Saeed, Bakhtiar I. and Mehrdadi, Bruce

Zero overshoot and fast transient response using a fuzzy logic controller

Original Citation


This version is available at http://eprints.hud.ac.uk/id/eprint/15017/

The University Repository is a digital collection of the research output of the University, available on Open Access. Copyright and Moral Rights for the items on this site are retained by the individual author and/or other copyright owners. Users may access full items free of charge; copies of full text items generally can be reproduced, displayed or performed and given to third parties in any format or medium for personal research or study, educational or not-for-profit purposes without prior permission or charge, provided:

- The authors, title and full bibliographic details is credited in any copy;
- A hyperlink and/or URL is included for the original metadata page; and
- The content is not changed in any way.

For more information, including our policy and submission procedure, please contact the Repository Team at: E.mailbox@hud.ac.uk.

http://eprints.hud.ac.uk/
Zero Overshoot and Fast Transient Response Using a Fuzzy Logic Controller

Bakhtiar I. Saeed & Bruce Mehrdadi
School of Computing and Engineering,
University of Huddersfield
Queensgate, Huddersfield HD1 3DH, UK
b.saeed@hud.ac.uk, b.mehrdadi@hud.ac.uk

Abstract— In some industrial process control systems it is desired not to allow an overshoot beyond the setpoint or a threshold, this could be a safety constraint or the requirement of the system. This study investigates the achievement of a zero overshoot step response using a fuzzy logic controller. A fuzzy PID controller is applied to stable, marginally stable and unstable systems and their step responses are compared with a PID controller. A comparative case study shows that the proposed fuzzy controller outperforms conventional PID controller in achieving zero overshoot response.

Keywords - fuzzy PID+I; PID controller; zero overshoot; scaling gain; tuning.

I. INTRODUCTION

An integral part in controller design and analysis is to achieve a satisfactory response in transient time and steady state. The characteristics of these states can be represented in parameters such as: overshoot, rise time, settling time and steady state error. In a stable system, the transient response exists for a short period of time. However, this might cause problems in some applications. For example, in some chemical processes, it is desired to have zero overshoot or an overshoot that does not exceed a specific threshold.

The conventional Proportional-Integral-Derivative (PID) controllers, which are the most popular feedback methods for their robustness and simplicity [1, 2] have some limitations, particularly when they are applied to obtain zero overshoot. These controllers can be tuned in several ways [3] to achieve zero overshoot if possible, but most of the time this is at the expense of rise time, and vice versa.

Some methods have been reported by researchers to find the values of PID gains to achieve zero overshoot [4-7]. By relating the step response overshoot to the positions of zeros and poles of a transfer function, a method has been derived to find the parameters of PID controller [6]. This method has been used to avoid overshoot in second order and lower order systems. A cascade sliding mode-PID controller has been proposed in literature [7].

On the other hand fuzzy logic controllers have been applied successfully in industrial processes and in some cases outperform PID controllers [8], in particular when the controlled system is complex or non-linear, as this is the case in many process control systems [9]. Nonetheless, designing fuzzy controllers is challenging. There is no systematic process for the design of fuzzy logic controllers that will produce a high performance controller for a wide range of applications [10, 11]. For example, it is difficult to find the relation between selecting membership function type or rule base, and the controller performance such as better rise time or less overshoot. In addition, unlike conventional controllers, fuzzy controllers have several parameters that can be adjusted, such as membership function shape, rules and scaling gains. Furthermore, there is no general rule of tuning these parameters. However, some techniques applied in tuning conventional controllers can still be utilised to some extent [10].

In this study, a fuzzy PID controller is adopted and is applied to different second order systems. Initially the controller gains are fixed and then manually tuned to achieve zero overshoot with a short rise time and settling time. A case study has been used to compare the performance of the fuzzy PID and conventional PID controllers for a second order system. The results show that fuzzy controllers outperform conventional controllers in achieving zero overshoot and fast transient response.

The remainder of this paper is organised as follows: Section 2 presents an overview of the fuzzy controller. Design and synthesis of the fuzzy PD+I controller is illustrated in section 3 using MATLAB and Simulink. Simulation results are shown in section 4. Finally, some conclusions are drawn in section 5.

II. STRUCTURE OF THE FUZZY CONTROLLER

The most widely used fuzzy controllers are fuzzy proportional-derivative controllers (FPD) and they act on two inputs: error and change in error (derivative of error) signals; therefore, designing rule base for these controllers is well understood and a straightforward procedure. This configuration exhibits a good performance at the transient response of the system, while encountering problems at the steady state when the error is close to zero [3, 12, 13].

To enhance the steady state performance of a system, integral action is required [3]. Thus the controller becomes fuzzy proportional-integral controller (FPI). Although these controllers have good performance, at the steady state they suffer from a slow response [12, 13].

Improving both the transient state and the steady state requires a controller that includes both derivative and integral actions. A fuzzy controller with this capability is
known as fuzzy proportional-derivative-integral controller (FPID).

In literature, various structures have been proposed to design FPID controllers [3, 13-15].

Fig. 1 shows a simple design proposed in [3] and was adopted as the fuzzy PID controller in this study.

The controller consists of a normal FPD with added integral action; therefore it is known as FPD+I. As the controller has three inputs: error, derivative of error and integral of error, it can provide all the benefits of conventional PID controllers, but still has some disadvantages such as derivative kick and integrator windup. Additionally, there is only one rule base with two inputs; therefore, designing the rule base is less complex than the structure proposed in [15] which has three input rule base. Furthermore, some techniques applied in tuning conventional PID controllers can still be utilised to some extent [3, 10].

### III. Simulation of Controllers

#### A. Fuzzy PD+I

Matlab (v7.9) and Simulink were used to build and simulate the model. Fig. 2 shows the FPD+I controller in a closed-loop feedback system.

The plant block represents the desired transfer function to be controlled.

The fuzzy PD+I controller was formed by adding the integrator to the output of the fuzzy PD controller. The controller has three input signals: error (e), change in error (ce) and integral of error (ie). The error signal is obtained from the difference between the setpoint (r) and the measured plant output (y), the change in error signal and the integral of error signals are produced by passing the signals through derivative and integral blocks respectively. The inputs have scaling gains: gain of error (GE), gain of change in error (GE) and gain of integral error (GIE). These gains along with the output gain (GU) can be tuned to achieve better performance [3, 10, 15-17]. Adjusting these gains is used frequently for tuning fuzzy controllers and it has been regarded as an effective approach [12, 15, 16, 18-21]. First of all, they have a global effect on the performance of the controller; their effects can be easily observed [18]. Secondly, there are few parameters to tune, thus the tuning process is computationally efficient in contrast with other methods, where there are several parameters to tune. Finally, they can be considered as conventional controller gain parameters; therefore, they are convenient to tune and some ideas from conventional controller tuning can be borrowed [12, 16, 18-21].

The controller output (u) is formed by adding the integral of error (ie) to the output of the fuzzy PD controller (cu).

To represent the values of the e, ce and cu, five symmetric triangle shape membership functions (except trapezoid for the two at the extreme ends for e and ce) with 50% of overlap were chosen [3, 10]. Although the choice of membership function shape and width is subjective, triangular shapes were chosen, because they are more popular and convenient [10, 11]. The interval of [-1, 1] was used for the universes of discourse of the input variables, while [-2, 2] was used for the output variable. The linguistic descriptions of the input and output membership functions are negative large (NL), negative small (NS), zero (ZE), positive small (PS) and positive large (PL). These are shown in Fig. 3 and Fig. 4 respectively.

The minimum operator was selected as an implication method, and the most popular and standard method of defuzzification process known as the centre of gratify (CoG) was selected.

The fuzzy rule-base is a mapping between the inputs, e and ce and the output, cu. A sample of the rule has the following form:
If error is PL and change in error is PL, then control signal is PL.

The rule implies that if the error is positive large (measured output far away from the set point) and the change of error is positive large, then the control signal should be positive large to return back the output near the setpoint.

As there are 5 linguistic variables for each input, 25 rules were created using Fuzzy Logic Toolbox, Table I shows the rules.

<table>
<thead>
<tr>
<th>Error</th>
<th>NL</th>
<th>NS</th>
<th>ZE</th>
<th>PS</th>
<th>PL</th>
</tr>
</thead>
<tbody>
<tr>
<td>NL</td>
<td>NL</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
<td>ZE</td>
</tr>
<tr>
<td>NS</td>
<td>NL</td>
<td>NS</td>
<td>NS</td>
<td>ZE</td>
<td>PS</td>
</tr>
<tr>
<td>ZE</td>
<td>NS</td>
<td>NS</td>
<td>ZE</td>
<td>PS</td>
<td>PS</td>
</tr>
<tr>
<td>PS</td>
<td>NS</td>
<td>ZE</td>
<td>PS</td>
<td>PS</td>
<td>PL</td>
</tr>
<tr>
<td>PL</td>
<td>ZE</td>
<td>PS</td>
<td>PS</td>
<td>PL</td>
<td>PL</td>
</tr>
</tbody>
</table>

Three standard closed-loop performance criteria were chosen as design specifications to measure the performance of the controller [2]: maximum percentage overshoot ($M_p$), rise time ($t_r$) and settling time ($t_s$).

Finally, a script code was developed to simulate the model, calculate the $M_p$, the $t_r$ and the $t_s$ and generate the required plots.

### B. PID controller

In order to compare the performance of the FPD+I controller with a conventional PID controller, a simulation model of a PID controller with auto tuning capability was created, this is shown in Fig. 5.

![Simulink model of conventional PID controller in a closed-loop control structure.](image)

The model contains a plant block that represents the desired transfer function to be controlled and a PID controller block with auto tuning capability. Also for this design, the same performance measures were chosen and a script code was developed to simulate the model.

### IV. RESULTS AND DISCUSSIONS

#### A. Tests

In order to evaluate the design, three different systems: stable, marginally stable and unstable were simulated using the FPD+I and the conventional PID models. The transfer functions of these systems are provided in Table II.

<table>
<thead>
<tr>
<th>System</th>
<th>Transfer Function</th>
<th>Stability</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$\frac{1}{5^1 + 0.5^1 + 1}$</td>
<td>Stable</td>
</tr>
<tr>
<td>2</td>
<td>$\frac{1}{5^1 + 1}$</td>
<td>Marginally stable</td>
</tr>
<tr>
<td>3</td>
<td>$\frac{1}{5^1 - 5^1 + 1}$</td>
<td>Unstable</td>
</tr>
</tbody>
</table>

Initially, the gains of the FPD+I controller GE, GCE, GIE and GU are set to 1, and then tuned to achieve zero overshoot with a fast rise time and short settling time, the tuned values are shown in Table III.

<table>
<thead>
<tr>
<th>System</th>
<th>GE</th>
<th>GCE</th>
<th>GIE</th>
<th>GU</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0.6</td>
<td>0.25</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0.4</td>
<td>0.15</td>
<td>12</td>
</tr>
<tr>
<td>3</td>
<td>0.8</td>
<td>0.025</td>
<td>0.01</td>
<td>2800</td>
</tr>
</tbody>
</table>

For the conventional PID controller, the Matlab PID auto tuner was used to obtain the values of PID gains (P, I and D), these values are shown in Table IV.

<table>
<thead>
<tr>
<th>System</th>
<th>P</th>
<th>I</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>11.74</td>
<td>0.85</td>
<td>8.85</td>
</tr>
<tr>
<td>2</td>
<td>12.04</td>
<td>1.09</td>
<td>12.088</td>
</tr>
<tr>
<td>3</td>
<td>3273.19</td>
<td>6345.72</td>
<td>325.43</td>
</tr>
</tbody>
</table>

The step responses of the three systems for the FPD+I and the conventional PID are combined together and shown in the Fig. 5 - Fig. 7.
The results obtained from the fuzzy PD+I clearly indicate substantial improvements in transient response of systems have been achieved.

**B. Case study**

The transfer function of a chemical process shown in (1) has been used by other researchers [6] to achieve zero overshoot in the closed-loop response. Accordingly, the parameters of the PID controller were calculated (P=7.2, I=0.972 and D=6.99).

\[
G(s) = \frac{1}{(1 s^2 + 2 s + 5)}
\]  
(1)

Three tests were conducted on the above system: the FPD+I controller with tuned gains, a conventional PID controller using the parameters proposed by the method in [6] and the conventional PID controller using Matlab auto tuner. The parameters of the three controllers were as follows: FPD+I (GE = 1, GCE = 0.7, GIE = 0.04, GU = 20), PID controller using the method in [6] (P = 7.2, I = 0.72, D = 6.99) (The original values were for KC, Ti and Td, they were converted to the values of P, I and D to be used within the setting of Matlab PID controller) and for the conventional PID controller using Matlab auto tuner (P = 1.74, I = 0.20, D = -4.58).

The closed-loop step responses of the three controllers along with the open-loop step response are shown in Fig. 9.

The performance measures of the controller are shown in Table V.

<table>
<thead>
<tr>
<th>System</th>
<th>Performance measure</th>
<th>Open-loop</th>
<th>Conventional PID</th>
<th>Fuzzy PD+I</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$M_p$</td>
<td>% 44.43</td>
<td>% 0</td>
<td>% 0</td>
</tr>
<tr>
<td></td>
<td>$t_r$</td>
<td>1.25</td>
<td>1.63</td>
<td>1.10</td>
</tr>
<tr>
<td></td>
<td>$t_s$</td>
<td>14.11</td>
<td>5.92</td>
<td>1.97</td>
</tr>
<tr>
<td>2</td>
<td>$M_p$</td>
<td>% 100.0</td>
<td>% 0.43</td>
<td>% 0.0</td>
</tr>
<tr>
<td></td>
<td>$t_r$</td>
<td>1.01</td>
<td>1.97</td>
<td>0.67</td>
</tr>
<tr>
<td></td>
<td>$t_s$</td>
<td>Not known</td>
<td>3.42</td>
<td>1.16</td>
</tr>
<tr>
<td>3</td>
<td>$M_p$</td>
<td>% 1.28 e+19</td>
<td>11.12</td>
<td>% 0.0</td>
</tr>
<tr>
<td></td>
<td>$t_r$</td>
<td>0.13</td>
<td>0.14</td>
<td>0.041</td>
</tr>
<tr>
<td></td>
<td>$t_s$</td>
<td>Not known</td>
<td>1.22</td>
<td>0.06</td>
</tr>
</tbody>
</table>
**C. Discussions**

It is evident from the results, that the FPD+I controller has achieved zero overshoot with faster rise time and shorter settling time compared to the conventional PID controller. Although the conventional PID controller has achieved zero overshoot for the first and second systems, it has been at the expense of the rise time and the settling time. Additionally, in the third system and in the case study the performance of the conventional PID controller was degraded as the response resulted with some overshoot.

**V. CONCLUSIONS**

A fuzzy PID controller was applied to a stable, marginally stable and unstable second order system. The results showed that fuzzy PID controller outperformed the conventional PID controllers in achieving zero overshoot and produced a faster transient response. This is an ongoing research and the next phase of the work will encompass the ability to fine tune the fuzzy gains automatically. Inclusion of predictive and intelligent agents in the fuzzy algorithms will reduce the tuning time.

**REFERENCES**


