PID Family and Fuzzy Logic Control of Real Non-Linear System

Derek C. Boase[1], Clive A. Plante[2],

[1]Student: Electrical and Computer Engineering, University of Ottawa, Ottawa, Canada, dboas065@uottawa.ca

[2]Student: Electrical and Computer Engineering, University of Ottawa, Ottawa, Canada, cplan082@uottawa.ca

***Abstract*—This paper compares the response characteristics of PID controllers and Fuzzy logic controllers as they are applied to a real, non-linear systems. Two PID controllers, classic PID and Pessen-Integral, were implemented using the Zeigler-Nichols tuning method. Conversely, two single-input single-output (SISO) fuzzy logic controllers were implemented, one having three membership functions and the other having five membership functions. The results were compared on the basis of rise time, settling time and steady state error.**

***Index Terms*—Zeigler-Nichols, PID controllers, Fuzzy-logic controllers, model-less systems, non-linear systems**

I. Introduction

Modern PID controls differ from classical PID control theory in that the software implementation of PID controllers is performed algorightmically. This means that a system may adapt its gain values rather than have fixed gains that are determined by hardware, traditionally op-amp circuits.[1][2] The digitization of PID controllers allowed this types of controllers to be implemented in circuits and tuned simply, without the need to redesign the circuit itself. This lead to the need for a systematic approach to tune controllers of the PID family, and although many such approaches are applicable, the Zeigler-Nichols is among the most popular method in industry.

On issue that still arises from PID controllers, despite the digitization, is the linked response characteristics. For example, to eliminate steady-state error and increase response times the integral gain should be increased, however this leads to oscillation and overshoot. To correct this oscillation and overshoot, the derivative gain should be increased, which leads to slower response times and the reintroduction of steady-stead error over finite time intervals. [1]

This problem is mitigated with the implementation of Fuzzy logic controllers. Fuzzy logic controllers allow for more system-specific responses by defining system outputs given certain system inpust. Although this gives the designer more control over the response, it requires an intuitive understanding of the system in order to implement.

In this paper, a classic PID controller and Pessen-Integral controller will be designed using the Zeigler-Nichols tuning method and compared to a three membership function and five membership function single-input single-output fuzzy logic controller.

The objective of the controllers was to control the input of a 12 V computer fan in order to elevate a Styrofoam ball to a certain target referred to as a set-point. This was done by calculating the numerator associated with the duty cycle calculation of an pulse-width modulated (PWM) digital output from an Arduino Uno and then controlling the average voltage delivered to a computer fan through a motor driver, H-bridge circuit.

II. Literature Review

Many different techniques may be used to control as system. The familiar ones are PID control, which may be achieved using hardware of software techniques. Other types are intelligent control, machine learning, deep learning, neural networks and fuzzy logic. A benefit of the PID controller model is that it is inexpensive and easy to implement. A major drawback, however is that the tuning of the PID parameters isn't systematic. There are many algorithmic approaches, but they still rely heavily on trial and error.[3][4] Furthermore, when the PID parameters are to be designed an acceptable range of targets must be precisely dictated, for example rise-time, settling-time, overshoot, etc,... This means that deterministic values must lead to deterministic outputs, which is how control engineers traditionally think about computer-controlled systems.[5] In contrast, fuzzy logic is a methodology that allows for ranges of input values to map to ranges of output values through the use of membership functions. The input(s) may be associated to an output by a mapping referred to as a membership function. The membership of each input to each possible output means that the exact threshold values of the parameters need not be know, rather the ranges of the inputs need to be known (similar PID control), but their exact output need not be known. [5][2] This concept facilitates the design process by allowing the designer to the about the response in a non-deterministic fashion. As an example, the statement "If the target is kind of far away, drive kind fast" is an implementable statement with fuzzy logic, whereas this would have no meaning in the design of a PID controller.

It is possible to use intelligent methodologies to interpolate useful data as opposed to simply controlling system outputs. This is achieved using methodologies like Kalman filtering. Kalman filtering uses a two step approach to determine the validity of the data that is being read. This model uses a linearlized Gaussian model to compare the predicted sensor input to the observed input in order to better approximate the readings.[6] Although this experiment presents the ideal use for Kalman filtering, it will not be used in this paper due to lack of resources. It is recommended to anyone recreating this experiment to use a Kalman filter and compare the difference in the results.

Two types of PID controllers were used in this experiment, the classic PID and the Pessen Integral PID. In order to tune the PID controllers without an accurate system model [1] the Zeigler-Nichols method was used. According to the method, seven different types of controllers can be closely approximated with the knowledge of the value of the proportional gain of the system that produces strong and consistent oscillations in the output. This value is referred to as the critical gain, $K_{critical}$ and the period of oscillations associated with the critical gain is referred to as the critical period, $T_{critical}$. Once these two parameters are known, the classic PID and Pessen Integral PID gains can be calculated using the following formulae, respectively.[7][8]

$$K_{p_{classic}} = 0.6K_{critical} \tag{1}$$

$$K_{i_{classic}} = 1.2\frac{K_{critical}}{T_{critical}} \tag{2}$$

$$K_{d_{classic}} = 0.075K_{critical}T_{critical} \tag{3}$$

$$K_{p_{pessen}} = 0.7K_{critical} \tag{4}$$

$$K_{i_{pessen}} = 1.75\frac{K_{critical}}{T_{critical}} \tag{5}$$

$$K_{d_{pessen}} = 0.105K_{critical}T_{critical} \tag{6}$$

Following the implementation of Equations 1-6 the gains of the the system may be adjusted to find the required response empirically. The tuning process is summarized in Figure 1

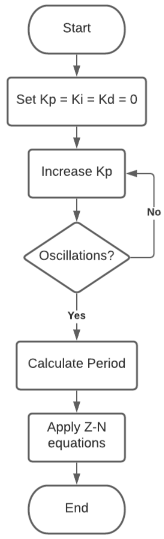

Fig. 1. Flow chart representing the process involved for the Zeigler-Nichols tuning method

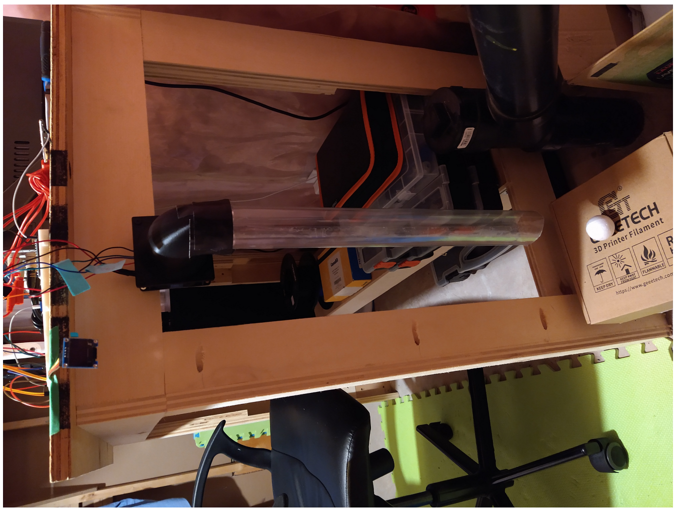

Fig. 2. Transparent tube with 3D printed fitting and 12 VDC computer fan

III. Experiment Setup

The objective of the experiment is to elevate a Styrofoam ball to a given set-point in a transparent plastic tube by controlling the average voltage input to a 12 VDC computer fan. The computer fan is coupled to the top of a 40 cm transparent cylinder through a 3D printed fitting that was designed specifically for this experiment. The fan is positioned such that it will create negative pressure inside the tube when it is on, lifting the ball upwards. At the outlet of the fan, there is a diffuser to encourage laminar flow and reduce back pressure at the top opening of the tube/fan connection. This system is shown in Figure 2 with the accompanying electronics shown in Figure 3.

Figure 3 shows an Arduino Uno connected to an HC-SR04 ultrasonic sensor, a l293dne motor driver and an organic light emitting diode (OLED) display. The ultrasonic sensor is attached to the top of the transparent cylinder and measures the distance from the top of the cylinder to the top of the ball. Based on the readings from the sensor, the control loop will determine the appropriate value of the numerator of the pulse-width modulation (PWM) which is directly proportional to the average voltage sent to the computer fan. Since the speed of the computer fan is approximately directly proportional to the average armature voltage, it can be said that the control loop directly influences the speed of the fan.

Since the measure that is of interest to this project is the height of the Styrofoam ball from the bottom of the transparent cylinder, a baseline height is calculated before the experiment begins. This is done by taking the average of multiple readings when the ball is at the bottom of the cylinder. Another reading is then taken and compared to the baseline height, if the absolute value of this reading is less

[1]No model was achieved due to the non-linear nature of the system

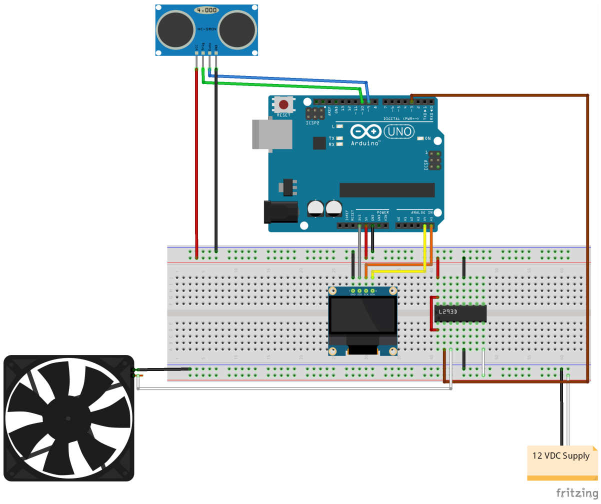

Fig. 3. Circuit diagram for the electronic components of the experimental setup.

than 1, the program continues with this value of the baseline, otherwise the baseline is re-calculated.

Once the baseline reading is found it is subtracted from all subsequent readings in order to determine the height of the ball from the bottom of the cylinder. The height is then passed to the control loop (either the PID control algorithm or fuzzy logic control algorithm) and duty cycle of the pulse-width modulation is set accordingly. Figure 4 shows the flowchart for the Arduino software.

The PID control algorithm was implemented using the ”PID_v2” library in the Arduino IDE. The PID control algorithm takes in height of the ball as the input to the and calculates the numerator of the duty cycle ratio based of the values of the height, K_p, K_i and K_d. The actual duty cycle of the PWM that is sent to the motor driver is then calculated by,

$$D = \frac{PID\ Output}{255}{}_{2} \tag{7}$$

The fuzzy logic controllers were implemented using the ”eFLL” library in the Arduino IDE. These controllers were tuned iteratively. This involved making guesses based on intuitive understanding of the system, evaluating the response and then trying different parameters. Each iteration gave the authors a better understanding of the systems response for each given change. The input to the fuzzy logic controller was the error, the difference between the current height and the set-point, and the output was the numerator of the duty cycle ratio, as was the case for the PID control algorigthm. The input and output membership functions are shown in Figures 5 - 8.

[2]The denominator is 255 since the period register of the Arduino Uno is 8-bit

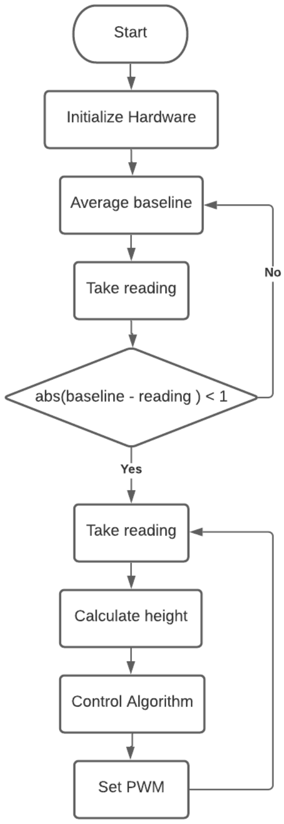

Fig. 4. Flow chart dictating the flow of the Arduino code

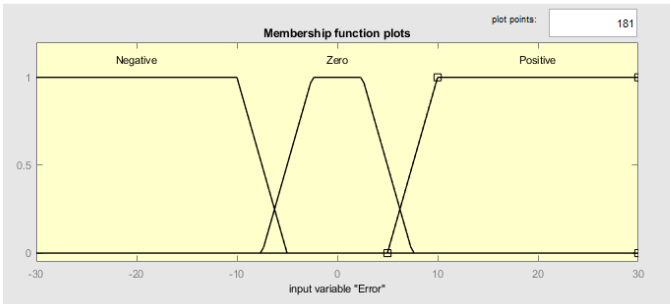

Fig. 5. Input functions of the three membership function controller

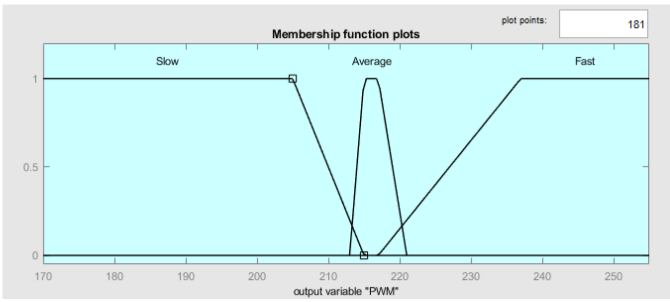

Fig. 6. Output functions of the three membership function controller

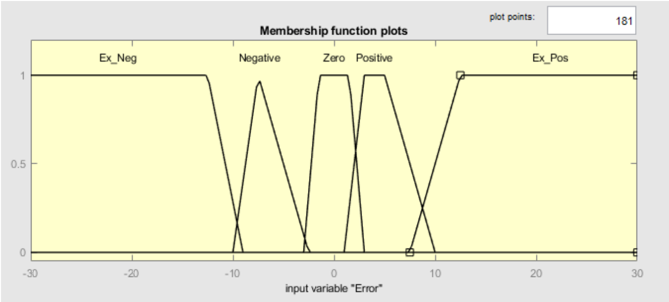

Fig. 7. Input functions of the five membership function controller

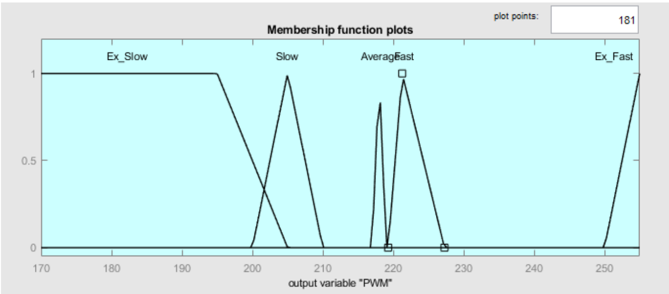

Fig. 8. Output functions of the five membership function controller

IV. Results

Following the procedure outlined in Section II, the following values were calculated for the classic PID and Pessen Integral PID controllers, respectively.

Controller Type	K_p	K_i	K_d
Classic PID	3.9	3.4106	1.1149
Pessen-Integral PID	4.55	4.9738	1.5609

TABLE I
TABULATED VALUES FOR CLASSIC PID AND PESSEN-INTEGRAL PID CONTROLLERS

To eliminate the noise associated with the inexpensive equipement which was used in this experiment, a 10th order median filtering approach was used on the data to ensure more accurate results. The filtered system responses of the classics PID controller, the Pessen-Integral PID controller, the three membership function fuzzy logic controller and five membership function fuzzy logic controller are shown in Figures 9 - 12, respectively.

The response characteristics of each of the controllers is tabulated in Table II.

Controller Type	Rise Time (s)	Settling Time (s)	Peak Height (cm)
Classic PID	1.98	12.17	30.8
Pessen-Integral PID	1.17	13.9	31.5
3 Membership	1.41	14.9	25.3
5 Membership	1.17	3.6	24

TABLE II
TABULATED RESPONSE CHARACTERISTICS OF THE FOUR TYPES OF CONTROLLERS THAT WERE IMPLEMENTED

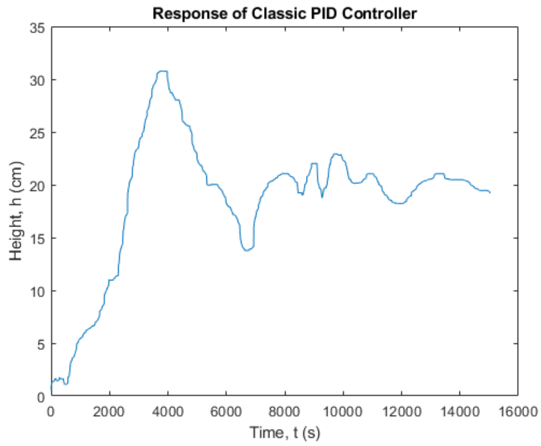

Fig. 9. Filtered system response of the classic PID controller

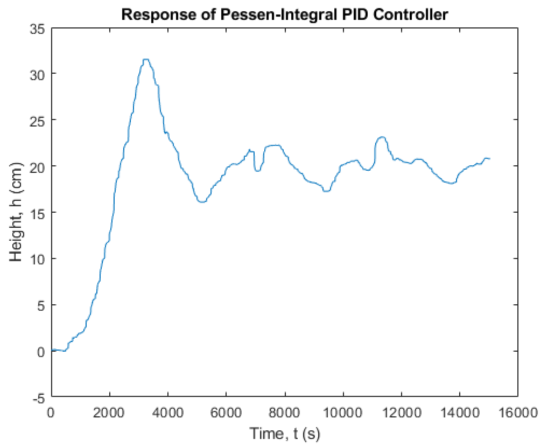

Fig. 10. Filtered system response of the Pessen-Integral PID controller

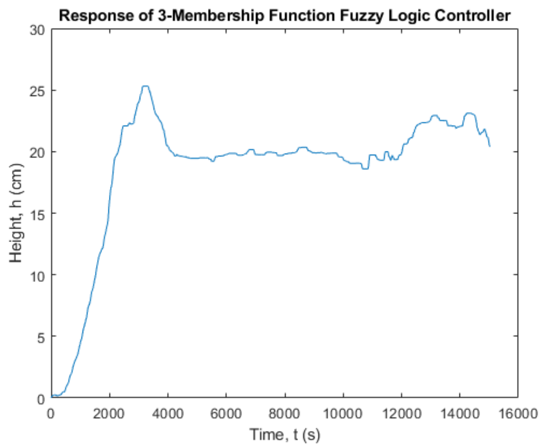

Fig. 11. Filtered system response of the three membership function fuzzy logic controller

It may be observed from the data that the rise times are all comparable, with the the Classic PID being the slowest with a rise time of approximately 2 seconds and the Pessen-Integral PID controller and five membership function fuzzy

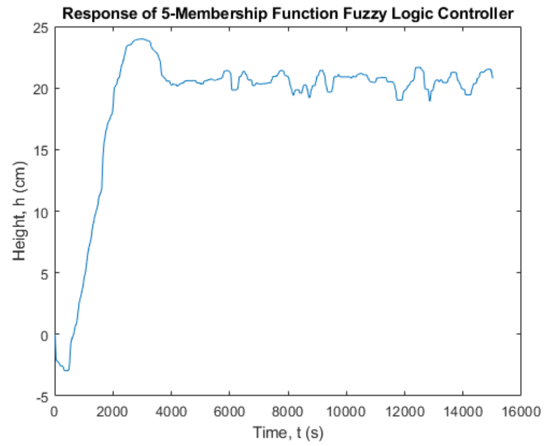

Fig. 12. Filtered system response of the 5 membership function fuzzy logic controller

logic controller being the quickest with an approximate rise time of 1.17 seconds. Comparing the settling times, it is clear that five membership function had the fastest settling time with approximately 3.6 seconds, whereas the other controllers all took more than 10 seconds to settle. It should be noted that a common error range associated with the settling time is ± 0.02, however due to the large fluctuation of the response, this value was increased to ± 0.04. Comparing the peak values, it is clear that the overshoot is much larger for the PID controllers than it was with the fuzzy logic controllers. In both of the PID controllers the peak value is approximately $30\ cm$ whereas the peak value of the fuzzy logic controllers is approximately $25\ cm$.

From the above, the five membership function fuzzy logic controller had the fastest rise time and settling time and the least amount of overshoot. It is the authors opinion that the settling time the fuzzy logic controller can be made to have a faster settling time, a tighter steady-state distribution around the set-point and have less overshoot by adding more membership functions around the set-point and by adding the fan speed as an input to the system.

V. Conclusion

In conclusion, the classic PID, Pessen-Integral PID and both fuzzy logic controllers were successful in utilizing feedback to eliminate the steady-state error, within a practical limit. The PID controllers were simpler to work with given the systematic approach that was dictated by the Ziegler-Nichols method. By following the steps and applying the algorithms, both PID controllers were fully functional and responded in accordance with the theory. The downfall of the PID controllers was that there was an inherent link between the rise-time, settling time and the peak height. For example, by decreasing the rise-time in the Pessen-Integral PID controller with respect to the classic PID controller, the peak value was increased and more oscillations were observed, leading to a longer settling time. By implementing the fuzzy logic controller, the designer gains more control over the response of the system at the expense of complication. Designing the parameters of the fuzzy logic controller required a stronger intuitive understanding of the system as it was iterative and did not have a systematic approach. The fuzzy logic controllers did resolve the link between the response parameters to some extent, as the controller with the fastest rise time also had the fastest settling time and smallest peak height. For the experiment presented in this paper, the five membership function fuzzy logic controller was the better controller in all measurable regards.

Acknowledgment

The authors would like to thank Dr. Riadh Habash for the opportunity to pursue an open-ended learning approach and for assisting in the research process and Hossein Ebrahimi Dinaki for reviewing and improving our work. Additional thanks goes to the University of Ottawa for access to the simulation software used in conducting this research.

References

[1] N. Nise. S, *Control Systems Engineering*. Hoboken, NJ: Wiley, 2019.
[2] P. Payeur, *Computer Control in Robotics*. Ottawa: University of Ottawa, 2020.
[3] M. Muhammad Sabir and j. a. Khan, "Optimal design of pid controller for the speed control of dc motor by using metaheuristic techniques," *Advances in Artificial Neural Systems*, vol. 2014, pp. 1–8, 12 2014.
[4] W. Khoury and P. Szemes, "Robust control system design for a brushed direct current motor using labview simulation loop," *ANNALS OF THE ORADEA UNIVERSITY. Fascicle of Management and Technological Engineering.*, vol. Volume XXVII (XVII), 2018/2, 01 2018.
[5] Z. TIR, O. Malik, M. Hamida, H. Cherif, Y. Bekakra, and A. Kadrine, "Implementation of a fuzzy logic speed controller for a permanent magnet dc motor using a low-cost arduino platform," 10 2017.
[6] A. Al Tahtawi, "Kalman filter algorithm design for hc-sr04 ultrasonic sensor data acquisition system," *IJITEE (International Journal of Information Technology and Electrical Engineering)*, vol. 2, 07 2018.
[7] M. A. Mat Dzahir, M. Hussein, B. Supriyo, K. Tawi, M. Yaakob, M. A. Mat Dzahir, and M. Mohamad, "Optimal tuning of a pid controller for emdap-cvt using particle swarm optimization," *Jurnal Teknologi*, vol. 75, 08 2015.
[8] J. G. Zielger and N. B. Nichols, "Optimum settings for automatic controllers," *Transactions of the ASME*, vol. 64, pp. 759–768, 1942.