Chapter 6: Experiments and Results

6.1. Introduction

This chapter presents extensive experiments and simulations in order to show the effectiveness of our proposed controller design. These experiments are divided into two main groups. First group of experiments shows the accuracy and validity of qualitative information of process responses and second group of experiments describes the performance and flexibility of our proposed controller. To demonstrate the effectiveness of our proposed controller design, four types of process are used for illustration. They are:

(1) First-order plus dead time process

\[ G_1(s) = \frac{e^{-\theta s}}{(s + a)} \]  

(6.1)

(2) Second-order plus dead time process

\[ G_2(s) = \frac{e^{-\theta s}}{(s + a)^2} \]  

(6.2)

(3) Third-order process

\[ G_3(s) = \frac{a}{(s + 0.5)(s^2 + 1.64s + 8.456)} \]  

(6.3)
(4) Higher-order process

\[ G_4(s) = \frac{a}{(s+1)(s+3)^3} \]  \hspace{1cm} (6.4)

The required specifications for the experiments are given in Table 6-1 [Zhao93, Ambrosio02].

Table 6-1. Required data for our experiments.

<table>
<thead>
<tr>
<th></th>
<th>(G_1(s))</th>
<th>(G_2(s))</th>
<th>(G_3(s))</th>
<th>(G_4(s))</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proportional gain (k_p)</td>
<td>4.8021</td>
<td>3.2448</td>
<td>2.19</td>
<td>3.1806</td>
</tr>
<tr>
<td>Integral gain (k_i)</td>
<td>13.34</td>
<td>2.207</td>
<td>2.1262</td>
<td>2.356</td>
</tr>
<tr>
<td>Derivative gain (k_d)</td>
<td>0.432</td>
<td>1.19268</td>
<td>0.565</td>
<td>1.0718</td>
</tr>
<tr>
<td>Sampling time TS</td>
<td>0.04 Sec</td>
<td>0.01 Sec</td>
<td>0.01 Sec</td>
<td>0.01 Sec</td>
</tr>
<tr>
<td>Delay parameter (\beta)</td>
<td>0.2</td>
<td>0.4</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Process parameter (a)</td>
<td>1</td>
<td>1</td>
<td>4.228</td>
<td>27</td>
</tr>
</tbody>
</table>

6.2. Knowledge-based Modular Neural Network Classifier Test

As we described in earlier chapters, the appropriate setting of controller parameters/structure are different based on the nature of process responses or control objectives. In Chapter 3, we decomposed the whole process response into four classes (rise time state, oscillation state, disturbance state and steady-state) in order to provide qualitative information to adaptive mechanism. Our proposed knowledge-based modular neural network classifier provides online this qualitative information of process response. It is used to switch the appropriate tuning strategy among a variety of tuning strategies.

Our aim of online process response estimation (qualitative information) is to apply in adaptive application. The stability of that kind of control system relies on estimation accuracy because it decides the appropriate tuning strategy and setting of
parameters at that moment of time. We observe the accuracy and validity of proposed knowledge-based modular neural network classifier based on two types of experiments: (i) accuracy test and (ii) refinement algorithm test.

6.2.1. Accuracy test

Fig. 6-1. The comparison of process response and our classifier's estimation (a) \( G_1(s) \); (b) \( G_2(s) \); (c) \( G_3(s) \) and (d) \( G_4(s) \).
In PID control system, online qualitative process response estimation would improve performance because there is a connection between parameters of PID controller and process responses (described in section 2.1.). Our goal of online qualitative estimation of process response is to switch to appropriate tuning strategy. There are seven different tuning strategies in our proposed modular adaptive mechanism (described in section 5.2) and they are developed based on the frequency response method and the nature of PID controller parameters in frequency domain.

In our work, a modular neural network classifier estimates online process response qualitatively. Firstly, we analyze the estimation accuracy of our proposed knowledge-based modular neural network classifier (described in section 3.2 and section 5.2).

Figures 6-1(a), (b), (c), and (d) show the comparison of actual process responses for four different types of process responses and our classifier's estimations. In Figures 6-1, Class 1 represents steady-state, Class 2 represents rise time state, Class 3 represents oscillation state and Class 4 represents disturbance state respectively. By observing these results, our classifier's estimation is almost same as the actual response. There are slight delays when process response changes from transient state to steady-state.

6.2.2. Refinement algorithm test

Although, our classifier design is based on well-known existing theory and principles, refinement of rules developed are necessary to guarantee the estimation accuracy. Our proposed knowledge-based refinement algorithm emphasizes on the set of discriminant function (Classifier-2 Block) that includes twenty individual classifiers (rules). There are two tests for incomplete domain knowledge and mismatch rule base, respectively. In this experiment and threshold value \( th_2 \) is defined as 0.9.

In order to examine incomplete domain knowledge, one rule (9th) is removed from Classifier-2 Block. During this test, our refinement algorithm examines the outputs of all remaining nineteen rules from Classifier-2 Block. Figure 6-2 shows activation of each rule at the specific time (the classifier is activated when its output is greater than threshold value \( th_2 \)).
Referring to this result, input subspace region is only partially matched with each rule at time $t = 2.36$ second (i.e., each rule produces less than threshold value $th2$). After which the knowledge refinement algorithm add on an independent rule for this input subspace based on target response (described in section 4.4).

![The Analyzing Incomplete Knowledge-Base](image)

Fig. 6-2. The analyzing the incomplete knowledge-base.

In test 2, one additional rule $(21^m)$ is added into Classifier-2 Block. During this test, our algorithm notices that more than one classifier are totally matched with the current input subspace at time $t = 2.28$ second (outputs of more than one rule are greater than threshold value $th2$). Our refinement algorithm debugs which classifiers are responsible for that and it selects a classifier that is most suitable in the current situation (described in Example 4.1). The main restriction of our proposed refinement algorithm is that it is not possible to use this in online applications because training data sets
(input/output data pairs) are required to detect these mismatch rules. The above two experiments reveal that the proposed refinement algorithm supports to improve the performance of classifier.

![Graph showing the analyzing the mismatched classifiers](image)

Fig. 6-3. The analyzing the mismatched rules (classifiers).

6.3. Performance Tests of Our Proposed Controller

In order to show effectiveness of our controller structure, the following three features are observed through several experiments.

1. First, the flexibility and transparency of our controller structure,
2. Second, the transient and steady-state response of our design and compare it with the other reported work (it includes set point tracking, disturbance rejection, stability and steady-state accuracy) and
3. Third, the robustness (sensitivity) of our controller structure.
6.3.1. Flexibility and transparency characteristics test

Flexibility and transparency are important characteristics for future high performance control system. Therefore, the observations of these characteristics are important in the controller design. In our proposed approach, there are seven fuzzy inference systems in modular structure gain scheduling mechanism and each fuzzy inference system has own tuning strategy (described in section 5.5). In this subsection, we analyze flexibility and transparency characteristics of proposed controller design based on processes $G_1(s)$ (Equation 6.1) and $G_2(s)$ (Equation 6.2) viz on our seven fuzzy inference system.

Fig. 6-4. The effect of fuzzy inference system $f_i$ (proportional gain) (a) in a delay period, (b) far from the set-point and (c) near the set-point.
Firstly, we analyze the effect of fuzzy inference system $f_1$ that is responsible to update proportional gain. The results (shown in Figures 6-4) show that high proportional gain in a delay period (tuning strategy $T1$) can cause oscillatory response. When process response is far from the set-point (tuning strategy $T2$), high proportional gain is needed to improve performance. Near the set-point (tuning strategy $T3$), low or medium proportional gain can improve stability without affecting the speed of the process response. It is noticed that the speed of process response can vary based on the different setting of proportional gain especially when process response is far from the set-point (Figure 6-4(b)).

![The Analysis of the Effect of Integral Gain in Rise Time Near the Set-Point](image)

**Fig. 6-5.** The effect of fuzzy inference system $f_2$ (integral gain) near the set-point.

Near the set-point in rise time interval, sufficiently large integral action is necessary to remove steady-state error. Figure 6-5 shows that low integral action causes steady-state error and an excessive overshoot occurs by large integral action. In this state, integral action is very sensitive and our proposed fuzzy inference system $f_2$ is adjusted based on two numerical sensory inputs (error and its first difference). The effectiveness of our proposed fuzzy inference system $f_2$ (derivative gain) is described in Figures 6-6. These results show that near the set-point, derivative action is more sensitive to reduce overshoot and settling time.
Fig. 6-6. The effect of fuzzy inference system $f_3$ (derivative gain) (a) far from the set-point and (b) near the set-point.

The effectiveness of proposed tuning strategy in oscillation state ($f_4$ and $f_5$) is described in Figure 6-7. The main objective of these fuzzy inference systems is to reach the steady-state as soon as possible. Therefore, fine control resolution (the small amount of gain variation) is necessary in order to keep stability. The proposed
fuzzy inference system $f_6$ is responsible to produce fine control resolution with the help of fuzzy inference system $f_6$ (Described in section 5.2). In this experiment we observed the effectiveness of various output ranges of fuzzy inference system $f_6$. The result shows that appropriate setting of the output range of fuzzy inference system $f_6$ helps to reduce overshoot and settling time without affecting the other characteristics.

The main task of our proposed fuzzy inference system $f_5$ (that is responsible to adjust integral gain when far from set-point in rise time state) is to eliminate excessive overshoot and oscillation. The effect of this fuzzy inference system is described in Figures 6-8. These results show when far from the set-point, low integral gain is necessary to reduce excessive overshoot and oscillation. But, in aggressive response case (Figure 6-8(a)), low integral gain can cause more oscillatory response. In our approach, appropriate integral action is selected online based on the current value of change in error signal.
Fig 6-9. The effect of the variation of integral action in disturbance state.

Effect of disturbance is eliminated by low frequency gain (integral gain). Figure 6-9 shows the effect of the variation of integral action in disturbance state. High integral action can eliminate fast the disturbance effect, but it leads to unstable system. The proposed fuzzy inference system $f_i$ is responsible for producing appropriate integral action in order to remove disturbance effect.

These experiments reveal that the flexibility of our proposed controller structure. Based on these experiment results, each fuzzy inference system has individual task and they are less sensitive to other performance. The corresponding task of each fuzzy inference system can be described as following:

1. $f_1$ - to speed up the process response
2. $f_2$ - to eliminate steady-state error
3. $f_3$ - to eliminate overshoot and improve settling time
4. $f_4$ - to improve settling time
5. $f_5$ - to eliminate excessive overshoot and oscillation
6. $f_6$ - to eliminate overshoot and improve settling time
7. $f_7$ - to remove disturbance effect
In our approach, the specific requirement can be achieved by adjusting output range of appropriate fuzzy inference system. It is also easy to understand and analyze the relationship between the variation of conventional PID controller parameters and a variety of process responses.

6.3.2. Transient and steady-state response characteristics test

In many practical cases, the desired performance characteristics of control systems are specified in terms of time-domain quantities. We analyze the performance characteristics (transient and steady-state response of control system) based on transient response to a unit-step input and unit-step load since it is easy to generate and is sufficiently drastic. Steady-state accuracy, rise time, maximum percent overshoot and settling time are analyzed in these experiments. We extracted and reproduced some results from other reported works in order to compare with our work.

![Comparison of set point tracking and disturbance rejection for first-order process plus time delay](image)

Fig. 6-10. The comparison of set point tracking and disturbance rejection for first-order process plus time delay (a) Ambrosio's work [Ambrosio02], (b) proposed work and TPID.
The performance of rise time and maximum percent overshoot normally conflict with each other. Ambrosio’s [Ambrosio02] work is the same as conventional PID controller with set point weighting (set point factor is 0.5). Though it can reduce the overshoot, rise time is increased. According to the results from Figures 6-10(b), 6-11(b), 6-13(a) and 6-14(b), obtained from our proposed idea, overshoot is obviously reduced without too much effect in rise time for three processes (apart from third-order process). The settling time (±2% of final value) is also improved. Disturbance rejection behavior of our controller is somewhat better than traditional PID controller in second-order process, third-order process and higher-order process. However, previous reported works only emphasize on the set point tracking as shown in Figures 6-12(a), 6-12(b), 6-12(c), 6-13(b) and 6-14(c) [Zhao93, Chen98, Guzelkaya03]. These results show that our proposed controller design can handle both disturbance rejection and set point tracking.

Fig. 6-11. The comparison of set point tracking and disturbance rejection for second-order process plus time delay (a) Ambrosio’s work [Ambrosio02] and (b) proposed work and TPID.
Fig. 6-12. The comparison of set point tracking for second-order process (a) Zhao's work (with time delay) [Zhao93], (b) Chen's work (with time delay) [Chen98], and (c) Guzelkaya’s work [Guzelkaya03] (RROM = Relative Rate Observer Method).

Fig. 6-13. The comparison of set point tracking for third-order process (a) proposed work and TPID (with disturbance rejection) and (b) Zhao's work [Zhao93].
Fig. 6-14. The comparison of set point tracking and disturbance rejection for higher-order process (a) Ambrosio's work [Ambrosio02], (b) proposed work and TPID and (c) Zhao's work [Zhao93] (without disturbance rejection).
6.3.3. Robustness test

Many systems have several parameters that are constants but uncertain within a range. To ascertain stability of the system, investigation is necessary for all possible combinations of parameters. The sensitivity of a control system to a parameter variation is of prime importance that is known as robustness of controller design.

Fig. 6-15. The comparison of responses with different value of parameter $a$ for (a) first-order process, (b) second-order process, (c) higher-order process.
A control system is robust when it has low sensitivities or it is stable over the range of parameter variations or the performance continues to meet the specifications in the presence of a set of change in the system parameters. The system should be able to withstand unexpected effects such as disturbance, unmodeled delay and unmodeled dynamics.

To analyze the robustness of our controller design we assume some parameters of system (parameter \( a \) and delay period \( \beta \)) can vary within bounded range. For characteristic equation of first-order process with known coefficient within bound is

\[
s + a = 0,
\]

(6.5)

where \(0.5 \leq a \leq 1.5\).

For characteristic equation of second-order process with known coefficients within bounds is

\[
s^2 + 2s + a = 0,
\]

(6.6)

where \(0.5 \leq a \leq 1.5\).

Known coefficients within bounds of higher-order process that is described in Equation 6.4 is \(17 \leq a \leq 37\).

Figure 6-15(a), (b) and (c) describe the robustness of our controller design and conventional P1D control system for various parameters values. The simulation results show that our controller design is well-suited for aggressive response case (i.e., \(a = 0.5\) for Figure 6-15(a), 6-15(b) and \(a = 37\) for Figure 6-15(c)) and in sluggish response case rise time performance of TPID is better than our model.

In practical control applications, the time delay introduces an additional phase lag and results in a less stable system. The robustness of unmodeled time delay is observed with the various time delay factors and results are given in Figure 6-16 (a) and (b). The proposed modular gain scheduling mechanism reduces the loop gain especially in low frequency gain (integral gain) in the time delay period of rise time and it becomes less sensitive to the effect of delay.

These results show proposed controller design can minimize the time delay effect and traditional PID controller cannot be considered adequate robust in unmodeled time delay parameter. It is also noticed that our proposed controller is less sensitive than traditional PID controller for parameters variation of the process.
Fig. 6-16. Comparison of responses with uncertain delay parameter for (a) first-order process (delay time is 0.3 sec), (b) second-order process (delay time is 0.6 sec).
6.4. Discussions

The accuracy and validity of our proposed knowledge-based modular neural network classifier and the performance of our proposed controller are analyzed with several experiments. Our fuzzy gain scheduling mechanism is a modular structure and it includes several different tuning strategies. It allows variation to both structure and parameters of standard controller (Conventional PID controller).

In adaptive control application, online observation of the process response is one of the important factors to implement adaptive law. In our approach, knowledge-based modular neural network classifier provides online estimation of process response. This information is used as a switching law to select appropriate module of modular adaptable mechanism. Consequently, the performance and stability of proposed technique is largely dependent on the accuracy of proposed modular structure classifier.

First group of experiments describe the accuracy and validity of our proposed knowledge-based modular neural network classifier. These experiment results (Figure 6-1, 6-2 and 6-3) show that proposed classifier has sufficient estimation accuracy in order for it to be applied in adaptive control applications and our proposed knowledge-based refinement algorithm can debug offline incomplete knowledge base or incompatible rules.

Second group of experiments show that the performance of our proposed controller. It has been empirically found that sufficient flexible PID controller provides excellent results in many practical applications. Several experiments (Figures 6-4, 6-5, 6-6, 6-7, 6-8 and 6-9) reveal the flexibility of proposed structures. They also show the effectiveness of each of tuning strategies (each of fuzzy inference systems) in different operating regions.

We then analyze the transient/steady-state response and the robustness of proposed controller design. First test observes the steady-state accuracy, disturbance rejection and set point tracking performance of our controller. Second test describes the robustness of our controller. It analyzes the unmodeled delay effect and parameter of uncertainty. These results show that it can handle wider operating ranges than non-modular adaptive controllers.
Though most of the adaptive PID control methods available so far concentrate on transient and steady-state response characteristics, this approach considers all of the key characteristics of control system such as stability, sensitivity, steady-state accuracy and transient response characteristic. We show that our controller design provides better performance than traditional PID controller along all the four important characteristics of the control system design.