Review on Proportional-Integral-Derivative Controllers

By Zhenyu Dai

Student ID: 10830268

Control Engineering

Coursework

School of Mechanical, Aerospace and Civil Engineering

University of Manchester

Abstract

This paper provides a brief literature review of PID(Proportional-Integral-Derivative) controllers. It has included origins of PID control, description of the PID control algorithm, ideal and standard forms of PID control, limitations and improvement of the algorithm.

1. Introduction

Control systems can be divided into two main categories, which are open-loop control systems and closed-loop control systems. An example of their block diagram is shown in Figures 1A and 1B. For open-loop systems, after selecting the input, the system functions and generates an output. Further alterations are not made and there will not be any response with regards to the output. The input then needs to be selected for the system based on experience of what gives the desired output. These systems are generally simple, reliable and have low costs, but they can be inaccurate due to the absence of correction of errors[1]. Therefore, closed-loop control systems have been developed to fulfil the need for accuracy. The closed-loop control system has similar input to the open-loop control system, the difference between them is that for closed-loop control system, the output is measured and compared with the required value, while the difference(the error) is recorded and used to modify the input. As an ideal result, the error would be eliminated, the system would then produce the required output.

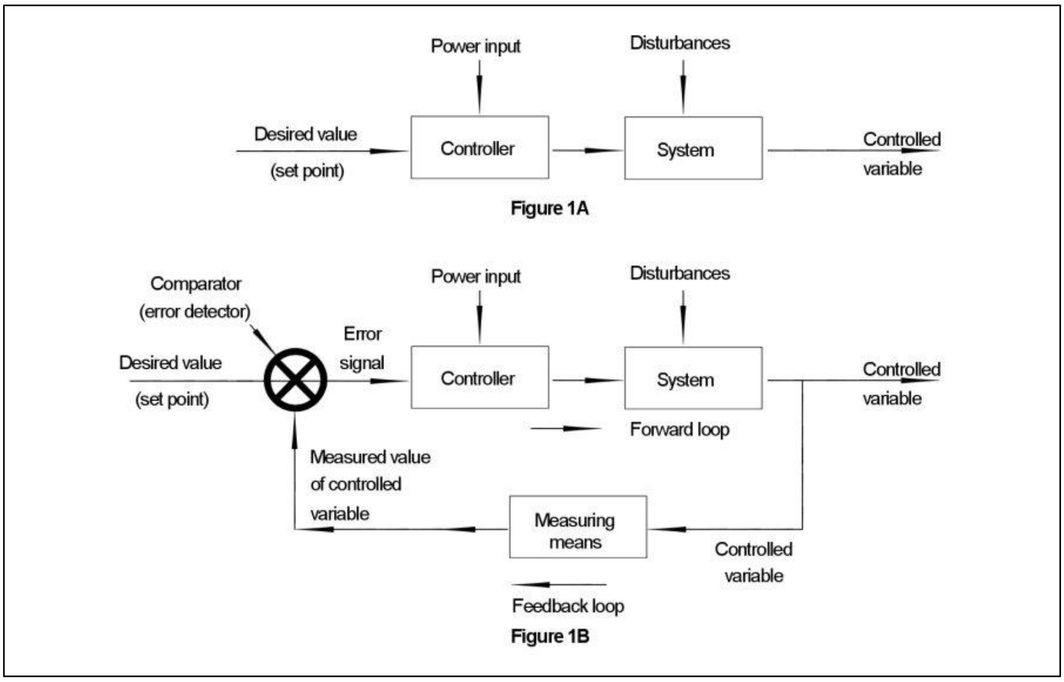

Figure 1A and 1B illustrate block diagrams of open loop and closed loop control system[2]

PID controllers are an example of closed-loop control systems, which use proportional, integral and derivative terms to alter the feedback loop[3]. In process control applications more than 95% of the controllers are of PID type. The maintenance and operation of PID controllers are easy and they are robust in nature. An example of process control using PID is shown in Figure 2.

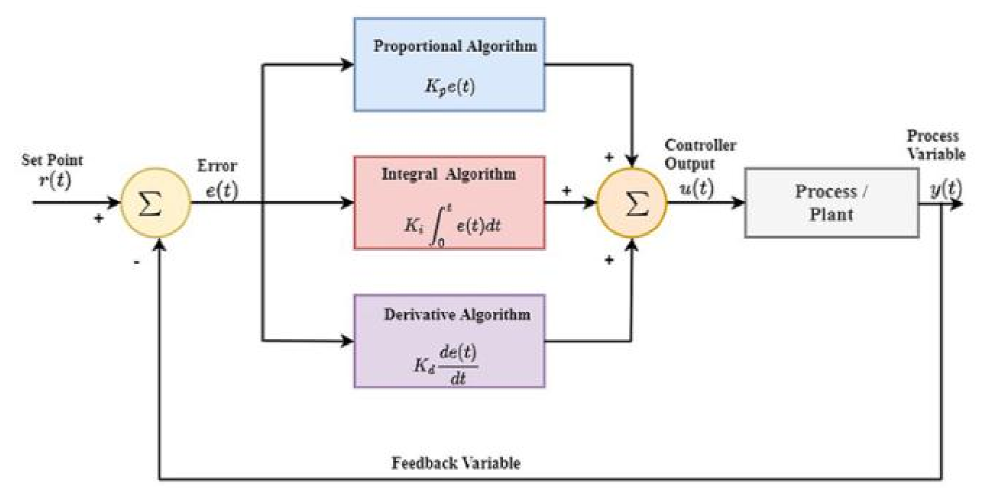

Figure 2

Figure 2 illustrates the block diagram of process control using PID[4]

As shown in Figure 2, PID control algorithm is divided into three parts, which are proportional, integral and derivative respectively. They each represent the present, past and future aspects of error. There will be a more detailed description of the three parts in later section Description of PID Control Algorithm.
Moreover, there are a series of applications in multiple fields where PID is used. One of the most common examples is its application in Real-Time Locating System(RTLS)[5]. In RTLS, high accuracy is required within a small coverage area. Conventional direct calculation methods for RTLS can be computationally heavy and less accurate, thus to address this issue, PID control algorithm is used. As a result, this approach would reduce the computational load and increase the robustness of the RTLS significantly.

2. Origins of PID Control

The PID controller is one of the earliest and most popular closed-loop controllers in the field of industrial control[6]. The simplicity, efficiency, and robustness of PID controllers have contributed to the phenomenon.
The first theoretical analysis of PID controllers was published by Nicolai Fyodorovich Minorsky (1885–1970), a Russian-born naval engineer, in his work «Directional stability of automatically steered bodies» in 1922[7]. Initially, the purpose of Minorskly was to develop an automatic ship steering system for the U.S. Navy based on observations of the steersmen's use of current error, past error, and rate of change to keep the ship on course[8]. Figure 3 shows the block diagram of

Minorsky's automatic steering system.

The practical origins of PID can be traced back to the 1930s. It is claimed that the first practical three-term controller in industry was introduced by the Taylor Instrument Company in 1936 when preact, that is, derivative action was added to their double response controller[9]. However, the amount of the preact term was initially fixed in the factory, until in 1939, a controller with a continuously variable derivative action was introduced by the company.

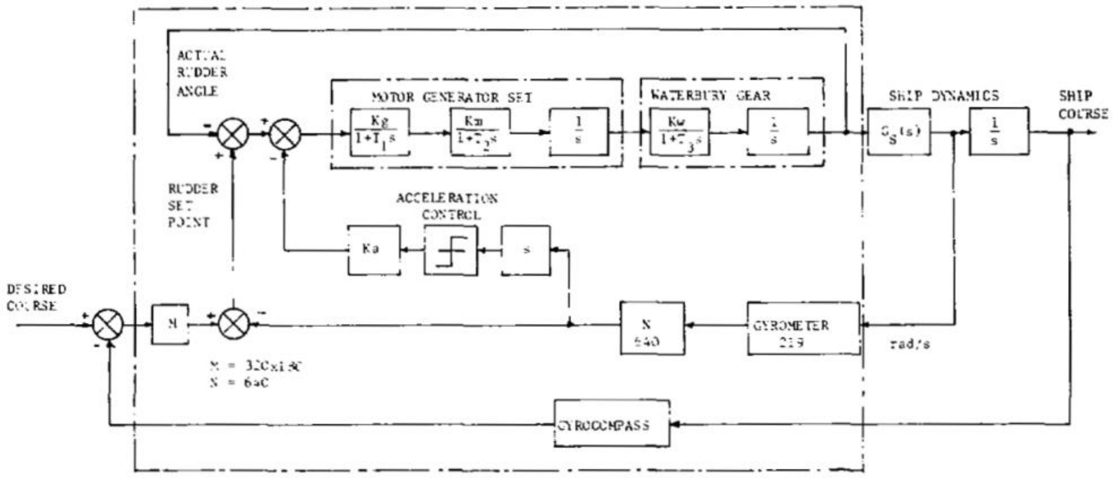

Figure 3

Figure 3 illustrates the block diagram of Minorsky's automatic steering system[9]

3. Description of PID Control Algorithm

As mentioned in the Introduction, PID control algorithm is divided into three parts, which are proportional, integral and derivative respectively. The three parts all contribute to determining the closed-loop response of the algorithm and all are important to ensure the robustness of the control output.

3.1 Proportional

As mentioned in the Introduction, in a closed-loop control system, the error(difference between the actual value measured and the targeted value) is used to modify the input[1]. In the proportional part, an amplification factor K_p is used to control the rate of the adjustment made to the input. The formula for the proportional part is shown below in equation(1), where p(t) is the function for Proportional Control, as this is the adjustment to the system that is directly proportional to the error function, which is e(t). K_P is the amplification factor, it is also called Proportional Gain.

$$p(t) = K_P * e(t) \tag{1}$$

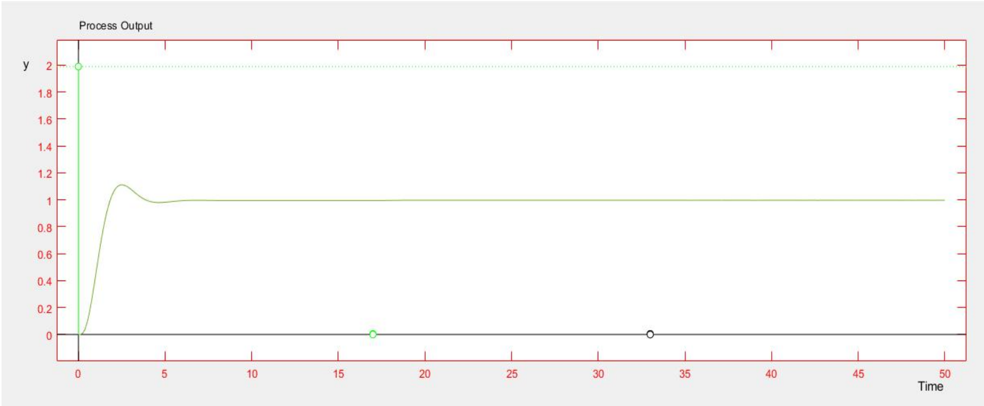

Figure 4

Figure 4 shows set point response from a P-controller simulation (in PID Basics) using K_P value of 1.0 and reference value y = 2.0[10]

One possible issue of using p-controller(controller that only has proportional part) is that residual error may arise due to changing choice of the initial input to the system. This error is nearly impossible to eliminate solely by proportional control. For example, as shown in Figure 4, a reference value of y = 2.0 is chosen for a p-controller simulation using K_P = 1.0 in the PID Basics simulator[10], giving a final stable process output of the p-controller of only around 1.0, showing a relatively important residual error is present. One solution to the situation is to add another term that sums instantaneous error over time, which could result in the residual error being eliminated.

3.2 Integral

The integral term, I is proportional both to the size of the error and to the duration of the error, as it is computed by the accumulation of error over time. The equation(2) is shown below, where K_I is the amplification factor of the integral term.

$$I = K_I * \int_0^t e(t)\,dt \tag{2}$$

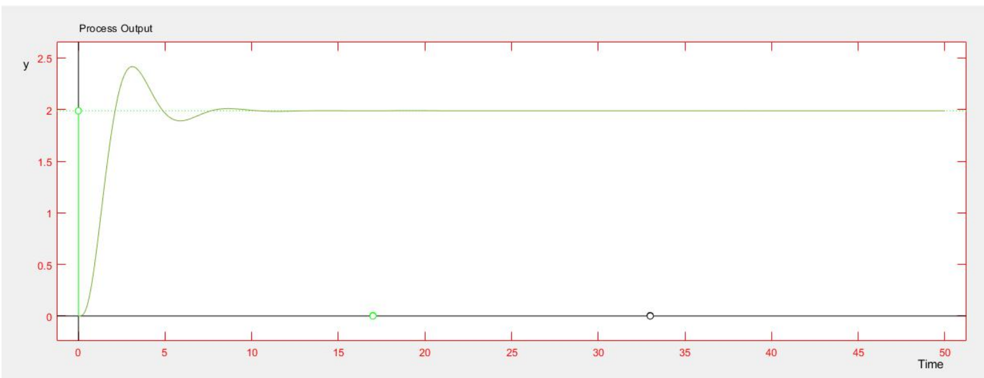

Figure 5

Figure 5 shows set point response from a PI-controller simulation(in PID Basics) using K_P value of 1.0, reference value y = 2.0 and K_I value of 1.0[10]

As shown in Figure 5, with the addition of integral term, the offset error is minimised and the final process output is close to 2.0. A PI(proportional-integral) system can be effective, but it will only respond after an error is present. It also has a large overshoot value, which is the over-reaction of the controller. It is shown in the region of Figure 5 that is within 0-5 time units. Therefore, in order to act before the error, as well as to slow the rate of the process output, derivative term is added to the control system.

3.3 Derivative

The derivative term is obtained from timing the gradient of the error function with the amplification factor, which is also the derivative gain. It therefore shows the rate of change of the error error function e(t). The formula(in equation 3) for the derivative part is shown below, where K_D is the derivative gain. The three amplification factors(gains) play an important role in PID control algorithms.

$$D = K_D * \frac{d}{dt} e(t) \tag{3}$$

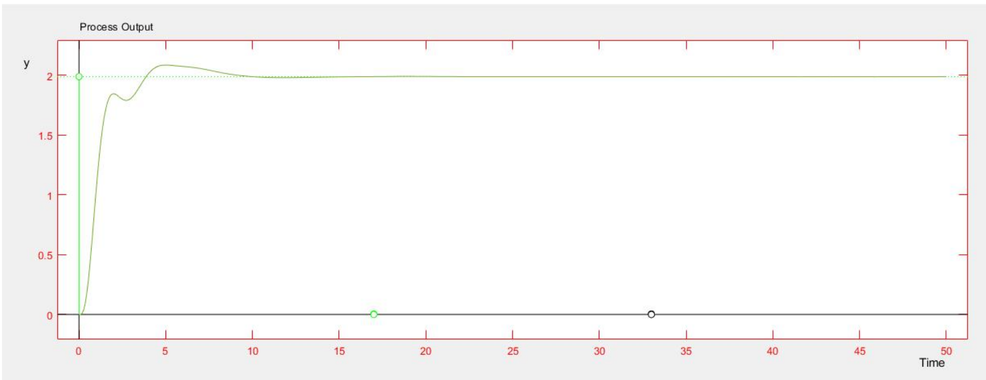

Figure 6

Figure 6 shows set point response from a PID controller simulation (in PID Basics) using K_P value of 1.0, reference value y = 2.0, K_I value of 1.0 and K_D values of 1.0[10]

With the addition of the derivative term, the phenomenon of non-intervention of PI control algorithm before the presence of error is reduced. As illustrated in Figure 6, the magnitude of overshoot has been reduced significantly. However, differentiation of a signal amplifies noise and therefore this term is highly sensitive to it, which can lead to instabilities.

3.4 PID Gain Components

As mentioned previously in this section, the three gains play important roles in PID control algorithm, this is because they determine the response of the controller and affect the stability and performance of the system[11].

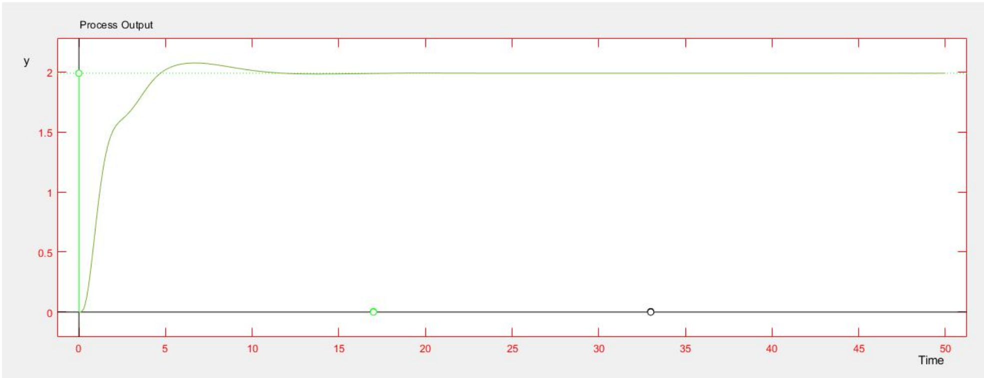

Figure 7

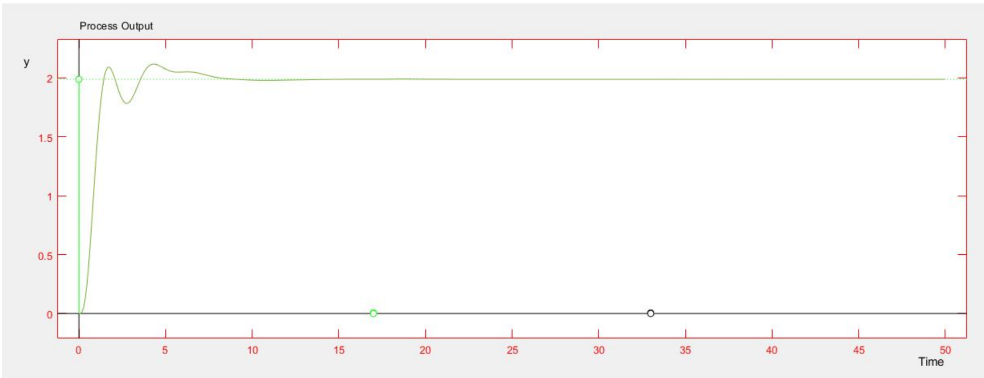

Figure 8

Figure 7 and Figure 8 show set point responses from a PID-controller simulation using K_P,K_I,K_D values of 0.7 and 1.3 respectively [10]

As shown in Figure 7 and Figure 8, where gains are 0.7 and 1.3 respectively, it is clear that the controller with a lesser value of gains has less overshoot or undershoot values. On the other hand, it also takes longer to reach the final process output compared to the controller with higher gains. Therefore, when choosing value of gains, a compromise of duration and magnitude of over-shoot and under-shoot is made.

4. Ideal and Standard Forms

The block diagram of an ideal form of PID control algorithm is shown in Figure 2, where the three terms are independent from each other[4]. The benefits of this form is that it is simple and intuitive, therefore it is commonly used in education of PID controllers. The equation(4) of the ideal form is shown below, where y(t) represents the process output.

$$y(t) = K_P * e(t) + K_I * \int_0^t e(t)\,dt + K_D * \frac{d}{dt} e(t) \tag{4}$$

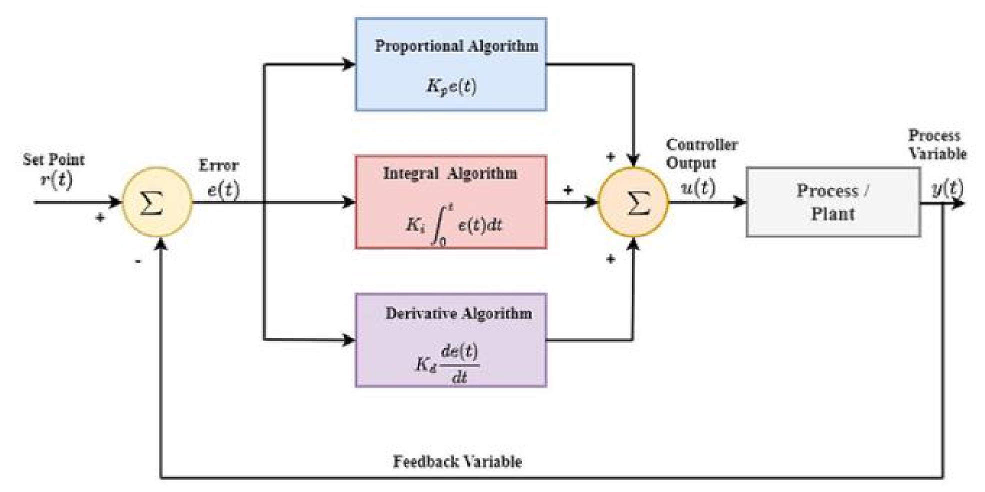

Figure 2

Figure 2 illustrates the block diagram of process control using PID[4]

However, in real-life scenarios, it could be difficult to use the ideal form of PID control algorithm. This is due to the computational load being heavy in determining the three gains of the algorithm to produce optimal results, which is also called tuning.

To reduce the computational load, a modified version is developed, which is the standard form of PID control algorithm[12]. The equation(5) of the standard form is shown below, where T_I = K_P/K_I and T_D = K_D/K_P.

$$y(t) = K_P * \left[e(t) + \frac{1}{T_I} * \int_0^t e(t)\,dt + T_D * \frac{d}{dt} e(t) \right] \tag{5}$$

As shown in the equation, the three terms are not independent, instead, the proportional gain is dominant and the main parameters of the equation change from K_P, K_I, K_D to K_P, T_I, T_D. Therefore, it is much more convenient as the operator is able to tune the controller using only one parameter, which is the proportional gain. The standard form is widely used in industry due to its convenience in tuning. The block diagram is shown in Figure 9.

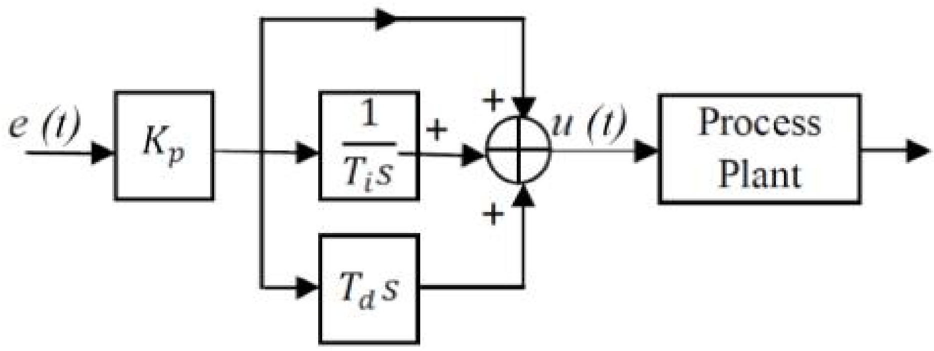

Figure 9

Figure 9 shows the block diagram for standard form of PID control algorithm [12]

5. Limitations of PID Algorithm

PID control algorithm has been used in the industry for decades, as the maintenance and operation of PID controllers are easy and they are robust in nature[3]. However, the algorithm has some limitations that could be difficult to cope with.

5.1 Tuning

As mentioned in Ideal and Standard Form, tuning is the action of determining the three gains of the PID control algorithm to produce optimal results. Classical methods for PID controller tuning found in the literature include Trial and Error Method, Ziegler–Nichols Step Response Method, Ziegler–Nichols Frequency Response Method, Relay Tuning Method and Cohen-Coon Method [4]. However, the classical tuning methods make certain assumptions about the plant and the desired output, these methods are simple to implement and have a very fast computation, but they are only good at the initial phase and they do not give the desired results all the way due to assumptions made and further tuning is needed. As tuning of the algorithm is difficult, in most of the scenarios, a compromise of being ‘sluggish’(lower gains) or having more over/undershoots(higher gains) has to be made.

5.2 Integral Term

Windup is a phenomenon found in PI and PID controllers due to the increase in the integral term when the input of the system is saturated according to the actuator limits[13]. As it is known, the actuators have physical limits, for this reason, the input of the controller must be saturated in order to avoid damage. When a PI or PID controller saturates, the integral part of the controller increases its magnitude producing performance deterioration or even instability.

5.3 Derivative Term

As mentioned in the Description of PID Control Algorithm, differentiation of a signal amplifies noise and therefore this term is highly sensitive to signal voice, which can lead to instabilities. This is mainly because the derivative term amplifies any small changes in the error function. Furthermore, derivative kick may occur due to the value of the error changes suddenly due to any adjustment made to the set point, the same phenomenon could happen to the proportional term as well. [14]. The derivative of a sudden jump in the error causes the derivative of the error to be instantaneously large and causes the controller output to saturate for one cycle at either an upper or lower bound. While this momentary jump isn't typically a problem for most systems, a sudden saturation of the controller output can put undue stress on the final control element or potentially disturb the process.

5.4 Time Delay and Ramp-Type Set-Point Change

It is well-known that PID controllers show poor control performances for a large time delay process and it cannot incorporate ramp-type set-point change or slow disturbance.[15]. PID controller cannot treat large time delay processes efficiently, the particular reason for this is that the time delay term may act as a bottleneck for the quick closed-loop reaction if it is significantly greater than the time constant of the process. For a ramp-type set-point change or disturbance, the controller shows offset and the performances of the PID controller are poor for slow set-point tracking or slow disturbance rejection.

6. Improvements to the Algorithm

Even though the limitations mentioned in previous section could be difficult to cope with, recent researches have suggested some possible methods to improve the performance of PID algorithm so that the effects of some of the above limitations could be reduced. Some examples of the possible methods have been shown below.

6.1 Intelligent tuning methods

Throughout the years, several useful findings were obtained for PID tuning methods for more performance-specific criteria and to deal with more complicated systems [4]. For example, In 1988, Fong-Chwee et al. introduced self-tuning PID controllers by pole assignment technique. Three types of self-tuning PID controllers have been discussed in it which can provide better control over dead time processes of different natures. Also, Ruano et al. have proposed a connectionist approach to PID auto-tuning which is to determine the required PID parameter values with the utilization of integral measures of the step response as the input to neural networks. One other example would be PID controller design based on genetic algorithms(GA) proposed by Liu Fan et al. [16]. The structure of the control system with GA-PID controller is shown in Figure 10. It consists of a conventional PID controller with auto-tuning its gain coefficients based on GA and a control plant.

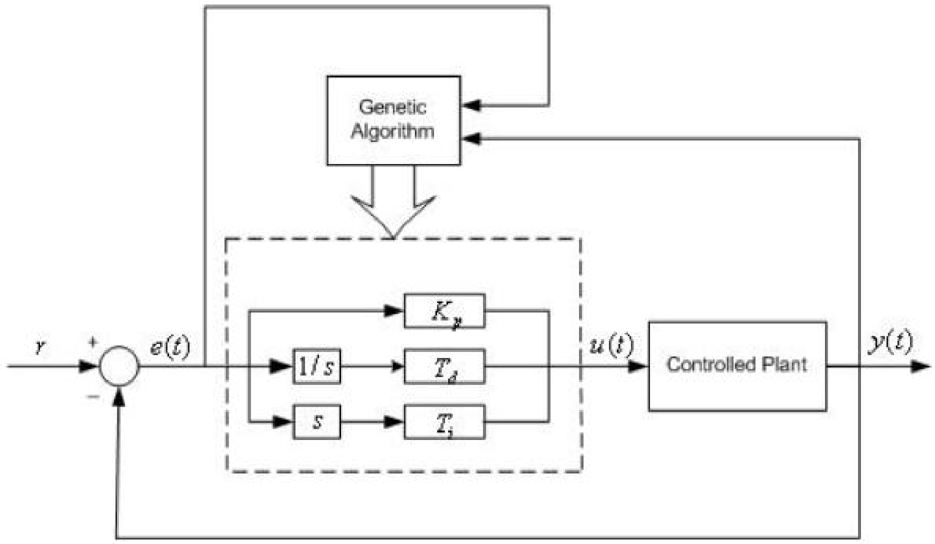

Figure 10

Figure 10 shows the block diagram for the GA-PID controller design[16].

As shown in Figure 10, an initial range of the three parameters K_P, T_D and T_I is chosen based on the traditional Z-N tuning formula, and random values within the range are chosen[16]. After the initial values are selected, the simulation runs and the error function not only goes to the initial set point like the traditional PID controllers, but also goes to the genetic algorithm where the data regarding 'sluggish' or over/undershoots is recorded and processed. After the processing of data in the genetic algorithm is done, modifications to the three parameters are made, and the loop continues until the data converges to a certain level. As a result, the optimal gains are obtained.

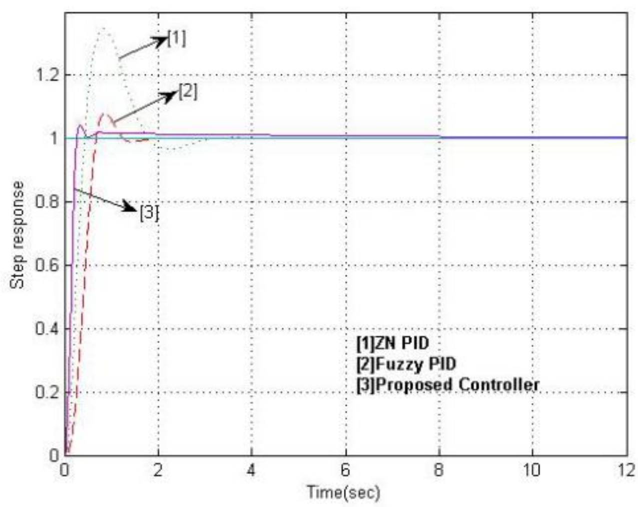

Figure 11

Figure 11 shows the step response simulation of ZN-PID controller, Fuzzy PID controller and the proposed GA-PID controller[16]

As illustrated in Figure 11, an example of step response simulation of ZN-PID controller, Fuzzy controller and the proposed GA-PID controller respectively. It is clear from the figure that ZN-PID controller gives both more 'sluggish' behaviour and more over/undershoots, while the proposed GA-PID controller has shown both the least 'sluggish' behaviour and least over/undershoots[16].

6.2 Anti Windup PID Controllers

One possible method of reducing the effects of windup is developed by implementing an internal model controller (IMC) [13]. Internal model control is a technique that consists in designing an appropriate controller according to the internal stability of the system, therefore, this control strategy is convenient for the design of an anti-windup control architecture, reducing the unwanted effects yielded by this phenomenon and improving the system performance. The anti-windup control strategy mentioned is developed by feedback the saturated input to the internal model controller so the effects of windup are minimized. The structure of the anti-windup controller is shown in Figure 12, where G_c(S) refers to the internal model PID controller, R(s) is the anti-windup compensator filter and G_p(s) is the plant transfer function(similar to process plant). As a result, the unwanted effects of windup are suppressed with the addition of a feedback loop which includes the saturated input signal through a filter that improves the system performance when the input is saturated and windup occurs in the PID controller.

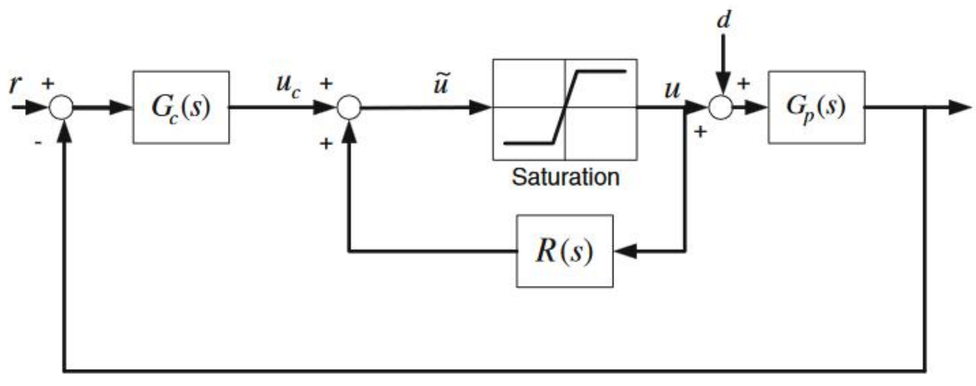

Figure 12

Figure 12 shows the block diagram of the anti windup controller[13].

6.3 PID Algorithms with No Proportional and Derivative Kick

As mentioned, proportional and derivative kick occurs due to the large error present as adjustment is made to the set point, an obvious solution is to not let the control action have the kick at the time of the set point changes. This calls for a PID form with no proportional and derivative kick, which simply means having the proportional and derivative mode acting on the negative sign of the controlled variable alone[17]. The original formula for PID algorithm is shown in equation(5), while the modified equation(6) is shown below.

$$y(t) = K_P * \left[- e(t) + \frac{1}{T_I} * \int_0^t e(t)\, dt - T_D * \frac{d}{dt} e(t) \right] \quad (6)$$

It is clear that as the proportional and derivative terms are changed to negative, when a change is being made to the set point, the positive integral and the two negative terms would offset each other thus giving a much smaller error value compared to the traditional PID algorithm[17]. Also, note that this PID form can be found in almost every industrial distributed control systems, thus no special control implementation is needed.

7. Conclusion

In this paper, a summary of PID controllers has been presented. A simple simulation of PID controllers using PID Basics[10] is carried out to demonstrate the function of each term in the PID controllers. PID controllers are commonly used, simple, and robust but also contain a series of limitations. Although corresponding improvements have been developed in recent years, a generally high-performance PID controller which has almost no defect is still not present.

8. Reference

[1]: https://www.controleng.com/articles/know-when-to-use-open-or-closed-loop-control/
[2]: https://www.pesquality.com/blog/open-loop-and-closed-loop-systems
[3]: Panda, Rames C., Cheng-Ching Yu, and Hsiao-Ping Huang. "PID tuning rules for SOPDT systems: Review and some new results." *ISA transactions* 43.2 (2004): 283-295.
[4]: Borase, Rakesh P., et al. "A review of PID control, tuning methods and applications." *International Journal of Dynamics and Control* 9 (2021): 818-827.
[5]: Giuseppe, Barone., Giovanni, Brusco., Alessandro, Burgio., Daniele, Menniti., Anna, Pinnarelli., M., Motta., Nicola, Sorrentino., Pasquale, Vizza. "A Real-Life Application of a Smart User Network." Energies, undefined (2018). doi: 10.3390/EN11123504
[6]: William, Montalvo. "Metaheuristics of the Artificial Bee Colony Used for Optimizing a PID Dahlin in Arm Platform." undefined (2023). doi: 10.1007/978-3-031-30592-4_18
[7]: Petitgirard, Loïc. "The Mathematics of Nonlinear Oscillations in the 1920s: A Decade of Trials and Convergence? Examples of the Work of Nicolai Minorsky." *Mathematical Communities in the Reconstruction After the Great War 1918–1928: Trajectories and Institutions* (2021): 227-251.
[8]: Ho, A.K.H. 2014. Fundamental of PID Control. [Online]. [17 February 2020]]. Available from: https://pdhonline.com/courses/e331/e331content.pdf
[9]: Bennett, Stuart. "Nicholas Minorsky and the automatic steering of ships." *IEEE Control Systems Magazine* 4.4 (1984): 10-15.
[10]: Lucas García (2023). PID Basics (https://www.mathworks.com/matlabcentral/fileexchange/37962-pid-basics), MATLAB Central File Exchange. Retrieved October 25, 2023.
[11]: Rajagopal, Veramalla., Sabha, Raj, Arya., Vishwas, Gundeboina., Bangarraju, Jampana., Rajasekharareddy, Chilipi., Santhosh, Madasthu. "Meta-heuristics algorithms for optimization of gains for dynamic voltage restorers to improve power quality and dynamics." Optimal Control Applications & Methods, undefined (2022). doi: 10.1002/oca.2881

[12]:Marzoughi, Ali & Selamat, Hazlina & Fua, Mohd & Rahim, Herlina. (2023). Optimized proportional integral derivative (PID) controller for the exhaust temperature control of a gas turbine system using particle swarm optimization.
[13]: Azar, Ahmad Taher, and Fernando E. Serrano. "Design and modeling of anti wind up PID controllers." *Complex system modelling and control through intelligent soft computations*. Cham: Springer International Publishing, 2014. 1-44.
[14]: https://apmonitor.com/pdc/index.php/Main/ProportionalIntegralDerivative
[15]: Sung, Su W., et al. "Limitations and countermeasures of pid controllers". Industrial &Amp; Engineering Chemistry Research, vol. 35, no. 8, 1996, p. 2596-2610. https://doi.org/10.1021/ie960090+
[16]: Fan, Liu, and Er Meng Joo. "Design for auto-tuning PID controller based on genetic algorithms." *2009 4th IEEE conference on industrial electronics and applications*. IEEE, 2009.
[17]: Chien, I-Lung, et al. “A Simple Multiloop Tuning Method for PID Controllers With No Proportional Kick.” Industrial & Engineering Chemistry Research, vol. 38, no. 4, American Chemical Society, Mar. 1999, pp. 1456–68. https://doi.org/10.1021/ie980595v.