International Journal of Engineering Applied Sciences and Technology, 2021
Vol. 5, Issue 9, ISSN No. 2455-2143, Pages 184-188
Published Online January 2021 in IJEAST (http://www.ijeast.com)

PID TUNING OF FOPDT SYSTEM USING MULTIVARIATE LINEAR REGRESSION WITH GRADIENT DESCENT

Thaker Maharsh Kalpeshbhai
ME Student - Applied Instrumentation
Instrumentation and Control Department
L. D. College of Engineering, Ahmedabad, Gujarat, India

Patel Vinod Purushottamdas
Associate Professor
Instrumentation and Control Department
L. D. College of Engineering, Ahmedabad, Gujarat, India

Abstract— **PID controllers are popular in industries for controlling process of a system or a plant. There are many different tuning techniques available to tune a PID controller. In this paper machine learning and optimization approach is used to tune the controller. Multiple Linear Regression algorithm of Machine Learning and Gradient Descent based optimization algorithm is used to obtain PID parameters. PID tuning is performed on a PID controller of Blending Control System. This paper presents a method of obtaining FOPDT model from system's transient specifications, generating data-set of FOPDT model and then applying Multivariate Linear Regression with Gradient Descent to obtain optimal tuned values of K_p, K_i and K_d parameters of PID controller.**

Keywords— ***Multivariate Linear Regression, Gradient Descent, PID, FOPDT, Blending Control, Data-set Generation, Machine Learning, Optimization***

I. INTRODUCTION

After the advent of PID controllers in the early 20th century, control theory has developed to a great extent. From the age of Pneumatic Controllers to this age of Artificial Intelligence and Automation, PID controllers have constantly evolved and still holds the most prominent place in Industrial Control. Similarly PID tuning techniques have also adopted new technologies to simplify the tuning process and obtain better control results. Starting from classical tuning approaches like Ziegler-Nicolas tuning[1], Cohen-Coon Tuning[2]to modern tuning techniques like IMC tuning, Lambda Tuning[3] and Meta-heuristic tuning techniques like Genetic Algorithm[4], Particle Swarm Optimization[5], TLBO[6], etc. are different tuning techniques suggested by different researchers. Current trend in PID algorithm is towards Machine Learning, Deep Learning and Fuzzy Logic[7] implementation on PID controller. Machine Learning approach consists of different supervised and unsupervised techniques like Regression, Classification, etc. And deep learning approach is driven by Reinforcement Learning, Neural Networks[8], etc.

All these different approaches to tune PID controller have its own advantages and limitations. Classical tuning approach is most widely known but least used practically. For tuning a controller by ZN Tuning algorithm we require the system to reach on the verge of instability, which is practically not implementable. Many papers have been written on Meta-heuristic techniques for PID tuning but meta-heuristic techniques like Genetic Algorithm requires more computation time and gives no guarantee of finding an optimal solution in a fixed time, also GA may converge to local optima if system is very complex. Deep learning and reinforcement learning algorithms require large datasets and many repetitions/iterations to get a perfect tuning model [9]. These techniques are more complex for a simple linear system. Linear regression with gradient descent is studied in paper [10] and [11] for first order and second order system respectively. This paper presents a method to tune simple FOPDT models by Linear Regression with Gradient Descent.

II. THE PID CONTROLLER

The PID controller contains Proportional term, Integral term and Derivative term. Proportional term accounts for present error value, integral terms reduces offset error and derivative terms increases the speed of response. PI and P controllers are also widely used depending on the application. Figure 1 shows the block diagram of PID controller and system negative feedback arrangement.

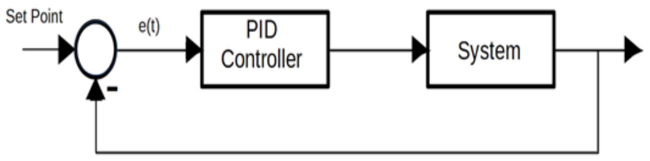

Fig. 1. PID Controller with Negative Feedback

Input to controller is error signal and controller output is given to system input. The mathematical form of PID controller is given by this equation:

$$u(t) = K_p e(t) + K_i \int_0^t e(t)dt + K_d \frac{de(t)}{dt} \qquad (1)$$

III. DEVELOPMENT OF FOPDT MODEL

A mixer tank has one inlet stream and one outlet stream. No reaction terms are considered for this application. It is assumed that mixing in the tank is perfect, so we can assume the concentration and temperature in the tank to be constant. The volume of the mixer tank is also assumed to be constant. By species balance we can derive the following equation of change in outlet concentration per unit time:

$$\frac{dC_a}{dt} = \frac{q}{V}(C_{af} - C_a) \qquad (2)$$

Where C_a is outlet concentration, C_{af} is feed concentration (inlet concentration). q is volumetric flow rate and V is the volume of tank. Units of all specifications are described in Table 1 [12]. The figure (2) below shows step response of mixing tank. Giving a step input to the Blender we obtain the as response shown in Figure 2. Table 2 shows the data of Time and Concentration values after providing step input to the system.

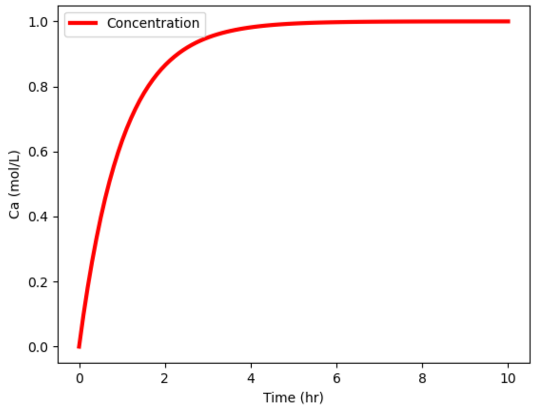

Fig. 2. Step Response Curve

A. FOPDT Model Generation –

Data of Concentration vs. Time is used to construct FOPDT (First Order Plus Dead Time) model. Time values are taken from 0 to 10 (in hours) with the steps of 0.0001 so 1000 values of concentration and time data are generated to obtain an accurate FOPDT model. Lesser the data less accurate will be modeling and identification of FOPDT system.

Table -1 Blender Specifications

Sr. No	Parameter	Value
1.	Volumetric Flow-rate	100 m3/hr.
2.	CSTR Volume	100 m3
3.	Feed Temperature	300 K
4.	Feed Concentration	1 mole/L

Using Smith's Method of identification of FOPDT model, the dead time (L) value of -0.333 and tau value of 0.222 is obtained. FOPDT identification is obtained by finding 63.2% and 28.3% values of time constants t_1 and t_2 respectively. Process dead time (L) and process time constant (tau) values are given by equations (3) and (4).

$$L = t_2 - t_1 \qquad (3)$$

$$tau = 1.5\,(t_2 - t_1) \qquad (4)$$

After performing step test we obtain the transfer function of FOPDT model shown in equation (5).

$$H = e^{-0.333s} \frac{1}{0.222s+1} \qquad (5)$$

Other popular FOPDT identification method is Sundaresan and Krishnaswamy [13] method to find values of dead time and tau. For this study Smith's FOPDT method is considered. As compared to Nishikawa's method [14] of FOPDT identification, Smith's method [15] and Sundaresan-Krishnaswamy method has lower computational cost.

Table -2 Time versus Concentration Data

Sr. No	Time	Concentration
1.	0.101	0.0960
2.	0.202	0.1829
3.	0.303	0.2614
4.	0.404	0.3323

IV. MULTIVARIATE REGRESSION WITH GRADIENT DESCENT

The Linear Regression is a mathematical approach which is based on modeling a system using linear relationship between input and output variables. Multivariate Linear Regression contains one output variable and multiple input variables. Linear Regression is a type of supervised learning technique. For this technique to function properly there should be linear relationship between input and output variable.

First finding closed loop transfer function. On this transfer function we apply Linear Regression that relates three input variables with one output variable. Transient specifications of system like Rise Time, Settling Time and Overshoot are used as input variables. Mathematical equation of Linear Regression is given by (6).

$$y = \sum_{n=0}^{j} \theta_{(j)} X_{(j)} \quad (6)$$

$\theta_{(j)}$ are weights associated with input variables $X_{(j)}$. j are the number of input variables. Gradient Descent algorithm minimizes the error between the predicted output variable and the actual output variable [6]. The error can be calculated using cost function given by equation (7).

$$J(\theta) = \frac{1}{2m}\sum_{j=0}^{m}\left(h(\theta)X_{(j)} - y_{(j)}\right)^2 \quad (7)$$

Where m is the number of iterations. After performing the partial derivation of cost function we get the update rule as given by equation (8).

$$\theta_{(i)} = \theta_{(i)} - \alpha\frac{1}{m}\sum_{j=0}^{m}\left(h(\theta)X_{(j)} - y_{(j)}\right)X_{(i)}^{j} \quad (8)$$

Where Alpha (α) is the Learning Rate. Learning rate being small, algorithm convergence will be slow and if the learning rate is very high then algorithm may miss the optimal point by converging faster than expected.

V. PROBLEM FORMULATION

Transfer function of closed loop system is created using FOPDT system and controller. This closed loop system comprises of controller, system (FOPDT) and feedback element in general. For training the model one of the parameter K_p, Ki and K_d is varied in a certain range and other two parameters are kept constant. For example, we vary K_d from 0.1 to 10 at steps of 0.1, keeping value of K_p at 100 and value of Ki at 7. Output is noted after giving a step input. For this study K_p, Ki and K_d are taken as output variables and Rise Time, Settling Time and Overshoot as considered as input variables. Transient parameters of rise time (t_{rise}), settling time (t_{set}) and overshoot (O_v) are obtained by using 'stepinfo' function of MATLAB. Similarly Ki is varied, keeping K_p and K_d constant. Then K_p is varied, keeping Ki and K_d constant. We obtain the optimal PID parameters by the given equations (9), (10) and (11) respectively.

$$K_p = t_{rise}\theta_1 + t_{set}\theta_2 + O_v\theta_3 \quad (9)$$

$$K_i = t_{rise}\theta_4 + t_{set}\theta_5 + O_v\theta_6 \quad (10)$$

$$K_d = t_{rise}\theta_7 + t_{set}\theta_8 + O_v\theta_9 \quad (11)$$

Theta are the optimal weights associated to the input variables. Other transient input variables like Peak and Peak time can be added to work with five input variables.

VI. SIMULATION

Sample Dataset for K_p: Table 2 shows sample data-set for K_p obtained by keeping Ki and K_d constant at 2.67 and 0.13 respectively. K_p is varied from 0 to 3.25 with the steps of 0.01 and transient response is recorded in a '.csv' file.

Table -3 Sample of Kp Dataset

K_p	Rise Time	Settling Time	Overshoot
0	0.662	5.72	35.99
0.01	0.66	5.69	35.4
0.02	0.659	5.67	34.81
0.03	0.657	5.63	34.23

Sample Dataset for Ki: Table 4 shows sample data-set for Ki obtained by keeping K_p and K_d constant at 1.62 and 0.13 respectively. Ki is varied from 0 to 5 with the steps of 0.05 and transient response is recorded in a '.csv' file.

Table -4 Sample of Ki Dataset

K_i	Rise Time	Settling Time	Overshoot
1	0.1719	6.55	0
1.05	0.1712	6.20	0
1.1	0.1691	5.87	0
1.15	0.1685	5.58	0

Sample dataset for K_d: Table 5 shows sample data-set for K_d obtained by keeping Ki and Kp constant at 2.49 and 1.26 respectively. K_d is varied from 0 to 0.25 at the steps of 0.001 and transient response is recorded in a '.csv' file.

Table -5 Sample of Kd Dataset

K_d	Rise Time	Settling Time	Overshoot
0	0.1844	1.93	19.30

0.001	0.1849	1.92	18.97
0.002	0.1852	1.91	18.65
0.003	0.1854	1.90	18.33

After generating the data-sets for Kp, Ki and Kd, model training is performed using multivariate linear regression with gradient descent algorithm. The result of the training is shown in Table 6 and compared with the results of MATLAB Auto-Tuner.

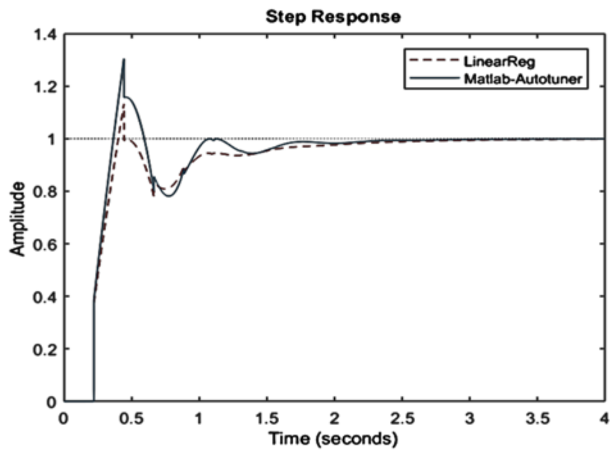

Fig. 3. Tuned PID Response

Table -6 Output Parameters

Method/Parameter	K_p	K_i	K_d
Linear Regression	1.62	2.99	0.13
MATLAB Auto-Tuner	1.9	3.03	0.12

Comparing the PID response obtained by linear regression with gradient descent algorithm versus the response obtained by MATLAB Auto-tuner we find that linear regression with gradient descent works better for this application. PID response obtained by linear regression with gradient descent has less overshoot as compared to that of the response obtained by MATLAB Auto-Tuner. Figure 4 shows the response of system to PID controller parameters tuned by linear regression with gradient descent.

Convergence curve of gradient descent algorithm shows that algorithm correctly converged after 2000 iterations. Figure 3 shows the convergence curve of gradient descent. The learning rate affects the convergence of the curve. Table 7 shows the effect of hyper-parameter (learning rate and number of iterations)changes on the K_p value.

If we change number of input variable it does not affect much on the PID tuning performance. Till now we were using three input parameters now after adding two more input parameters (peak and peak time) the result obtained is almost same as with three input parameters.

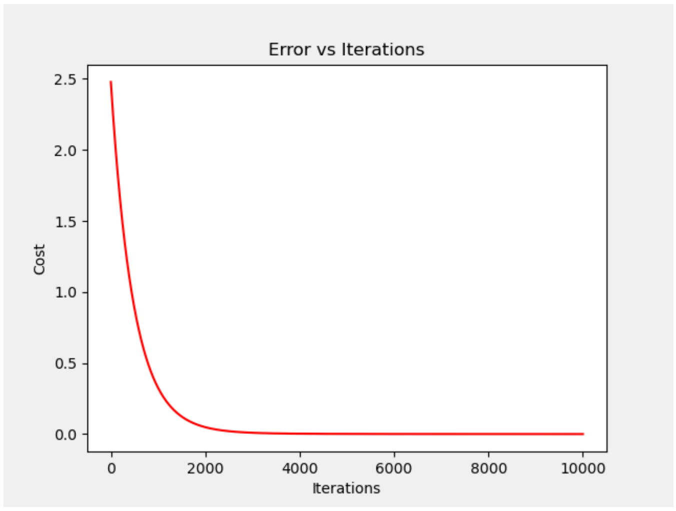

Fig. 4. Gradient Descent Convergence Curve

Table -7 Hyper-Parameter Effect on Kp

Alpha Value	K_p **at 500 Iterations**	K_p **at 3000 Iterations**	K_p **at 10000 Iterations**
0.01	1.614	1.624	1.625
0.005	1.492	1.624	1.625
0.001	0.639	1.54	1.624
0.0001	0.079	0.421	1.072

From the data of learning rate (Alpha) and number of iterations of gradient descent it is clear that small values of learning rate requires more number of iterations to achieve proper convergence. If the value of alpha is large and number of iterations are less than algorithm training will result in incomplete convergence and tuning will not take place as expected.

Figure 4 shows how the PID controller response changes with change in number of iterations. For this response the value of learning rate selected is 0.005. Vale of n equal to 500 and Alpha equal to 0.005 gives less overshoot as compared to n = 3000 with α = 0.005 and n = 10000 with α = 0.001. Using optimal value to alpha and number of iterations we can perform fine tuning of PID controller.

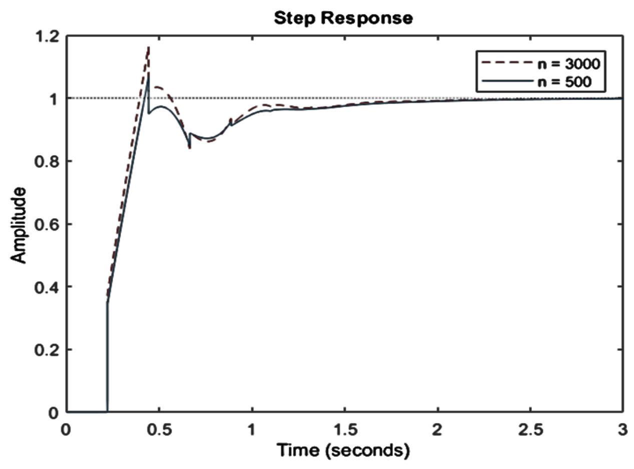

Fig. 5. PID Controller Response on Variation in Iterations

VII. CONCLUSION

Very less research work is done till now on tuning PID parameters with linear regression techniques so there is vast scope of research in the field of PID controller tuning. This study shows the result of PID tuning of FOPDT system using multivariate linear regression with gradient descent algorithm, this tuning technique can be applied to variety of systems including first order, second order and FOPDT systems. Selection of optimal hyper-parameters results in fine tuning of the model. For tuning of Non-Linear systems using the technique presented in the paper, we can linearize the system at any one parameter and perform training of the model. Feasibility check of PID tuning of Non-Linear system can also be performed.

VIII. REFERENCE

[1] Ziegler J. G. and Nicolas N. B. (1942). Optimum Settings for Automatic Controllers. Transactions of the American Society of Mechanical Engineers. (Vol.64, p-p. 759-768).

[2] Cohen G. H. and Coon G. A. (1953). Theoritical Consideration of Retarded Control. Transactions of the American Society of Mechanical Engineers. (Vol.76, p-p. 827-834).

[3] K. J. Astrom and T. Hagglund. (1984). Automatic Tuning of Simple Regulators with Specifications on Phase and Amplitude Margins. Automatica Journal. (Vol.20, No. pp. 645-651).

[4] Meena D. C. and Devanshu Ambrish. (2017). Genetic Algorithm Tuned PID Controller for Process Control. IEEE International Conference on Inventive Systems and Control. https://doi.org/10.1109/ICISC.2017.8068639.

[5] K. Latha, V. Rajinikanth, and P. M. Surekha. (2013). PSO-Based PID Controller Design for a Class of Stable and Unstable Systems, ISRN Artificial Intelligence. (vol. 2013, Article ID 543607). https://doi.org/10.1155/2013/543607.

[6] P. Wang. And D. P. Kwok. (1992). Optimal Fuzzy PID Control based on Genetic Algorithm. Proceedings of the 1992 International Conference on Industrial Electronics, Control, Instrumentation, and Automation. (p-p 977-981).

[7] S. Akhyar and S. Omatu. (1993). Self-Tuning PID Control by Neural Networks. Proceedings of 1993 International Conference on Neural Networks (IJCNN-93-Nagoya, Japan). (p-p 2749-2752).

[8] R. Parvathy and Rani Devi. (2014). Gradient Descent Based Linear Regression approach for Modelling PID parameters. International Conference on Power, Signals, Controls and Computation (EP SCICON) https://doi.org/10.1109/EPSCICON.2014.6887482.

[9] Andrew Z. (2017). Towards Explicit PID Controller Tumimg using Machine Learning. IEEE Africon Proceedings. (p-p 430-433).

[10] Rukshan Pramoditha, (2017) Linear Regression with Gradient Descent, Data Science 365. https://www.researchgate.net/publication/342163679_Linear_Regression_with_Gradient_Descent.

[11] Shang X. Y. and Ji M. S. (2013). Parameter Optimization, of PID Controllers by Reinforcement Learning. 5th. Computer Science and Electronic Engineering, Conference (p-p. 77-81).

[12] Apmonitor Dynamics and Control - Tank Blending Article https://apmonitor.com/pdc/index.php/Main/TankBlending

[13] K. R. Sundaresan, P. Krishnaswamy. (1978). Estimation of Time Delay in Time Constant Parameters in Time, Frequency and Laplase Domain, in Canadian Journal of Chemical Engineering. https://doi.org/10.1002/cjce.5450560215

[14] Nishikawa, H. (2007). A first-order system approach for diffusion equation. i: Second-order residual-distribution schemes. Journal of Computational Physics, Elsevier. (v. 227, n. 1, p. 315–352).

[15] SMITH, C. L. (1972). Digital computer process control. [S.l.]: Intext Educational Publishers.