Fuzzy Adaptive PID Controller Applied to an Electric Heater in MATLAB/Simulink

Guillermo Valencia Ochoa¹, Jorge Duarte Forero² and Luis Obregón Quiñones³

¹,² Efficient Energy Management Research Group, Universidad del Atlántico
Carrera 30 No 8 – 49, Puerto Colombia, Colombia

³ Research Group on Sustainable Chemical and Biochemical Processes
Universidad del Atlántico, Carrera 30 No 8 – 49, Puerto Colombia, Colombia

Copyright © 2018 Guillermo Valencia Ochoa et al. This article is distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract

This document shows the methodological development of a fuzzy adaptive PID controller in simulation over a furnace, which is modeled with a first order transfer function plus dead time. Two types of membership functions are compared in this article, for self-tuning fuzzy controller driver. The performance of each controller was determined by calculating the integral of the absolute error value, in which a slight advantage for the Gaussian function was obtained with respect to the triangular function.

Keywords: Adaptive Control, Fuzzy Control, control system tuning PID

1. Introduction

Within the framework of the subject of fuzzy control, it has been decided to develop a comparative performance analysis in simulation of two membership functions for the autotuning of an adaptive PID controller [1]. Therefore, you have selected a heating system due to the significant demand for energies that these present, so that to a large extent to minimize this energy impact is by improving the closed-loop system response[2].
Temperature control systems are of great interest, because they have transport delays and thermal inertias that make them non-linearities and in one way or another make the implementation of controllers attractive intelligent [3], [4]. The object of this article is to propose a temperature control law of an electric heater system by means of a fuzzy self-adaptive PID controller implemented in Matlab that can be used as tutorial and theoretical guide in an advanced control class[5], in addition to comparing these results with the results obtained with a triangular membership function [6], [7]. The result obtained is the comparison between a classic PID and a PID autotuned by fuzzy logic, using the function of both Gaussian and triangular membership in both the variables anticipated as consequential.

2. Methodology

2.1 Process Modeling
The process is an electric heater or an evaporator which is either an electric heater or an evaporator modeling from equation (1):

\[
G(s) = \frac{K}{1+Ts} e^{-ts}
\]  

(1)

where the values of K, T and the transport delay are 1, 11 and 1.8 respectively (Xin, Liu, Huang, & Tang, 2009). By substituting the values in equation (1), equation (2) is obtained.

\[
G(s) = \frac{1}{1+11s} e^{-1.8s}
\]  

(2)

2.2 Fuzzy adaptive controller
This diffuse adaptive controller is made up of a set of fuzzy rules that attach to the classic PID controller as shown in Figure 1.

![Figure 1. Control loop implementing a diffuse adaptive PID.](image)

The first block estimates the values of the controller gains from the error and error change, which are sent to the second block, PID controller (Block 2), where the operations shown in equations (3), (4) and (5) are performed to obtain the new value of Kp, Ki and Kd.
where $K_p, K_i, K_d$ are the output values after tuning and $K_{p^0}, K_{i^0}, K_{d^0}$ are the starting values of the PID controller and $K_{p,fuzzy}, K_{i,fuzzy}, K_{d,fuzzy}$ are the output values of the first block.

The block system in Simulink shown in Figure 2 presents the output of the fuzzy controller compared to that of a PID. The values of $K_p$, $K_i$ and $K_d$ are 1.2, 0.23 and 0.3 respectively for the classic PID.

**Figure 2.** Block diagram of the system.

### 2.3 Fuzzy adaptive block

To install the fuzzy logic controller block in Simulink, first configure the fuzzy logic controller as a variable in the Matlab Workspace window.

Initially you start by entering in the command window, `>> fuzzy`, where a window called FIS Editor: Untitled will appear.

In the edit button you have added the two inputs (e and de) and the three output variables ($K_p, K_i, K_d$) by clicking Add Variable. By default, the variables appear with the name of input1, input2, output1, output2 and output3. In making double-click in the boxes opens the function editor of the membership, where for each variable you entered the respective membership functions.

For input error, a range of $[-6 6]$ was first introduced. Then, in the edit button you removed all the membership functions that come out by default and later on in the first and the second as automatically thrown by the ordination. Each function of membership was assigned names of NB, NM, NS, ZO, PS, PM and PB as shown in Figure 3 for both membership functions.
For the input from. The same operation was performed as in variable e, whose membership functions are displayed in the Figure 4 for the output variable Kp. It worked with a range of [-0.5 0.5] and in parameters it was assigned to all the functions 0.075 in the first parameter and in the second according to the automatic sorting as shown in Figure 3. For the output variable Ki. Worked with a range of [-0.5 0.3]. For the NB membership function, values were assigned as follows says figure 3. For functions from NM to PS, the first parameter with a value of 0.11 and the second with a value of 0.11 was used. According to the automatic order. The PM function was left both parameters according to the automatic sorting and PB was changed to a gauss2mf function, the functions are shown in Figure 3.

![Figure 3](image1.png) ![Figure 3](image2.png)

**Figure 3.** Member ship functions for: a) variable Kp and b) Ki.

The same procedure was performed as the other variables, but the range will be between [-1 -2]. The first parameter was assigned a value of 0.2124 for all functions and the second according to the automatic sorting.
Fuzzy adaptive PID controller applied to ...

Immediately, in the edit button, the Rules option, heuristics were added according to Table 1. In addition, in the Surface option inside the view tool, you can see the surface graphs of the input variables with respect to the output variables.

![Image](image1.png)

**Figure 4.** Control surface for the variable Kp.

In the sale of Rule Editor: Untitled in the bottom left corner of the screen there is an area called connection, where rules 15, 24, 25, 26, 27, 30, 31, 33, 34, 35, 36, 37, 38, 40, 41 and 49 were worked with the or option. Finally, in the File button, the fuzzy logic with the name of the final_driver was exported to the WORKSPACE window.

![Image](image2.png)

**Figure 5.** Control surface for the variable Kd.
Once you have the "final_controller" variable in the workspace window, open the fuzzy logic controller block in the fuzzy block subsystem of the main block diagram, and type the variable name: final_controller. In this way, the program will be able to simulate.

\[ u(t)_{PID} = K_p * e(t) + K_i * \int_0^t e(t) dt + K_d * \frac{d(e(t))}{dt} \] (6)

where the constant values are calculated from equations (3), (4) and (5). From equation (6), we make the block diagram for the PID as shown in Figure 12.
3. Results

Figure 9 shows the system response for a simulation time of 70. The IAE for the Gaussian surface was 5.3443 for the fuzzy controller, while for the triangular function 5.355, so this surface presented a better performance for the test performed. In both cases the results of the classic PID presented a higher EAR than the diffuse adaptive one, with a value of 6.953, which shows the benefits of this control law in the presence of non-linearities.

Figure 9. Response of the classic PID and an the adaptive fuzzy PID controller.
4. Conclusions

The Gaussian membership function presented better results in terms of IAE than the triangular function, so this type of function is recommended for the operation of the PID controller autotuning rules. For thermal systems that have significant downtime or transport delays, the fuzzy adaptive controller performs better than the classic PID controller when compared in terms of IAE.

References


Received: July 15, 2018; Published: August 1, 2018