

---

# Ziegler–Nichols Tuning Method\*

Understanding the PID Controller

---

*Vishakha Vijay Patel*

The objective of this article is to study the speed control of a DC motor using PID controller and understand the Ziegler–Nichols (ZN) tuning method for a PID controller. PID controllers are widely used in many industries such as paper mill, cotton textile industries. PID controller also finds its applications in drones. The usefulness of PID lies in its general applicability.

## 1. Introduction

Speed control of DC motors finds applications in various industries—cotton mills to space applications. In various applications, due to fluctuations in load, the speed of the motor varies. DC motor has desirable characteristics, such as high starting torque and a wide range of speed control, due to which it is used in many industries. A PID controller is used widely in the speed control of a motor—over half the industries use PID controllers or their modified forms. It is required to control the motor speed as per need to ensure higher efficiency and optimum performance. The usefulness of PID is due to its general applicability to various control systems. They are feasible, easy to design, and implement. They combine the advantage of proportional, derivative, and integral control actions.

## 2. Mathematical Modelling of DC Motor

Transfer function is defined as the ratio of Laplace transform of the output variable to Laplace transform of the input variable, assuming all initial conditions to be zero.



Vishakha Patel has completed her BE Electrical and Electronics Engineering from the CMR Institute of Technology, Bangalore. Her work interests are in the domain of control systems, especially controls in aeronautical and biomedical fields.

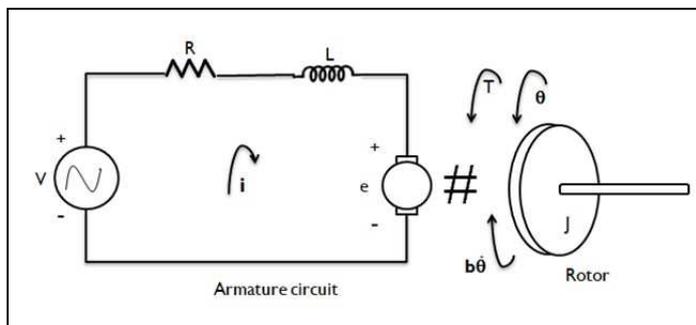
### Keywords

D C Motor, PID controller, Ziegler–Nichols technique, MATLAB.

---

\*Vol.25, No.10, DOI: <https://doi.org/10.1007/s12045-020-1058-z>

**Figure 1.** DC motor diagram depicting armature circuit as a KVL loop and rotor as the free body diagram.



Moment of inertia of rotor	$J = 0.02 \text{ Kg} - \text{m}^2/\text{s}^2$
Damping co-efficient of mechanical system	$b = 0.2 \text{Ns/m}$
Electromotive force and torque constant	$K_t = K_e = K = 0.1$
Armature resistance $R$	$R = 2\Omega$
Armature inductance $L$	$L = 0.5 \text{H}$
Input	Voltage source $V$ (volts)
Output	Speed at which motor runs $\dot{\theta}$ (rps)

**Table 1.** The physical parameters considered for the analysis of DC motor.

## 2.1 Transfer Function Modeling

Consider a DC motor as shown in *Figure 1*. The rotor is mechanically coupled, and its free body diagram is shown. The armature is represented as an electrical circuit.

The input voltage  $V$  is applied to the field winding which has a resistance and inductance of  $R$  and  $L$  respectively. The armature is supplied with a constant armature current, and thus the motor shaft is controlled by the input voltage. The field current produces a flux in the machine, which in turn produces a torque at the motor shaft [1].

The physical parameters considered for the analysis are as shown in *Table 1* [4]. The symbols used in the article are listed in *Table 2*.

In a DC motor, the torque  $T$  is directly proportional to the arma-



---

Current	$i$ (Ampere)
Angular shaft velocity	$\dot{\theta}$ (Revolutions per second – rps)
Proportional gain constant	$K_p$
Integral time constant	$T_i$ (Second)
Derivative time constant	$T_d$ (Second)
Integral gain constant	$K_i$
Derivative gain constant	$K_d$

**Table 2.** Symbols used in the article.

ture current  $i$

$$T \propto i .$$

$$\text{Therefore, } T = K_t i , \quad (1)$$

And the back emf  $e$  is directly proportional to the angular shaft velocity  $\dot{\theta}$

$$e \propto \dot{\theta} .$$

$$\text{Therefore, } e = K_b \dot{\theta} . \quad (2)$$

Applying Newton's law to the rotor system

$$J\ddot{\theta} = T - b\dot{\theta} .$$

Substitute for  $T$  from (1)

$$\Rightarrow J\ddot{\theta} = K_t i - b\dot{\theta} . \quad (3)$$

Applying KVL to the armature circuit

$$V - e = L \frac{di}{dt} + Ri .$$

Substitute for  $e$  from (2)

$$\Rightarrow V - K_b \dot{\theta} = L \frac{di}{dt} + Ri . \quad (4)$$



---

Performing the Laplace transformation on (3)

$$sJ\dot{\theta}(s) = KI(s) - B\dot{\theta}(s),$$

$$\Rightarrow I(s) = \left( \frac{\dot{\theta}(s)(sJ + B)}{K} \right). \quad (5)$$

Performing the Laplace transformation on (4)

$$V(s) - K\dot{\theta}(s) = [sL + R] I(s). \quad (6)$$

Substitute for  $I(s)$  from (5) in (6)

$$V(s) - K\dot{\theta}(s) = (sL + R) \frac{(sJ + B)}{K} \dot{\theta}(s),$$

$$\Rightarrow V(s) = \dot{\theta}(s) \left( \frac{(sL + R)(sJ + B)}{K} + K \right), .$$

Thus, the transfer function  $G(s)$  can be written as

$$\text{Therefore, } G(s) = \frac{\dot{\theta}(s)}{V(s)} = \frac{K}{(sL + R)(sJ + B) + K^2}. \quad (7)$$

## 2.2 State Space Modeling

Consider the DC motor and define its state variables as

$$x_1 = i \quad \text{and} \quad x_2 = \dot{\theta}$$

Rewriting equations (3) and (4)

$$\dot{x}_1 = \frac{1}{L}(-Rx_1 - Kx_2 - V),$$

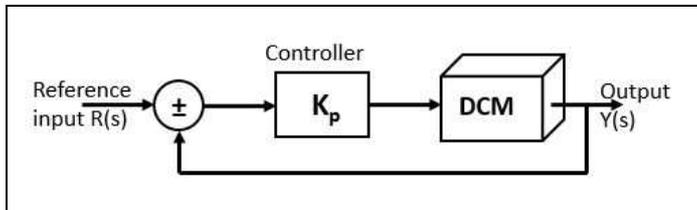
$$\dot{x}_2 = \frac{1}{J}(Kx_1 - bx_2). \quad (8)$$

Let the output to be the angular speed of shaft  $\dot{\theta}$

$$y = x_2 .$$

Writing the above equations in matrix form of state space

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} \frac{-R}{L} & \frac{-K}{L} \\ \frac{K}{J} & \frac{-b}{J} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} \frac{1}{L} \\ 0 \end{bmatrix} \begin{bmatrix} V \\ 0 \end{bmatrix} [y] = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (9)$$



**Figure 2.** Proportional controller in a feedback control loop.

### 3. PID Controller

#### 3.1 Proportional Controller

In a proportional (P) (*Figure 2*) controller, the control output is proportional to the error signal. The error signal is defined as the difference between the output (or feedback signal) and the reference signal. Proportional controller amplifies the control signal. The controller will push harder for a given error, and the closed-loop system will react quickly with an increase in overshoot. P controller does reduce the steady-state error, but does not eliminate it. Changing the control parameter  $K_p$  can change the closed-loop dynamics.

#### 3.2 Integral Controller

The integral controller (*Figure 3*) acts on the history of response error. The control signal at any instant is the area under the actuating-error-signal curve up to that instant. This leads to the conclusion that even when the error signal has become zero, the control signal  $u$  will have a non-zero value. The control signal will consist of the addition of the area under the error signal before it became zero [1].

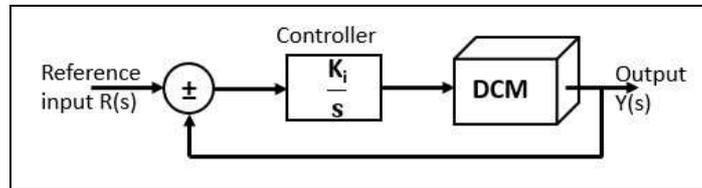
Hence, the integral controller is used to eliminate the offset error. When there is a persistent (constant) steady-state error, the integrator builds and builds, thereby increasing the control signal and driving the error down.

The control signal increases continuously. The integral controller

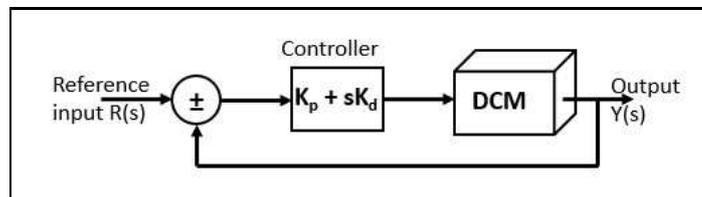
The integral controller is used to eliminate the offset error. When there is a persistent (constant) steady-state error, the integrator builds and builds, thereby increasing the control signal and driving the error down.



**Figure 3.** Integral controller in a feedback control loop.



**Figure 4.** Proportional derivative controller in a feedback control loop.



increases the overshoot, and also makes the system sluggish, as the error signal changes sign, it takes a while for the control signal to change path. It is unable to eliminate the oscillations from the system.

The integral controller acts on the history of response error. The control signal at any instant is the area under the actuating-error-signal curve up to that instant.

### 3.3 Derivative Controller

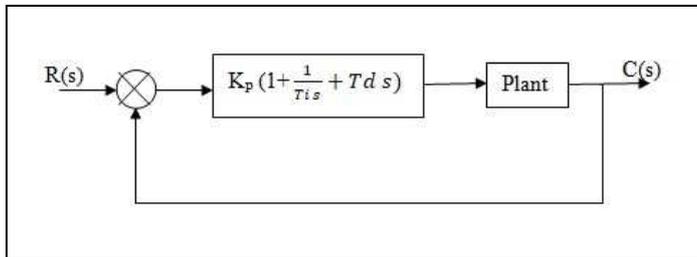
The derivative (D) controller (*Figure 4*), as the name suggests acts on the rate of change of error signal. The controller predicts how the error will be (future) and acts accordingly in the present. Prevention is better than cure.

The controller takes action even before the error has become too big or severe. Because the derivative controller operates on the rate of change of error, and not on the error itself, D controller is never used alone. It is always used in combination with P or PI controller.

Derivative control thus anticipates the actuating error, initiates an early corrective action and tends to increase the stability of the system [1]. Derivative controller adds on to the damping, which

Derivative control anticipates the actuating error, initiates an early corrective action and tends to increase the stability of the system.





**Figure 5.** PID controller in a feedback control loop. The PID controller's mathematical equation can also be written as  $K_p + \frac{K_i}{s} + sK_d$ .

Close-Loop Response	Rise time	Overshoot	Settling time	Steady state error
P controller	↓	↑	~	↓
I controller	↓	↑	↑	↓
D controller	~	↓	↓	~

**Table 3.** Comparative study of the characteristics of a step response for a second order system for P, I, and D controller.

results in the decrease in overshoot, and hence decrease in the settling time.

### 3.4 PID Controller

PID controllers (*Figure 5*) are most popular due to their general applicability in various control domains. It is used for its zero steady error, fast response, no oscillations, high stability, and elimination of overshoot.

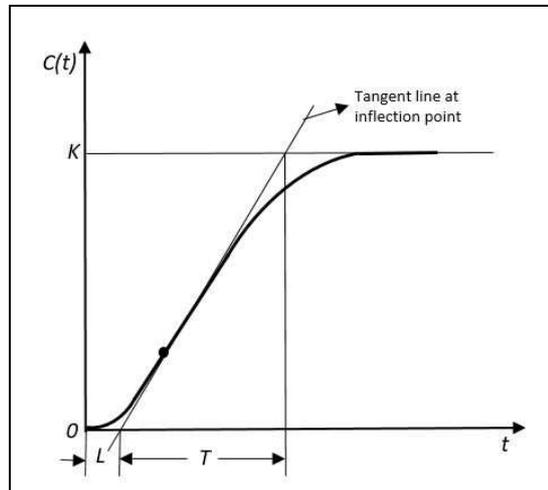
From the understanding of the different controllers, one can conclude and compare the P controller, I controller and D controller as shown in *Table 3*.

## 4. Ziegler–Nichols First Tuning Method

Ziegler–Nichols (ZN) rules are widely used to tune PID controllers for which the plant dynamics are precisely not known, it can also be applied to plants of known dynamics. Ziegler and Nichols proposed rules for determining values of proportional gain  $K_p$ , integral time  $T_i$ , and derivative time  $T_d$  based on the transient response characteristic of a given plant [1].



**Figure 6.** S-shaped step response of a plant. The tangent to the curve is drawn at the point of inflection, to calculate the delay time  $L$  and time constant  $T$ .



Type of Controller	$K_p$	$T_i$	$T_d$
<b>P</b>	$\frac{T}{L}$	$\infty$	0
<b>PI</b>	$0.9\frac{T}{L}$	$\frac{L}{0.3}$	0
<b>PID</b>	$1.2\frac{T}{L}$	$2L$	$0.5L$

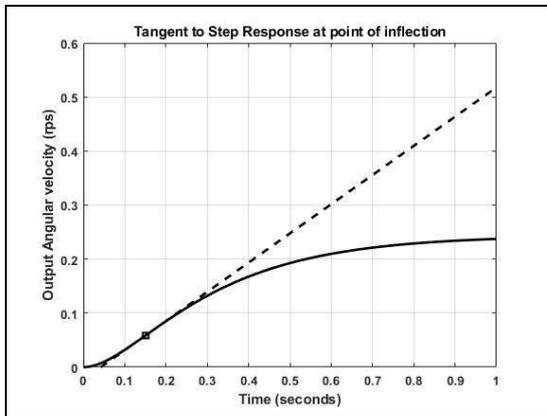
**Table 4.** Gain parameter calculation for ZN tuning method 1, according to the values of  $L$  and  $T$  obtained.

PID controller is defined as  $K_p(1 + \frac{1}{T_i s} + T_d s)$ .

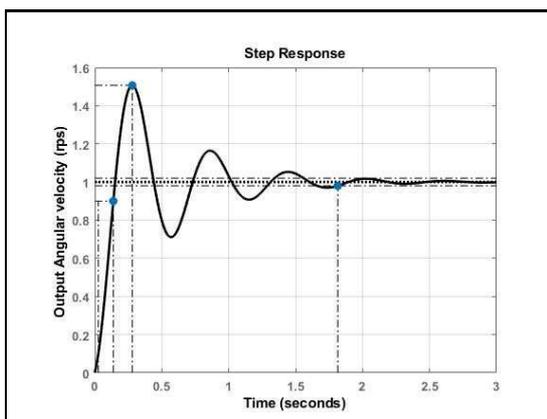
The first tuning method is applied for plants that show an S-shaped characteristic for their step response. The S-shaped curve is characterized by two constants—the delay time,  $L$  and time constant,  $T$ . A tangent is drawn to the S-shaped curve at the point of inflection (*Figure 6*). The delay time,  $L$  is determined by the intersection of the tangent line with the time axis and time constant,  $T$  is determined by intersection of tangent line with final value of step response  $c(t)=K$ . Ziegler and Nichols suggested setting the values of the parameters  $K_p$ ,  $T_i$ , and  $T_d$  according to *Table 4*.

The real usefulness of ZN tuning method is seen when the plant dynamics are unknown. The main advantage provided by ZN tun-





**Figure 7.** The step response of the DCM plant is obtained as an S-shaped curve. The code to draw the tangent at the point of inflection is build and executed in MATLAB.



**Figure 8.** Step response plotted for the DCM with PID parameters as calculated from *Table 4*, according to Ziegler–Nichols first tuning method.

ing rules is that they provide a starting point for the determination of the parameter values. Once we have obtained these parameter values ( $K_p$ ,  $T_i$ , and  $T_d$ ), we can always further fine-tune the plant's closed loop response by changing the values, so as to obtain a satisfactory transient response. On an average, 25% overshoot is obtained when tuning by the ZN method.

MATLAB code to draw the tangent line to the S-shaped curve (*Figure 7*) and also obtain the parameters  $L$  and  $T$  is given in the *Appendix*. Based on *Table 4*, the PID parameters are calculated, and the step response is plotted (*Figure 8*).

Observe that the step response is oscillatory in the time domain.

The real usefulness of ZN tuning method is seen when the plant dynamics are unknown. The main advantage provided by ZN tuning rules is that they provide a starting point for the determination of the parameter values.



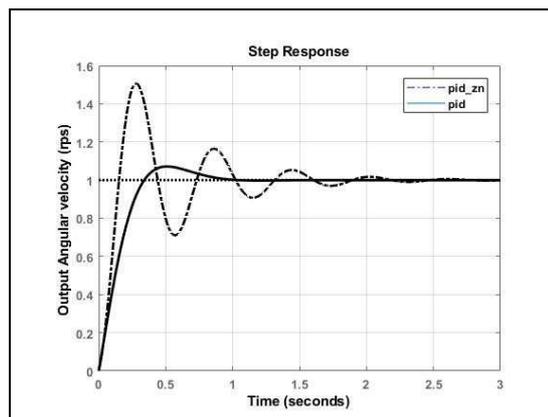
**Table 5.** Gain parameter values as found from Ziegler–Nichols tuning method, and then fine tuning those parameters for a better response.

Parameters values	ZN first method	Fine tuning
Proportional Gain $K_p$	12.919	8.346
Integral Gain $K_i$	154.168	36.94
Derivative Gain $K_d$	0.27	0.399

**Table 6.** Characteristics from Ziegler–Nichols tuning method, and then fine tuning those parameters for a better response.

Characteristics	ZN method	Fine-tuning
Rise Time	0.112 seconds	0.243 seconds
Overshoot	50.7%	7.04%
Settling Time	1.81 seconds	0.817 seconds

**Figure 9.** Step response plotted for the DCM with ZN tuning method and the tuning method for PID controller. The fine-tuned curve (pid) is less oscillatory in the time domain giving a more desirable response.

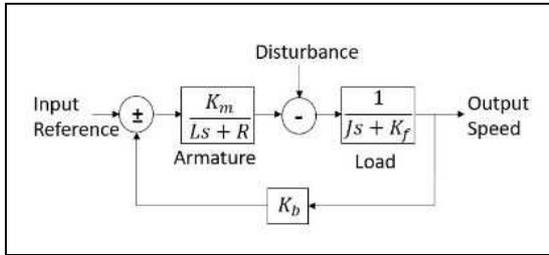


To further tune this, the MATLAB tool is used. Utilize the obtained parameter values as the initial values, and then tune each parameter till the desired result is obtained. Comparison of the PID parameters and the characteristic of the step responses is shown in *Table 5* and *Table 6*.

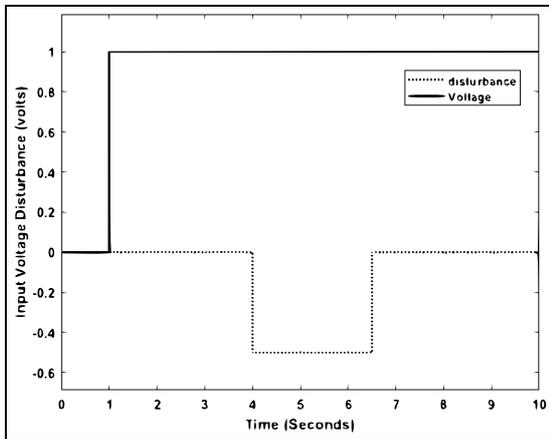
Fine-tuning the parameters obtained provides a better result. The oscillation's, maximum overshoot is reduced, and the characteristics are practical values that can be implemented.

However, ZN tuning rules can only be applied to a plant exhibiting S-shaped behavior for a unit step unit or exhibiting sustained oscillation for any gain  $K$ . In some cases, for a plant having an integrator, the ZN rules may not be applicable as it won't satisfy

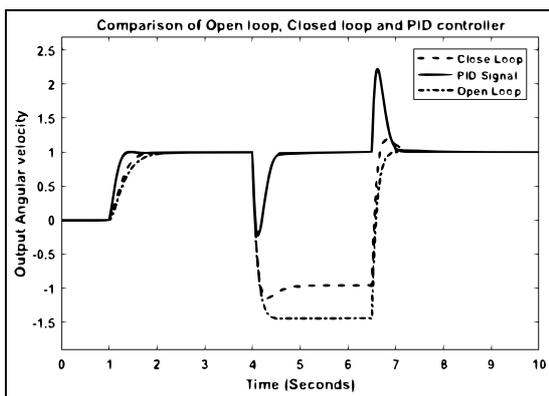




**Figure 10.** Load disturbance applied to a DC motor plant. For further understanding of load disturbance, refer MATLAB documentation.



**Figure 11.** Unit step input provided for a duration of 9 seconds with a load disturbance of  $-0.5$  amplitude for 2.5 seconds.



**Figure 12.** Comparison of the behavior of controllers in the presence of disturbance. Open-loop and closed-loop controller is the most affected due to disturbance. PID minimizes the fluctuation in speed due to disturbance.

the above conditions.

On comparing this design of PID controller with an open-loop



---

One must understand that controller design depends entirely on the plant modelling and users requirement. Here, the PID controller works better providing a satisfactory response.

and closed-loop response, the PID controller is seen to be better suited (*Figure 9*). It prevents the change in speed better in the presence of a load disturbance. One must understand that controller design depends entirely on the plant modelling and users requirement. Here, the PID controller works better providing a satisfactory response.

To understand the performance of the system under an applied disturbance, a model (*Figure 10*) is implemented and studied. Load disturbance, as described in *Figure 11*, is applied, and the responses of open-loop, closed-loop, and PID controlled systems are compared (*Figure 12*).

## 5. Conclusion

The individual working of proportional, integral, and derivative controller part in a PID controller is attempted to understand through this paper. PID controller can be used to deal with higher-order system and tuning with the Ziegler–Nichols method, which provides an educated guess to further tune the PID parameters. The Ziegler Nicholas method gives 10% to 60% overshoot and can be used when dealing with non-complex dynamic systems exhibiting S-shaped step response or continuous oscillating response. Systems exhibiting a continuous oscillating response require tuning by the second ZN method. Based on the different industrial applications of the DC motor, the controller should be chosen and appropriately tuned.

## Suggested Reading

Address for Correspondence  
Vishakha Patel  
Villa 12, Neeraja Halcyon  
Horamavu  
Bangalore 560 043, India.  
Email:  
vishupatel197@gmail.com

- [1] K Ogata, *Discrete-Time Control Systems*, University of Minnesota, Prentice Hall, 1987.
- [2] N Adhikari, M Choubey and R Singh, DC motor control using Ziegler–Nichols and genetic algorithm technique, *International Journal of Electrical, Electronics and Computer Engineering*, April 2012.
- [3] Abdulameer M Sulaiman, MSM Aras and D Saleem, Tuning methods of PID controller for DC motor speed control, *Indonesia Journal of Electrical Engineering and Computer Science*, Vol.3 No.2, Aug 2016.



---

[4] MATLAB help document “DC motor control”

<https://in.mathworks.com/help/control/examples/dc-motor-control.html>

<https://in.mathworks.com/help/control/examples/dc-motor-control.html>

### Box 1. Appendix: MATLAB Code

```
% ZN first method of tuning
K=0.1;R=2;L=0.5;J=0.02;b=0.2;
num=[K];
den=[(L*J) ((L*b)+(R*J)) (R*b)+K^2];
G=tf(num,den) %Plant Transfer function
[Y t]=step(G); % Plant step response
M=gradient(Y,t); % Gradient/slope at each point of step response
[m i]=max(M);% Max slope (at point of inflection)
c=Y(i)-(m*t(i));
figure(7);plot(t,Y,t,(m*t)+c,'--',t(i),Y(i),'s');
grid on;axis([0 1.2 -0.2 0.6]);
L=-c/m % Delay constant
T=((Y(end)-c)/m)-L % Time constant
%Construct PID Controller
Kp=1.2*(T/L);
Ti=2*L;
Td=0.5*L;
s=tf('s');
PIDc=Kp*(1+(1/(Ti*s))+Td*s);% PID controller
sys_pid=feedback(PIDc*G,1);% Closed loop system
figure(8);step(sys_pid);grid on;
%Display PID parameters Kp, Ki and Kd
Kp
Ki=Kp/Ti
Kd=Kp*Td [2]
```

