

Article

Equivalence between Fuzzy PID Controllers and Conventional PID Controllers

Chun-Tang Chao, Nana Sutarna, Juing-Shian Chiou * and Chi-Jo Wang

Department of Electrical Engineering, Southern Taiwan University of Science and Technology, 1, Nan-Tai St., Yongkang District, Tainan City 71005, Taiwan; tang@stust.edu.tw (C.-T.C.); da320208@stust.edu.tw (N.S.); chijo@stust.edu.tw (C.-J.W.)

* Correspondence: jschiou@stust.edu.tw; Tel.: +886-916-221-152; Fax: +886-6-3010-069

Academic Editor: Hung-Yu Wang

Received: 25 March 2017; Accepted: 6 May 2017; Published: 2 June 2017

Abstract: This paper proposes the equivalence between fuzzy Proportional-Integral-Derivative (PID) controllers and conventional PID controllers. A well-designed conventional PID controller, with the help of the proposed method, can be rapidly transformed to an equivalent fuzzy logic controller (FLC) by observing and defining the operating ranges of the input/output of the controller. Furthermore, the knowledge base of the proposed equivalent fuzzy PID controller is represented as a cube fuzzy associative memory (FAM), instead of a combination of PD-type and PI-type FLCs in most research. Simulation results show the feasibility of the proposed technique, both in continuous and discrete time. Since the design techniques of conventional linear PID controllers have matured, they can act as preliminary expert knowledge for nonlinear FLCs designs. Based on the proposed equivalence relationship, the designer can further tune the membership functions of fuzzy variables in the control rules to exhibit the nonlinearity of a FLC and yield more satisfactory system responses in an efficient way.

Keywords: equivalence; conventional PID controller; fuzzy logic PID controller

1. Introduction

Proportional-Integral-Derivative (PID) controllers are widely used in industrial process control. The three-mode controller contains a proportional, an integral, and a derivative term. The popularity of a PID controller can be attributed to its good performance and functional simplicity, which allows engineers to operate it in a simple and straightforward manner. For example, the three controller gains can be chosen independently by an engineer, based on one's experience or through some simple selection methods such as the classical tuning rules proposed by Ziegler-Nichols [1]. For simplicity of the controller design, a PI or PD controller are also popular for practical applications. A PI controller can add damping to a system and reduce steady-state error, but yields penalized rise time and settling time. A PD controller also adds damping and reliably predicts and reduces large overshoots, but does not improve the steady-state error. Thus, for complete design considerations, a PID controller should be employed to obtain a desirable system response in settling time, steady-state error, and overshoot.

On the other hand, since Lotfi Zadeh rediscovered and promoted fuzziness in 1965, the subsequent two fuzzy inference techniques proposed by Mamdani [2] and Sugeno [3] have inspired research in fuzzy logic controllers (FLC). The heuristic fuzzy rules, which reflect the experience of human experts, can be applied to plants that are difficult to model mathematically. The most common FLCs are PI-type or PD-type controllers [4–6], which possess the same characteristics as traditional PI or PD controllers, respectively. Moreover, they exhibit superior applicability compared with traditional PI or PD controllers [7].

The FLC commonly outperforms the corresponding PI, PD, or PID controller because a FLC is a nonlinear controller, while a PI, PD, or PID controller is linear. This raises the equivalence problem between a fuzzy PID controller and a conventional PID controller. It is well known that the design of fuzzy rules for a FLC requires expert knowledge, and those who are skilled in conventional PID controller design are then qualified as the experts. For a well-tuned conventional PID controller, design work is saved by replacing it with an equivalent linear FLC, then improving the performance over a conventional PID controller by slightly modifying the fuzzy rules.

Moon [8] revealed that when a PI controller is given, an FLC output is identical to that of the PI controller by using specified fuzzy logic operations. However, Moon’s design is limited to PI controllers, and the design procedure is not clear enough. Several studies have investigated fuzzy PID controller structures, by taking different combinations of the fuzzy PID structural elements [9–11]. This involves a large number of parameters in defining the fuzzy rule base. Manikandan et al. [12] presented a design for an equivalent fuzzy PID controller from the conventional PID controller, but the tuning procedure was too complicated and the resulting FLC was not purely linear according to the control surface view of the study. Therefore, the equivalence problem between different systems is very crucial to many research fields [13].

The objective of the study is to extend the significant results derived by Moon [8] by examining the equivalence relationship and design procedure between a traditional PID controller and its corresponding equivalent FLC. This research proposes an equivalent fuzzy PID controller which has a simple PID structure design with a 3-dimensional fuzzy rule table, instead of the combination of different fuzzy PID structural elements or a hybrid controller structure [14]. Moreover, to achieve optimal control performance for a FLC, some artificial intelligent techniques such as Genetic Algorithm and Neural Network are efficient approaches [15,16]. This inspires us in the future to propose nonlinear factors for tuning the membership functions to develop an optimal fuzzy PID controller design with less parameters.

This study presents the equivalent fuzzy PID controller design (Section 2), followed by the simulation results of Matlab/Simulink for verifying the proposed design (Section 3). Finally, the concluding remarks and implementation issues (Section 4) are discussed.

2. The Equivalent Fuzzy PID Controller Design

The fuzzy PID controller design proposed is equivalent to a conventional PID controller, and is derived from the equivalence equations. First, for a conventional PID controller, the equation for output $u(t)$ in the time domain is

$$u(t) = K_P e(t) + K_I \int e(t) dt + K_D \frac{de(t)}{dt}, \tag{1}$$

where the controller provides a proportional term, an integration term, and a derivative term. The output $u(t)$ and the three inputs $e(t)$, $\int e(t)$, and $\dot{e}(t)$ can be thought as fuzzy variables in the FLC design. It is assumed that the operating ranges for $u(t)$, $e(t)$, $\int e(t)$, and $\dot{e}(t)$ are $OR_u = [-a_u, a_u]$, $OR_e = [-a_e, a_e]$, $OR_i = [-a_i, a_i]$, and $OR_d = [-a_d, a_d]$, respectively. Figure 1 shows the membership functions for graphically defining the four fuzzy variables.

As shown in Figure 1, the m fuzzy sets are equally-spaced and triangular-shaped for each input fuzzy variable $e(t)$, $\int e(t)$, or $\dot{e}(t)$. On the other hand, the output fuzzy variable $u(t)$ is fuzzified by $3m - 2$ singleton membership functions. Let e_k , i_k , d_k , and u_k denote the center of fuzzy sets E_k , I_k , D_k , and U_k , respectively, so that we obtain the following equations

$$e_k = \frac{(2k - m - 1)}{m - 1} a_e, \quad i_k = \frac{(2k - m - 1)}{m - 1} a_i, \quad d_k = \frac{(2k - m - 1)}{m - 1} a_d, \quad 1 \leq k \leq m \tag{2}$$

$$u_k = \frac{(2k - 3m + 1)}{3(m - 1)} a_u, \quad 1 \leq k \leq (3m - 2). \tag{3}$$

We further define the distance between u_k and u_{k+1} as

$$\Delta u = u_{k+1} - u_k, \tag{4}$$

which will be used in later equation simplification. Based on the above fuzzy variables definition, the expression of antecedent (IF) and consequent (THEN) for each fuzzy rule is defined as

$$\text{IF } e(t) \text{ is } E_i \text{ and } \int e(t) \text{ is } I_j \text{ and } \dot{e}(t) \text{ is } D_k \text{ THEN } u(t) \text{ is } U_l, \tag{5}$$

where three input fuzzy variables $e(t)$, $\int e(t)$, and $\dot{e}(t)$ are taken into consideration simultaneously. In the proposed linear fuzzy PID design, the overall fuzzy rules for the three-by-one system can be represented by the sliced cube fuzzy associative memory (FAM), as shown in Figure 2. Furthermore, we have the following equation related to Equation (5)

$$l = i + j + k - 2. \tag{6}$$

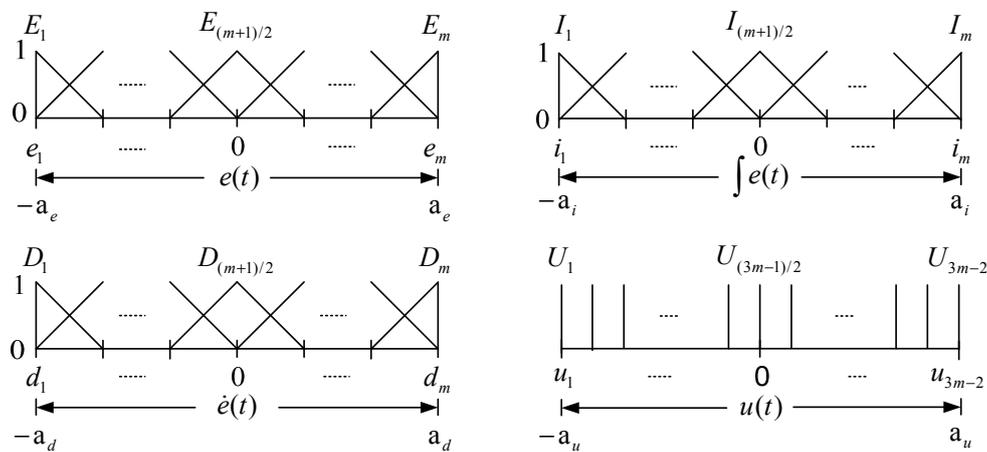


Figure 1. Graphical definition of membership functions for fuzzy variables, $e(t)$, $\int e(t)$, $\dot{e}(t)$, and $u(t)$.

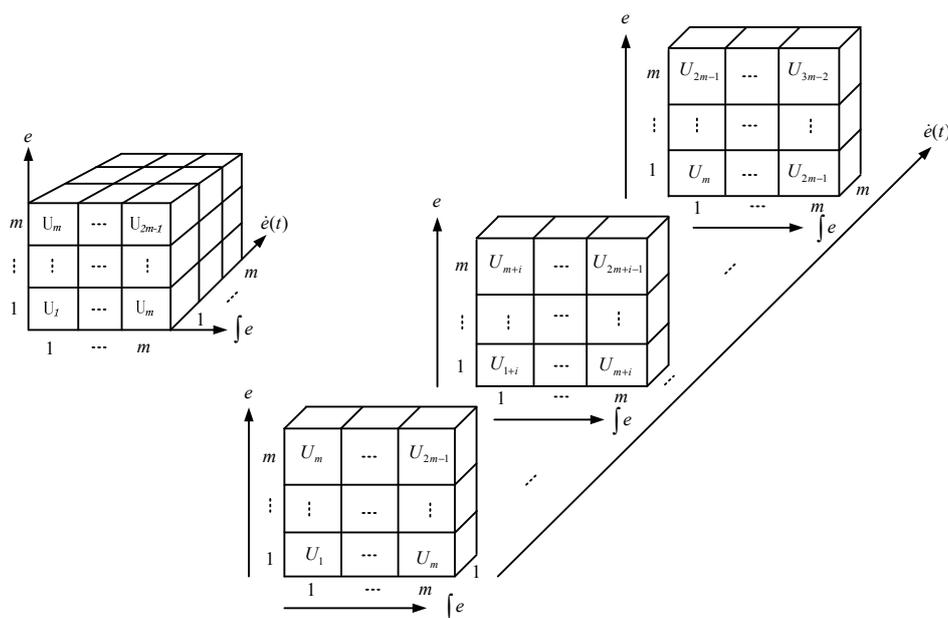


Figure 2. Sliced cube fuzzy associative memory (FAM) representation of knowledge base.

Subsequently, verification of the proposed design is done by applying the Sugeno-style inference, the resulting controller output $u(t)$ for controller inputs $e(t)$, $\int e(t)$, and $\dot{e}(t)$ can be calculated by carrying out an aggregation of the form

$$u(t) = \frac{\sum u_l \times (\mu_{E_i}(e(t)) \times \mu_{I_j}(\int e(t)) \times \mu_{D_k}(\dot{e}(t)))}{\sum (\mu_{E_i}(e(t)) \times \mu_{I_j}(\int e(t)) \times \mu_{D_k}(\dot{e}(t)))}, 1 \leq i, j, k \leq m, l = i + j + k - 2, \quad (7)$$

where the product operation rule is used for the fuzzy logic implications and the center of gravity (COG) is applied for the defuzzification process. It is determined that there is at most, eight rules to be fired for any controller inputs $e(t)$, $\int e(t)$, and $\dot{e}(t)$. To clarify, consider the crisp input $e(t)$ corresponding to the membership functions E_i and E_{i+1} to the degrees of p and $1 - p$, respectively. Similarly, consider that $\int e(t)$ maps the membership functions I_j and I_{j+1} to the degrees of q and $1 - q$, respectively. Also, it is assumed that $\dot{e}(t)$ has degrees of r and $1 - r$ with respect to the membership functions D_k and D_{k+1} . Based on the above assumption, the membership degrees p , q , and r can be described as

$$p = \frac{e_{i+1} - e(t)}{e_{i+1} - e_i}, \quad q = \frac{i_{j+1} - \int e(t)}{i_{j+1} - i_j}, \quad \text{and} \quad r = \frac{d_{k+1} - \dot{e}(t)}{d_{k+1} - d_k} \quad (8)$$

The fired eight rules are listed below, and Figure 3 is an illustration for these eight rules.

1. IF $e(t)$ is E_i and $\int e(t)$ is I_j and $\dot{e}(t)$ is D_k THEN $u(t)$ is $U_{i+j+k-2}$
2. IF $e(t)$ is E_i and $\int e(t)$ is I_j and $\dot{e}(t)$ is D_{k+1} THEN $u(t)$ is $U_{i+j+k-1}$
3. IF $e(t)$ is E_i and $\int e(t)$ is I_{j+1} and $\dot{e}(t)$ is D_k THEN $u(t)$ is $U_{i+j+k-1}$
4. IF $e(t)$ is E_i and $\int e(t)$ is I_{j+1} and $\dot{e}(t)$ is D_{k+1} THEN $u(t)$ is U_{i+j+k}
5. IF $e(t)$ is E_{i+1} and $\int e(t)$ is I_j and $\dot{e}(t)$ is D_k THEN $u(t)$ is $U_{i+j+k-1}$
6. IF $e(t)$ is E_{i+1} and $\int e(t)$ is I_j and $\dot{e}(t)$ is D_{k+1} THEN $u(t)$ is U_{i+j+k}
7. IF $e(t)$ is E_{i+1} and $\int e(t)$ is I_{j+1} and $\dot{e}(t)$ is D_k THEN $u(t)$ is U_{i+j+k}
8. IF $e(t)$ is E_{i+1} and $\int e(t)$ is I_{j+1} and $\dot{e}(t)$ is D_{k+1} THEN $u(t)$ is $U_{i+j+k+1}$

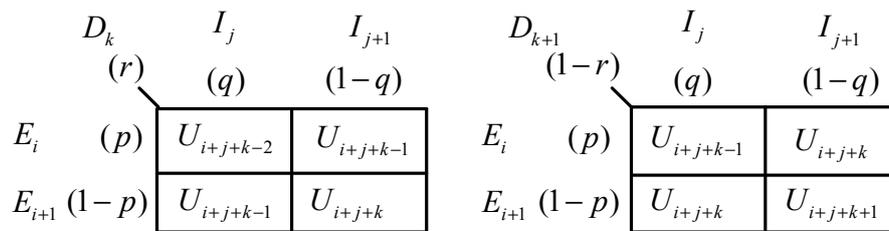


Figure 3. The fired eight rules.

As shown in Equation (7), the crisp output $u(t)$ can be evaluated as $u(t) = \frac{\text{num } u(t)}{\text{den } u(t)}$ by taking the weighted average of the eight rules consequents. Thus, the denominator of $u(t)$ with 8 terms will be finally reduced to 1, as shown in Equation (9).

$$\begin{aligned} \text{den } u(t) &= pqr + pq(1-r) + p(1-q)r + p(1-q)(1-r) + (1-p)qr \\ &\quad + (1-p)q(1-r) + (1-p)(1-q)r + (1-p)(1-q)(1-r) \\ &= 1 \end{aligned} \quad (9)$$

On the other hand, the nominator part of $u(t)$ with 8 terms is obtained by

$$\begin{aligned} \text{num } u(t) &= pqr u_{i+j+k-2} + pq(1-r) u_{i+j+k-1} + p(1-q)r u_{i+j+k-1} + p(1-q)(1-r) u_{i+j+k} \\ &\quad + (1-p)qru_{i+j+k-1} + (1-p)q(1-r) u_{i+j+k} + (1-p)(1-q)r u_{i+j+k} \\ &\quad + (1-p)(1-q)(1-r) u_{i+j+k+1} \end{aligned}, \quad (10)$$

which can also be confirmed in Figure 3. In order to simplify Equation (10), we use a method of applying Equation (4) and defining $u_x = u_{i+j+k}$. Then Equation (10) can be reduced to

$$\begin{aligned} \text{num } u(t) &= u_x + \Delta u(1 - p - q - r) = u_{i+j+k+1-p-q-r} \\ &= \frac{(2(i+j+k+1-p-q-r)-3m+1)}{3(m-1)} a_u \end{aligned} \tag{11}$$

where Equation (3) is also applied.

By substituting Equation (2) into Equation (8), we can rewrite the membership degrees p , q , and r as

$$\begin{aligned} p &= \frac{(2i-m+1)a_e - (m-1)e(t)}{2a_e}, \quad q = \frac{(2j-m+1)a_i - (m-1)\int e(t)}{2a_i}, \\ \text{and } r &= \frac{(2k-m+1)a_d - (m-1)\dot{e}(t)}{2a_d}. \end{aligned} \tag{12}$$

With Equation (12) substituted into Equation (11), $\text{num } u(t)$ is finally obtained as follows

$$\begin{aligned} \text{num } u(t) &= \frac{a_u}{3(m-1)} [2i + 2j + 2k - 3m + 3 + \frac{(m-1)e(t)}{a_e} - \frac{2ia_e}{a_e} + \frac{(m-1)a_e}{a_e} + \frac{(m-1)\int e(t)}{a_i} \\ &\quad - \frac{2ja_i}{a_i} + \frac{(m-1)a_i}{a_i} + \frac{(m-1)\dot{e}(t)}{a_d} - \frac{2ka_d}{a_d} + \frac{(m-1)a_d}{a_d}] \\ &= \frac{a_u}{3a_e} e(t) + \frac{a_u}{3a_i} \int e(t) + \frac{a_u}{3a_d} \dot{e}(t). \end{aligned} \tag{13}$$

Thus, the crisp output $u(t)$ of the proposed linear FLC is given by

$$u(t) = \frac{a_u}{3a_e} e(t) + \frac{a_u}{3a_i} \int e(t)dt + \frac{a_u}{3a_d} \frac{de(t)}{dt}, \tag{14}$$

which implies a linear PID controller with

$$K_P = \frac{a_u}{3a_e}, \quad K_I = \frac{a_u}{3a_i}, \quad \text{and} \quad K_D = \frac{a_u}{3a_d}. \tag{15}$$

Equation (15) shows that if a FLC design is based on the fuzzy knowledge from Figure 2 and the defuzzification process in Equation (7), then it will yield a linear PID controller and the resulting PID parameters have no relation with m , the number of membership functions, but is strongly correlated to the operating ranges of the control input/output. With the derived important equivalence result, the designer can obtain a FLC design prototype based on a conventional PID controller design.

In practical application, a FLC will be finished by digital implementation. When considering a digital PID controllers, the equation for the output $u[n]$ at each sampling time will be

$$u[n] = K_P e[n] + K_I T_s \sum e[n] + K_D \frac{e[n] - e[n-1]}{T_s}, \tag{16}$$

where T_s is the sampling time. In the subsequent section, the performance of the FLC implemented in digital form is verified.

3. Simulation Results

In this section, the proposed equivalence relationship is verified by use of Matlab/Simulink. A three-order controlled plant is employed with transfer function [17], which is shown below

$$P(s) = \frac{1.2}{0.36s^3 + 1.86s^2 + 2.5s + 1}. \tag{17}$$

A conventional PID controller design for $P(s)$ with $K_P = 1.2$, $K_I = 0.36$, and $K_D = 1$, which was simulated by Simulink is shown in Figure 4, and the PID controller can simultaneously improve system responses in rise time, settling time, steady-state error, and overshoot. The magnitude of the

input step signal is set as 5, and the resulting error signal, error integral, error derivative, control signal, and system output are shown in Figure 5 (red line).

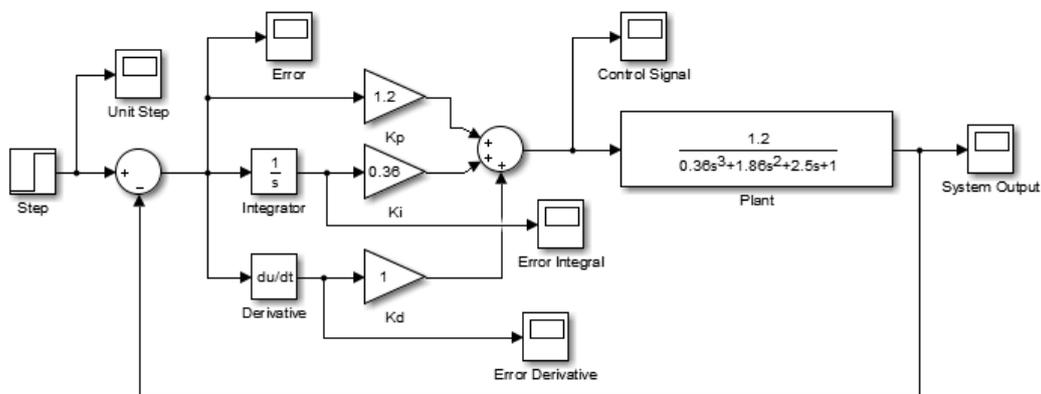


Figure 4. The Proportional-Integral-Derivative (PID)-controlled system in Simulink.

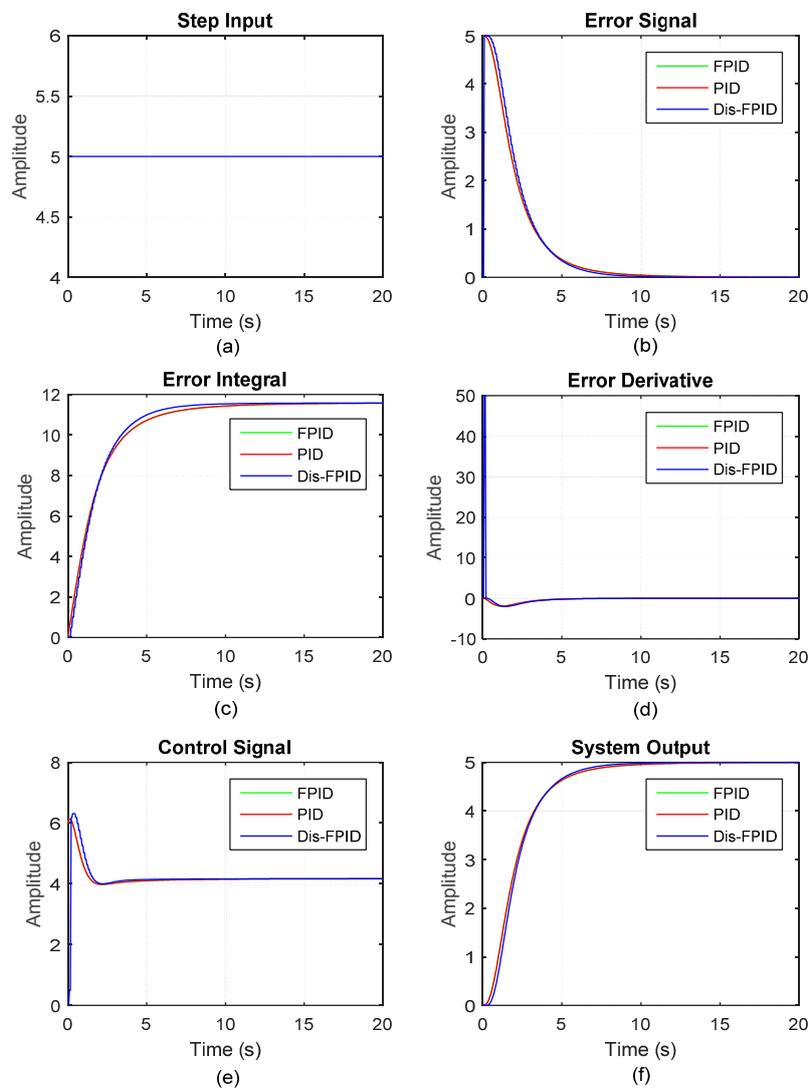


Figure 5. The (a) step input; (b) error signal; (c) error integral; (d) error derivative; (e) control signal and (f) system output with PID controller, the equivalent fuzzy logic controller (FLC), and the equivalent FLC in discrete form.

Subsequently, we built the equivalent FLC based on the above conventional PID controller design according to the derived equivalence equation. By observing the system responses $e(t)$, $\int e(t)$, $\dot{e}(t)$, and $u(t)$ with the above conventional PID controller in Figure 5 (red line), the operating ranges OR_e , OR_i , OR_d , and OR_u can be defined in accordance with Equation (15), as the procedure below shows.

1. OR_e is set as $[-a_e, a_e] = [-5, 5]$, which is the range for $e(t)$.
2. OR_u is set as $[-a_u, a_u] = [-18, 18]$ to satisfy $K_P = 1.2$.
3. OR_i is set as $[-a_i, a_i] = [-16.67, 16.67]$ to satisfy $K_I = 0.36$.
4. OR_d is set as $[-a_d, a_d] = [-6, 6]$ to satisfy $K_D = 1$.

For this case, it shows good results in defining the operating ranges. Furthermore, this is not a limitation as there are four parameters a_e , a_u , a_i , and a_d for adjustment to satisfy the three control parameters K_P , K_I , and K_D in Equation (15). The Fuzzy Logic Designer in Matlab/Simulink was applied for the equivalent FLC design and simulation. Figure 6 shows the feedback control structure.

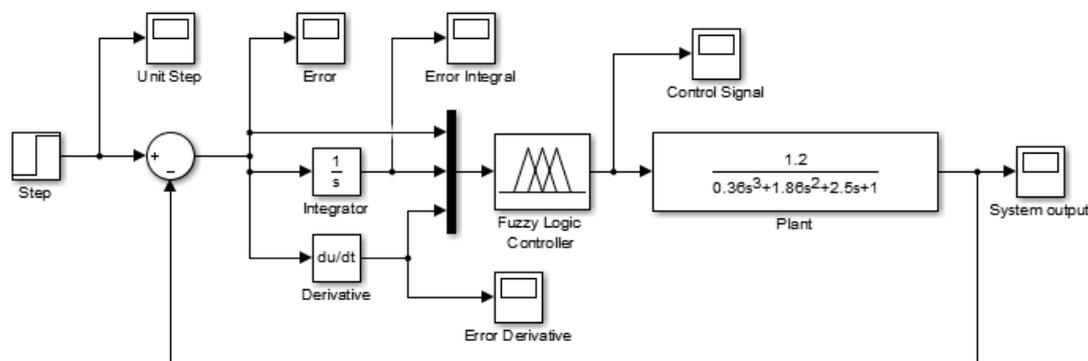


Figure 6. The equivalent FLC-controlled system in Simulink.

The “FIS Type” of the FLC design in Figure 6 should be set as “Sugeno”. Based on the above operating ranges of four fuzzy variables, the corresponding membership functions can be defined by Figure 1. The parameter m was set as 5, resulting in 5 fuzzy sets, which are equally-spaced and triangular-shaped, for each input fuzzy variable $e(t)$, $\int e(t)$, or $\dot{e}(t)$. On the other hand, the output fuzzy variable $u(t)$ is fuzzified by 13 singleton membership functions with singleton values $-18, -15, -12, \dots, -3, 0, 3, \dots, 12, 15$, and 18. Figure 7 shows the settings of all the membership functions in the Matlab environment. The 125 fuzzy rules are defined according to the knowledge base in Figure 2, and the resulting system input and responses are shown in Figure 5 (green line). However, it is found that the system responses with a green line cannot be examined, which is due to the overlap of system responses in the red line. In Figure 5, the green lines (responses by the equivalent FLC) were plotted prior to the red lines (responses by the PID controller). On the contrary, if we plot system responses by the PID controller (red line) first, it will turn out that all the red responses are covered by the latter green responses in the equivalent FLC. This verifies the proposed equivalence relationship between the PID controller and the equivalent FLC. Figure 8 further shows the linearity of the equivalent fuzzy PID controller with the control surface view under $\int e(t) = 0.36$.

For the equivalent FLC-controlled system in Figure 6, the Matlab/Simulink finishes the simulation in a manner of a continuous-time system. Alternatively, for practical applications, a FLC could be implemented in a discrete form. Therefore, to enhance the applicability of this research, the Sample and Hold unit and the Zero Order Hold unit, which can be used to model A/D (analog-to-digital) and D/A (digital-to-analog) converters, are added to make a discretized FLC, as shown in Figure 9. It must be noted that the Fuzzy Logic Controller in Figures 6 and 9 are identical. So, in Figure 9, the gain (K) in the “Discrete-Time Integrator” or “Discrete Derivative” does not denote K_P or K_I , and is set as 1. The system input and the resulting responses (blue line) are shown in Figure 5, with sampling period T_s .

set as 0.1 s. Simulation results show the equivalent FLC implemented in a discrete form have similar responses compared with the conventional PID controller. If the sampling period T_s is set to a smaller value, the closer of the two responses will be obtained.

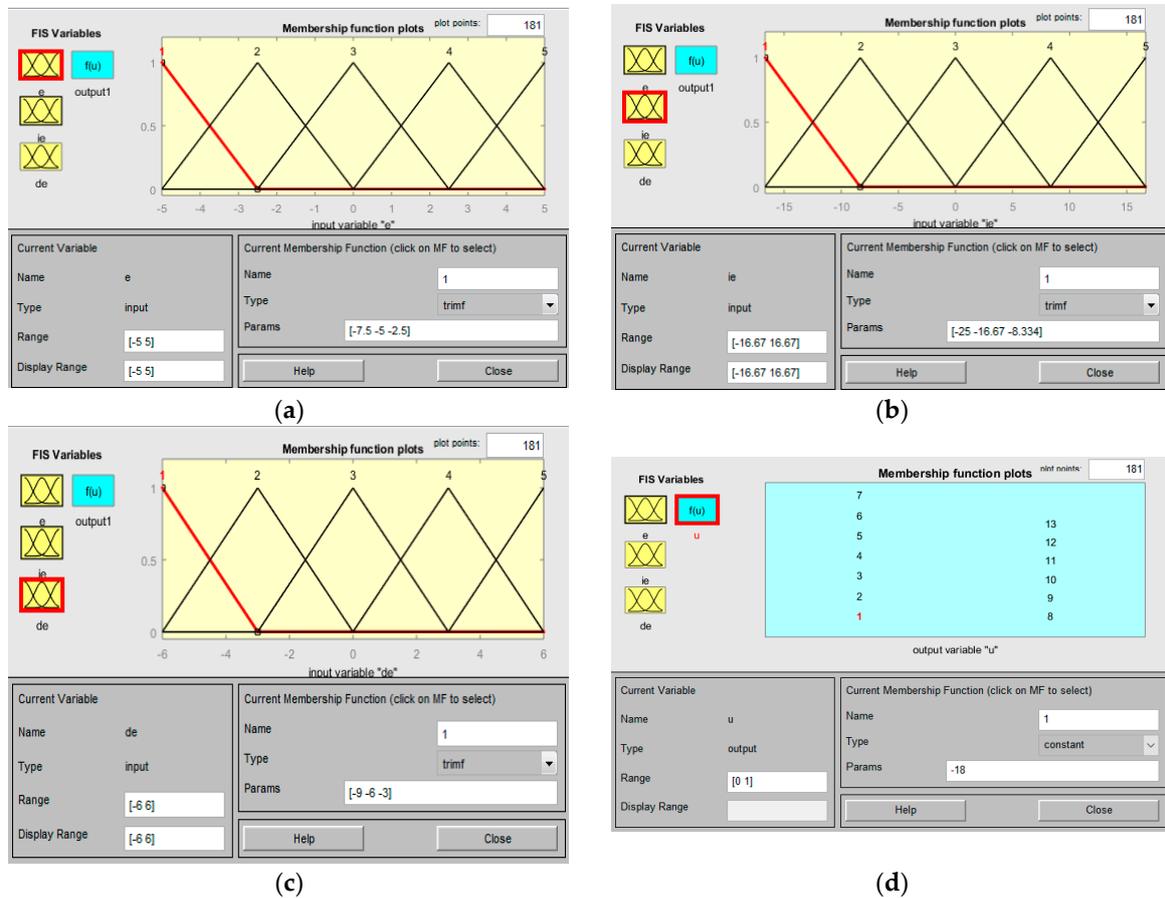


Figure 7. Membership functions for fuzzy variables (a) $e(t)$; (b) $\int e(t)$; (c) $\dot{e}(t)$; (d) $u(t)$ in Matlab.

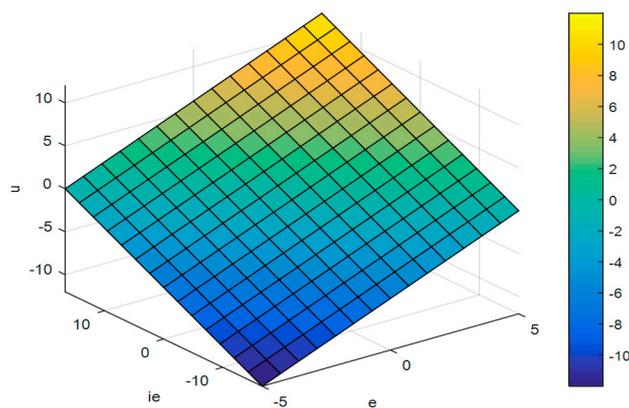


Figure 8. The control surface view of the equivalent fuzzy PID controller ($\int e(t) = 0.36$).

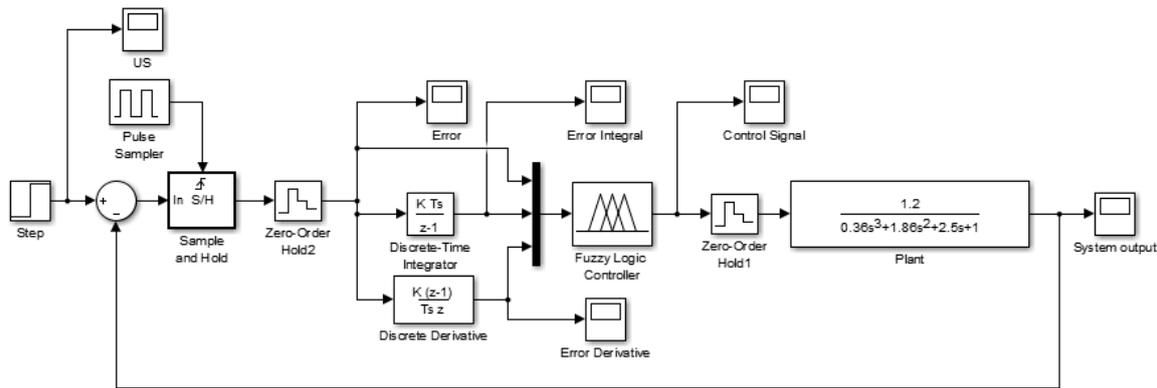


Figure 9. The equivalent FLC-controlled system in discrete form.

We slightly adjusted the membership functions (MFs) of the equivalent FLC design in Figure 7. It could be found that MFs are no longer equally-spaced, which implies that the FLC has become nonlinear, as shown in Figure 10. The 13 singleton values for the output fuzzy variable $u(t)$ are $-18, -16.91, -15.05, -12.44, -9.05, -4.91, 0, 4.91, 9.05, 12.44, 15.05, 16.91,$ and 18 . Figure 11 shows that the control surface view is no longer a plane. The resulting error signal, error integral, error derivative, control signal, and system output are shown in Figure 12 (blue line), and can be compared with the previous PID controller, and the equivalent FLC (red lines for both).

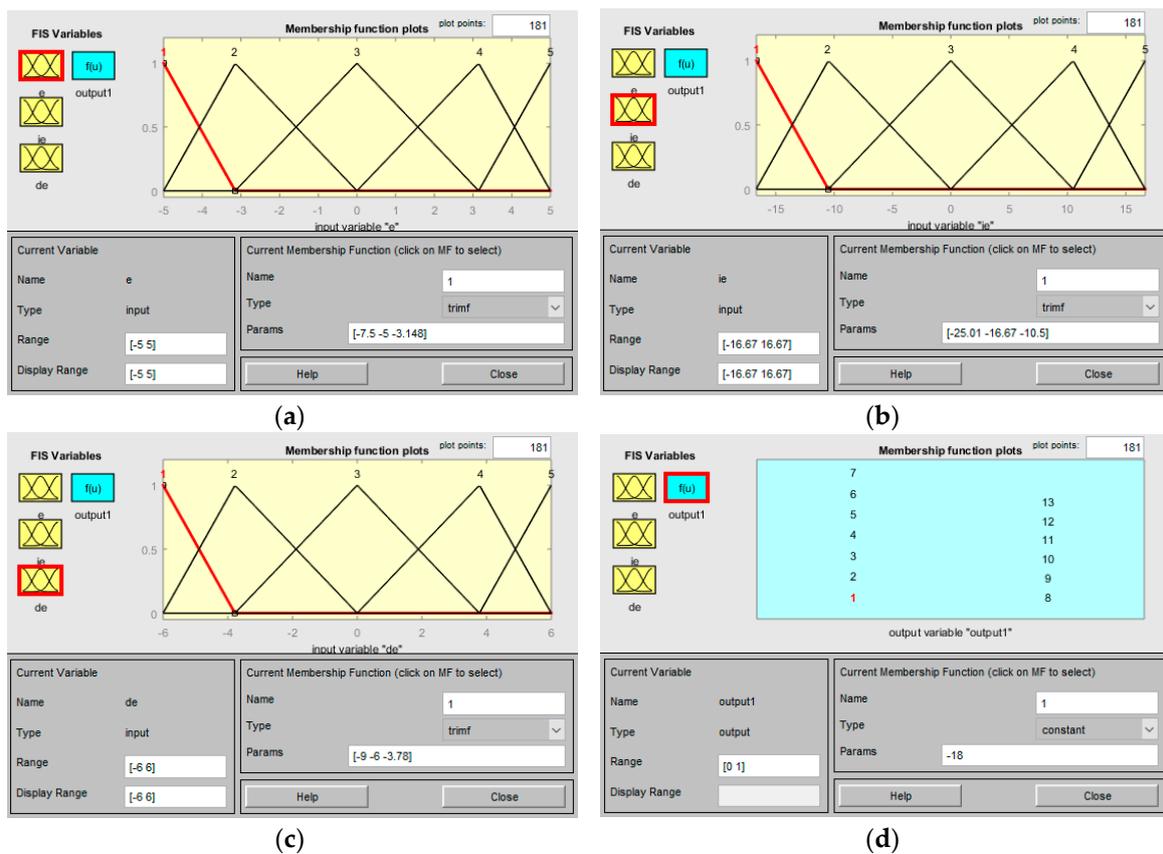


Figure 10. Membership functions adjustment for fuzzy variables (a) $e(t)$; (b) $\int e(t)$; (c) $\dot{e}(t)$; (d) $u(t)$ in Matlab.

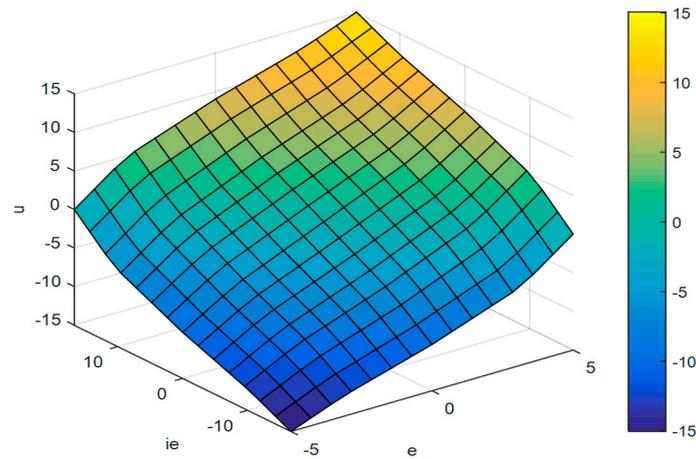


Figure 11. The control surface view of the equivalent fuzzy PID controller ($\int e(t) = 0.36$) under MFs adjustment.

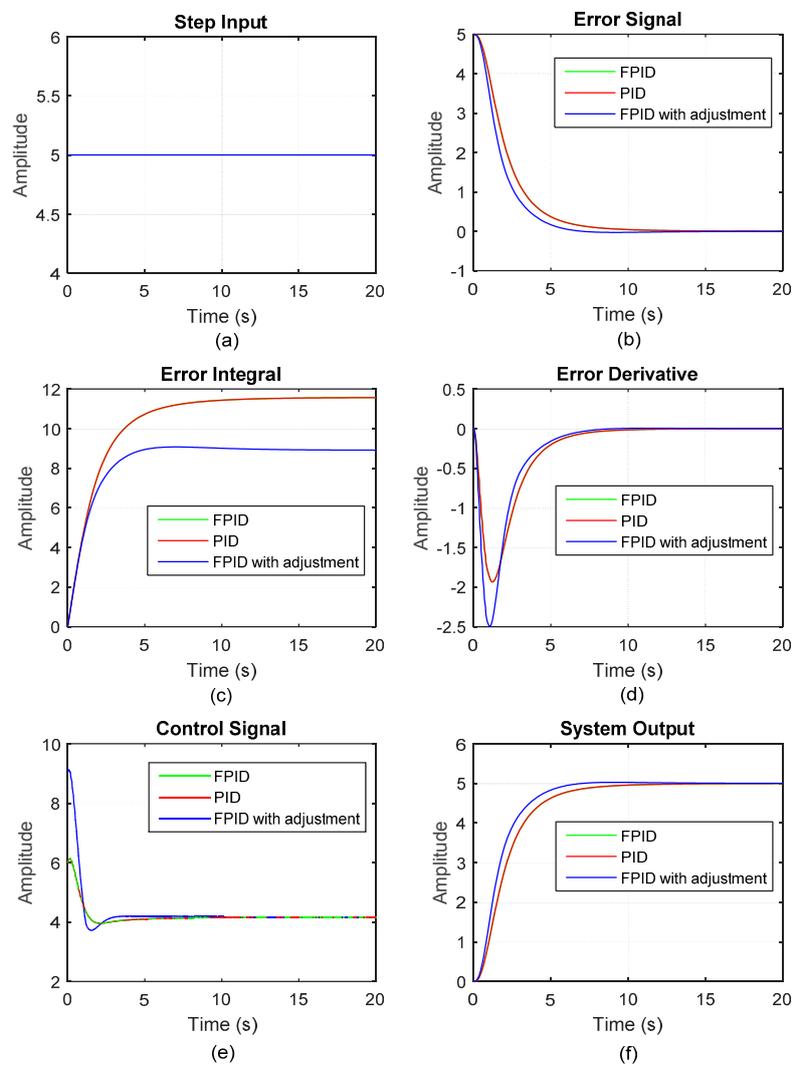


Figure 12. The (a) step input; (b) error signal; (c) error integral; (d) error derivative; (e) control signal and (f) system output with PID controller, the equivalent FLC, and the equivalent FLC under MFs adjustment.

Table 1 summarizes the response performance, including the rise time (Tr), the settling time (Ts), the percentage overshoot ($P.O.$), and the steady-state error (E_{ss}), of the three controllers in Figure 12. Which controller has better performance cannot be determined since the performance criterion is not defined. But based on this study, a fuzzy PID controller may outperform a conventional PID controller quickly by fine-tuning the MFs of the fuzzy variables.

Table 1. Response performance of different controllers.

Controller	Tr (s) 0.1–0.9	Ts (s) $\pm 5\%$	$P.O.$ (%)	E_{ss}
PID, FPID	3.75	5.76	0	0
FPID with adjustment	3.04	4.57	0.56	0

We have found that some learning-based techniques or evolutionary algorithms have been applied in the optimal FLC design [14–16]. Experienced researchers should agree on the importance of setting initial values or weights in the learning system, which will greatly influence the learning results and convergence speed. With the proposed equivalence relationship, one can easily and quickly obtain a fuzzy PID controller through a conventional PID controller design, then the derived equivalent FLC can be set as one of the initial designs. This process will result in the optimal FLC design in an efficient way.

4. Conclusions

This paper proposed and clearly identified the equivalence relationship between a conventional PID controller and a FLC. The derived equivalence equation is straightforward, so a well-designed conventional PID controller can be easily transformed to an equivalent FLC by simply defining the input/output operating ranges and following the Sugeno-style inference. The knowledge base for the equivalent FLC can be built by a cube FAM, instead of the combination structure of PI-type or PD-type FLCs. Simulation results demonstrate the effectiveness of the proposed approach, where system responses with the conventional PID controller or the equivalent FLC are similar. Moreover, the equivalent FLC implemented in discrete form was also provided and simulated, with comparable system responses to the original conventional PID controller. Based on the result of this study, the mature design results of traditional PID controllers can be applied as prior knowledge for an FLC design. Subsequently, a nonlinear FLC can outperform a traditional linear PID controller by changing fuzzy rule design or fuzzy membership functions. The proposed equivalent FLC can be set as the initial design for some learning-based techniques or evolutionary algorithms, which may achieve the optimal FLC design and considerably improve the convergence speed. Our future work will involve developing an optimal fuzzy PID controller with a simpler structure and fewer parameters, which will be designed to be more efficient for practical applications.

Acknowledgments: The authors would like to express the sincere thanks to the reviewers for their invaluable comments and suggestions. This work is supported by the Ministry of Science and Technology, Taiwan, under Grant no. MOST 103-2221-E-218-027.

Author Contributions: Juing-Shian Chiou and Chun-Tang Chao developed the methodology and drafted the manuscript. Moreover, Nana Sutarna and Chi-Jo Wang implemented the Matlab/Simulink simulations. The authors approved the final manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ziegler, J.G.; Nichols, N.B. Optimum settings for automatic controllers. *Trans. ASME* **1942**, *64*, 759–768. [[CrossRef](#)]
2. Mamdani, E.H.; Assilian, S. An experiment in linguistic synthesis with a fuzzy logic controller. *Int. J. Man Mach. Stud.* **1975**, *7*, 1–13. [[CrossRef](#)]

3. Sugeno, M. *Industrial Applications of Fuzzy Control*; Elsevier Science Inc.: New York, NY, USA, 1985.
4. Mudi, K.R.; Pal, R.N. A self-tuning fuzzy PI controller. *Fuzzy Sets Syst.* **2000**, *115*, 327–388. [[CrossRef](#)]
5. Oh, S.K.; Jang, H.J.; Pedrycz, W. Optimized fuzzy PD cascade controller: A comparative analysis and design. *Simul. Model. Pract. Theory* **2011**, *19*, 181–195. [[CrossRef](#)]
6. Chao, C.T.; Teng, C.C. A PD-like self-tuning fuzzy controller without steady-state error. *Fuzzy Sets Syst.* **1997**, *87*, 141–154. [[CrossRef](#)]
7. Pitalua-Díaz, N.; Herrera-López, E.J.; Valencia-Palomo, G.; González-Angeles, A.; Rodríguez-Carvajal, R.A.; Cazarez-Castro, N.R. Comparative analysis between conventional PI and fuzzy logic PI controllers for indoor Benzene concentrations. *Sustainability* **2015**, *7*, 5398–5412. [[CrossRef](#)]
8. Moon, B.S. Equivalence between fuzzy logic controllers and PI controllers for single input systems. *Fuzzy Sets Syst.* **1995**, *69*, 105–113. [[CrossRef](#)]
9. Kang, C.S.; Hyun, C.H.; Kim, Y.T.; Baek, J.; Park, M. A design of equivalent PID structure control using Fuzzy gain scheduling. In Proceedings of the 10th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI), Jeju, Korea, 30 October–2 November 2013; pp. 354–356.
10. Mann, G.K.I.; Hu, B.G.; Gosine, R.G. Analysis of direct action fuzzy PID controller structures. *IEEE Trans. Syst. Man Cybern. B* **1999**, *29*, 371–388. [[CrossRef](#)] [[PubMed](#)]
11. Hu, B.G.; Mann, G.K.I.; Gosine, R.G. A systematic study of fuzzy PID controller-function-based evaluation approach. *IEEE Trans. Fuzzy Syst.* **2001**, *9*, 699–712.
12. Manikandan, R.; Arulprakash, A.; Arulmozhival, R. Design of equivalent fuzzy PID controller from the conventional PID Controller. In Proceedings of the IEEE International Conference on Control, Instrumentation, Communication and Computational Technology (ICCICCT), Thuckalay, India, 18–19 December 2015; pp. 356–362.
13. Li, H.X.; Philip-Chen, C.L. The equivalence between fuzzy logic systems and feedforward neural networks. *IEEE Trans. Neural Netw.* **2000**, *11*, 356–365. [[PubMed](#)]
14. Chiou, J.S.; Tsai, S.H.; Liu, M.T. A PSO-based adaptive fuzzy PID-controllers. *Simul. Model. Pract. Theory* **2012**, *26*, 49–59. [[CrossRef](#)]
15. Pelusi, D. PID and intelligent controllers for optimal timing performances of industrial actuators. *Int. J. Simul. Syst. Sci. Technol.* **2012**, *13*, 65–71.
16. Pelusi, D.; Mascella, R. Optimal control algorithms for second order systems. *J. Comput. Sci.* **2013**, *9*, 183–197. [[CrossRef](#)]
17. Ogata, K. *Modern Control Engineering*, 5th ed.; Prentice Hall: Upper Saddle River, NJ, USA, 2010; p. 583.



© 2017 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).