for the profile temperature to stabilize to within 0.1 °C of the steady state reference setpoint following the step change.

A linear PI controller and three fuzzy PI controllers were individually tuned at 600 °C using the Ziegler-Nichols method. For the linear controller, windup was prevented by freezing the integrator whenever the output was saturated. Each fuzzy controller used a different response curve from Fig. 9, and its integral band was quickly optimized as part of the tuning process. The performance of all four controllers is compared in Table I. For the larger step, the fuzzy controllers performed equally well and their settling times are 8-10% lower than that of the linear controller. All of the controllers produced nearly identical settling times in response to the smaller step. Although this is not an exhaustive test, it does suggest that the shape of the response curve is not as important as careful tuning and optimization.

Table II shows the results of optimizing both the integral band and integral time over a range of values for the proportional band. In this experiment, the fuzzy controller used the rule base that produced the linear response curve in Fig. 9. Best case settling times for both the large and small steps are about 26% faster than for the linear PI controller but occur at different values of proportional gain. It might be beneficial to use one set of parameters only during the initial stabilization and another for all subsequent processing. However, the parameters that produce the best case small step response also result in an excellent settling time from the large step and may be a good compromise.

The fuzzy gain scheduler was evaluated by tuning the system at 600 °C and then observing the response to steps from 0 °C to various temperatures in the range 400-1000 °C. Its main effect was improved response at higher temperatures as shown in Table III, although it also improves stability at lower temperatures. Better results were obtained when the integral band was also optimized.

### Table I
COMPARISON OF PERFORMANCE WITH Z-N PARAMETERS.

<table>
<thead>
<tr>
<th>Controller</th>
<th>PB (°C)</th>
<th>TI (min)</th>
<th>IB (°C)</th>
<th>0–620 °C (min)</th>
<th>610–620 °C (min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
<td>47.3</td>
<td>2.96</td>
<td>-</td>
<td>55.8</td>
<td>16.4</td>
</tr>
<tr>
<td>Fuzzy-linear</td>
<td>47.3</td>
<td>2.96</td>
<td>57.5</td>
<td>50.4</td>
<td>16.4</td>
</tr>
<tr>
<td>Fuzzy-higher slope</td>
<td>93.9</td>
<td>2.96</td>
<td>72.0</td>
<td>50.7</td>
<td>16.4</td>
</tr>
<tr>
<td>Fuzzy-lower slope</td>
<td>23.3</td>
<td>2.96</td>
<td>41.5</td>
<td>51.3</td>
<td>16.5</td>
</tr>
</tbody>
</table>

### Table II
OPTIMIZED FUZZY CONTROLLER PERFORMANCE.

<table>
<thead>
<tr>
<th>PB (°C)</th>
<th>TI (min)</th>
<th>IB (°C)</th>
<th>0–620 °C (min)</th>
<th>610–620 °C (min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>45</td>
<td>2.91</td>
<td>73</td>
<td>45.0</td>
<td>17.0</td>
</tr>
<tr>
<td>55</td>
<td>2.45</td>
<td>71</td>
<td>45.4</td>
<td>17.2</td>
</tr>
<tr>
<td>65</td>
<td>2.31</td>
<td>74</td>
<td>41.3</td>
<td>16.4</td>
</tr>
<tr>
<td>75</td>
<td>2.26</td>
<td>69</td>
<td>44.0</td>
<td>15.0</td>
</tr>
<tr>
<td>85</td>
<td>2.19</td>
<td>73</td>
<td>43.0</td>
<td>12.2</td>
</tr>
<tr>
<td>95</td>
<td>2.28</td>
<td>75</td>
<td>48.0</td>
<td>17.4</td>
</tr>
</tbody>
</table>

### Table III
FUZZY GAIN SCHEDULER PERFORMANCE.

<table>
<thead>
<tr>
<th>Step Change (°C)</th>
<th>Fuzzy-linear, no gain sched</th>
<th>Fuzzy-linear, gain sched</th>
<th>Fuzzy-linear, optimized IB, gain sched</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-400</td>
<td>119</td>
<td>115</td>
<td>112</td>
</tr>
<tr>
<td>0-600</td>
<td>57</td>
<td>56</td>
<td>50</td>
</tr>
<tr>
<td>0-800</td>
<td>76</td>
<td>61</td>
<td>56</td>
</tr>
<tr>
<td>0-1000</td>
<td>91</td>
<td>87</td>
<td>69</td>
</tr>
</tbody>
</table>
at each temperature. This indicates that a second gain scheduler applied to the integral band is needed to obtain the best overall performance over the full operating temperature range.

V. CONCLUSIONS

A fuzzy PI controller has been described which includes a fuzzy gain scheduler and uses entries in the rule base to prevent integrator windup. In general, its performance exceeds that of a linear PI controller.

A brief comparison using linear and nonlinear response curves produced nearly identical results indicating that there may not be a significant effect as long as the controller is well tuned. A systematic study of the effects of nonlinear and asymmetric response curves is still needed, however.

When all of the fuzzy controller's parameters are simultaneously optimized, substantial improvements in the settling time following a step change in reference setpoint are realized. There is some advantage to using different controller parameters for large and small steps, but excellent results can be obtained for both by compromising on a single configuration.

The gain scheduler improves performance over the whole operating temperature range. It is most effective when the fuzzy controller's integral band has been optimized at each temperature which indicates that a second gain scheduler is needed.

Based on these results, a fuzzy PI or PID controller combined with two fuzzy gain schedulers would produce superior performance when operated hands-off over the full temperature range of the system.

VI. REFERENCES


