Tuning Methods of PID Controller

Dr. Ir. Yeffry Handoko Putra, M.T

Session Outlines & Objectives

Outlines

- Tuning methods of PID controller:
  - Ziegler-Nichols Open-loop
  - Coon-Cohen Open-loop
  - Ziegler-Nichols Closed-loop
  - Lambda Tuning
  - Visual Loop Tuning
  - Autotuning

Objectives

- Know the meaning of controller tuning
- Be able to use several PID tuning methods and choose the right tuning methods for specific process control application

Introduction (1)

Controller tuning

- A systematic-adjusting procedure of the controller parameters to obtain a desired performance of the control system

PID control tuning

- It is a matter of selecting the right mix of P, I, and D action to achieve a desired performance

Introduction (2)

Performance criteria for closed-loop systems

- Stable
- Minimal effect of disturbance
- Rapid, smooth response to set point change
- No offset
- No excessive control action
- Robust to plant-model mismatch

Trade-offs in control problems

- Set point tracking vs. disturbance rejection
- Robustness vs. performance

Introduction (3)

How do we know when it’s tuned?

- The process didn’t blow up ☺
- The process measurements stay close enough to the setpoint
- Boss says OK, and you can go home
- You buy a new controller which has different PID algorithm

Introduction (4)

The Problem

- We have the knowledge about the effect of each PID modes to closed-loop response
- But, from what values of P, I and D modes we would pick to start to tune?

The Solutions

- If you have tuned the process before, use **slightly different** values of the old PID controller parameter
- If the results are still not satisfy you, use a PID controller tuning method that we will learn just in a moment that is most suit to your process control application. Keep watch on …

Introduction (5)

General Tuning Procedure

- Before tuning, **FAMILIARIZE** with the **OPERATION RISK**
- Get help with experienced operators, explain your work to him and tell him that **NO PERMISSION IS REQUIRED** if their intervention is **NECESSARY** to save the loop if things go wrong

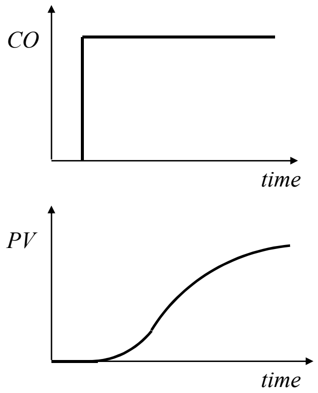
Introduction (6)

Precaution:

All kinds of tuning method should be used for initial setting and fine tuning should be done!!!

Cohen-Coon Open-loop Tuning Method (1)

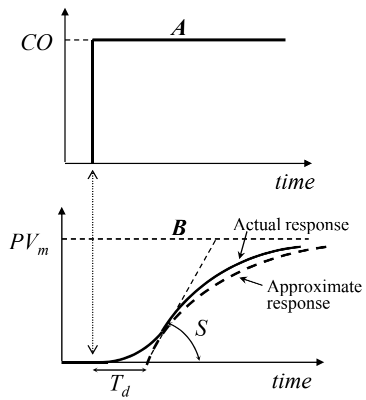
- Proposed in 1953 by G. H. Cohen and G. A. Coon[1]
- Main principles:
  - The process output is affected **not** only by the dynamics of the main process but also by the dynamics of the measuring sensor and final control element
  - They observed that the response of most processing unit to an input change had a **sigmoidal** shape



[1] G. H. Cohen and G. A. Coon, *Theoretical Consideration of Retarded Control*, Trans. ASME,Vol. 75, pp. 827, 1953.

Cohen-Coon Open-loop Tuning Method (2)

- Main principles: (contd.)
  - The sigmoidal shape can be **adequately approximated** by the response of a **first order system with dead time**



$$G_{fpm} = \frac{PV_m}{CO} \simeq \frac{K\, e^{-T_d s}}{\tau s + 1},$$

where

$$K = \frac{B}{A}$$

$$\tau = \frac{B}{S}$$, S is the slope of the sigmoidal response at the point of inflection

T_d = time elapsed until the system responded

Cohen-Coon Open-loop Tuning Method (3)

- Once the value of process parameter are obtained, the PID parameter can be calculated from the following table

Controller	P	I_m	D
P only	$\frac{1}{K}\frac{\tau}{T_d}\left(1+\frac{T_d}{3\tau}\right)$	-	-
PI	$\frac{1}{K}\frac{\tau}{T_d}\left(0.9+\frac{T_d}{12\tau}\right)$	$T_d\frac{30+3T_d/\tau}{9+20T_d/\tau}$	-
PID	$\frac{1}{K}\frac{\tau}{T_d}\left(\frac{4}{3}+\frac{T_d}{4\tau}\right)$	$T_d\frac{32+6T_d/\tau}{13+8T_d/\tau}$	$T_d\frac{4}{11+2T_d/\tau}$

Ziegler-Nichols Open-loop Tuning Method (1)

- Proposed in 1942 by J. G. Ziegler and N. B. Nichols of Taylor Instruments (now part of ABB instrumentation in Rochester, N.Y.) [2]
- It is done in **manual** mode
- It is a way of relating the process parameters (i.e. delay time, process gain and time constant) to the controller parameters (i.e. controller gain and reset time)
- It has been developed for use on **delay-followed-by-first-order-lag** processes

[2] J. G. Ziegler and N. B. Nichols, *Optimum Setting for Automatic Controllers*, Trans. ASME,Vol. 64, pp. 759-768, 1942.

Ziegler-Nichols Open-loop Tuning Method (2)

The Procedure

1. Put the control system in **MANUAL** (without feedback)
2. Adjust the controlled system manually to the desired operating point (start-up control loop)
3. Apply manually a **STEP** change of the controller output (CO) (usually 5 – 10 % or depending of your process gain)
4. Wait until the process variable (PV) is settled at steady-state condition

Ziegler-Nichols Open-loop Tuning Method (3)

5. Determine process parameter (delay time, process gain and time constant) from the graphics

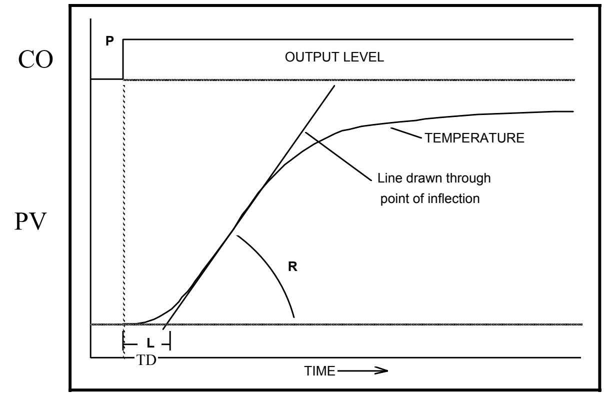


FIGURE 6: PROCESS REACTION CURVE - SIMPLE

Ziegler-Nichols Open-loop Tuning Method (4)

6. Once the value of process parameter are obtained, the PID parameter can be calculated from the following table

Controller	*P*	I_m	*D*
P only	$\frac{1}{K}\left(\frac{\tau}{T_d}\right)$	-	-
PI	$\frac{0.9}{K}\left(\frac{\tau}{T_d}\right)$	$0.33\ T_d$	-
PID	$\frac{1.2}{K}\left(\frac{\tau}{T_d}\right)$	$2\ T_d$	$0.5\ T_d$

Ziegler-Nichols Closed-loop Tuning Method (1)

- Proposed in 1942 by J. G. Ziegler and N. B. Nichols of Taylor Instruments (now part of ABB instrumentation in Rochester, N.Y.)
- Also known as **continuous cycling** or **ultimate gain** methods
- It has been developed for use on **delay-followed-by-first-order-lag** processes
- It has been refined for other specific process control objectives

Ziegler-Nichols Closed-loop Tuning Method (2)

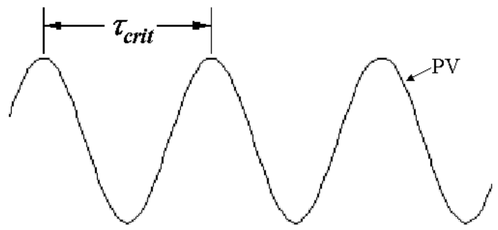
The Procedure

1. At the controller, select proportional-only (**P-ONLY**) control, i.e. set ***P*** to the lowest value (***PB*** to the highest value) and I_m to infinity (I_r to zero) and ***D*** to zero (smallest possible influence of the controller)
2. Adjust the controlled system manually to the desired operating point (start-up control loop)
3. Set the manipulated variable of the controller to the manually adjusted value (reset bias ***b***) and switch to automatic operating mode
4. Continue to gradually increase ***P*** (decrease ***PB***) until the controlled variable encounters harmonic oscillation. If possible, small step changes in the setpoint should be made during the ***P*** adjustment to cause the control loop to oscillate
5. Take down the adjusted ***P*** value as critical proportional-action coefficient P_{crit}

Ziegler-Nichols Closed-loop Tuning Method (3)

The Procedure (contd.)

6. Determine the time span for one full oscillation amplitude as t_{crit}, if necessary by taking the time of several oscillations and calculating their average



Ziegler-Nichols Closed-loop Tuning Method (4)

The Procedure (contd.)

7. Once the value for P_{crit} and τ_{crit} are obtained, the PID parameter can be calculated from the following table

Controller	*P*	I_m	*D*
P only	0.5 P_{crit}	-	-
PI	0.45 P_{crit}	0.833 τ_{crit}	-
PID	0.6 P_{crit}	0.5 τ_{crit}	0.125 τ_{crit}

Ziegler-Nichols Closed-loop Tuning Method (5)

Modified Ziegler-Nichols setting

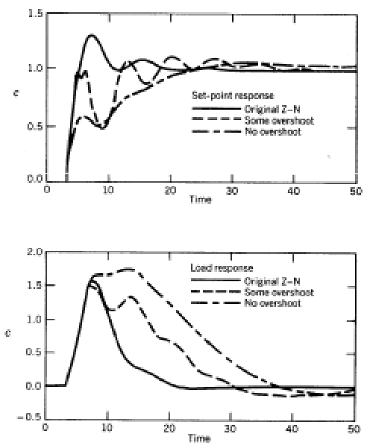
Controller	*P*	I_m	*D*
PID original	0.6 P_{crit}	0.5 τ_{crit}	0.125 τ_{crit}
PID some overshoot	0.33 P_{crit}	0.5 τ_{crit}	0.33 τ_{crit}
PID no overshoot	0.2 P_{crit}	0.3 τ_{crit}	0.5 τ_{crit}

Ziegler-Nichols Closed-loop Tuning Method (6)

❑ **Examples**

$$G_p(s) = \frac{4e^{-3.5s}}{7s+1} \quad \left.\begin{matrix} P_c = 0.95 \\ \tau_c = 12 \end{matrix}\right.$$

Controller	P	I_m	D
PID original	**0.57**	**6.0**	**1.5**
PID some overshoot	**0.31**	**6.0**	**0.4**
PID no overshoot	**0.19**	**6.0**	**4.0**



Ziegler-Nichols Closed-loop Tuning Method (6)

Advantages of continuous cycling method

- ❑ No a priori information on process required
- ❑ Applicable to all stable processes
- ❑ Only a single experimental test is needed
- ❑ It does not require trial and error
- ❑ The controller settings are easily calculated

Ziegler-Nichols Closed-loop Tuning Method (7)

Disadvantages of continuous cycling method

- ❑ Time consuming
- ❑ Loss of product quality and productivity during the tests
- ❑ Continuous cycling may cause the violation of process limitation and safety hazards
- ❑ Not applicable to open-loop unstable process
- ❑ First-order and second-order process without time delay will not oscillate even with very large controller gain

→ Motivates Relay Feedback Method (Åström and Hagglund, 1984)

Lambda Tuning (1)

- ❑ Developed for achieving smooth setpoint response or load change
- ❑ Guarantees stability, robustness and no overshoot
- ❑ Very popular in pulp & paper industry

Two basic steps of lambda (λ) tuning:

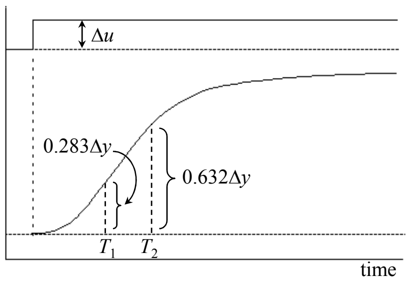
- Process model identification
- Lambda tuning
  - 1st order process with dead time
  - Integrating process (i.e. level control)

Lambda Tuning (2)

1st Order Process with Dead Time

Procedure:

- Manually, bump the *CO* then observe the PV_m



The estimated process parameters:

- Process gain: $G_p = \dfrac{\Delta y}{\Delta u}$
- Process time constant: $\tau_p = 1.5(T_2 - T_1)$
- Process dead time: $T_d = T_2 - \tau_p$

Lambda Tuning (3)

1st Order Process with Dead Time (contd.)

Lambda Tuning:

- Choose the desired closed-loop time constant, λ (typically 2 to 3 times the process constant) ← sluggish response!

PID tuning parameters:

- Proportional gain: $P = \dfrac{2\tau_p + \tau_d}{2G_p(\lambda + \tau_d)}$
- Integral action: $T_m = \tau_p + \dfrac{\tau_d}{2}$
- Derivative action: $D = \dfrac{\tau_p \, \tau_d}{2\,\tau_p + \tau_d}$

Lambda Tuning (4)

Lambda Tuning's Rule of Thumb:

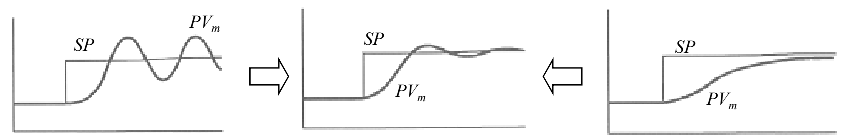
- Integral time should no be smaller than the process time constant
- Level control oscillating? Remove nearly all integral action
- Poll time should be less than one-tenth the process time constant
- Filter time constant should be less than one fifth the process time constant
- Closed-loop time constant is usually greater than the process time constant

Visual Loop Tuning (1)

Problems

- The loop is unstable (or apparently so)
- The loop is sluggish in response to upsets or setpoint changes

How to improve the performance of a loop by using **NO** algebra?

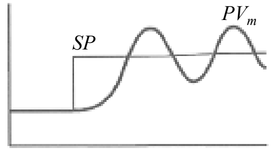


Apparently unstable response — "Good" response? — Sluggish response

Visual Loop Tuning (2)

Apparent Instability

- The loop oscillates
  - Because of excessive feedback, or
  - Of being perturbed periodically by another process



Procedure:

- Put the loop in manual (if it is safe to do so)
- In manual mode, the process appears to settle down → poorly tuned

Tuning problems:

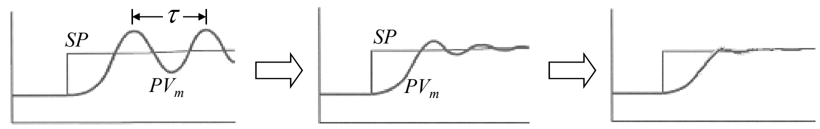
- Is the oscillation caused by too much or too little gain/integral/ derivative or their combinations?

Visual Loop Tuning (3)

Apparent Instability (contd.)

Tuning procedure:

- **Self-regulating Processes**
  - **If** the value of I_m (min/repeat) is **less than half** of the oscillation period τ
    - First, **Increase** the value of I_m
    - If the value of I_m (min/repeat) is **longer than** the oscillation period, it is safe to decrease the gain (*P*)



Apparently unstable response | Increasing the value of I_m (min/repeat) | Lowering the gain (*P*)

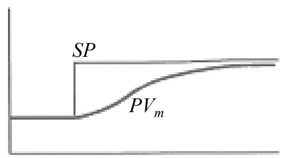
- **Non Self-regulating Processes**
  - Use the longest value of I_m (min/repeat) as much as possible or completely remove the integral action. If the problem persists, then lowering the gain (*P*)

Visual Loop Tuning (4)

Sluggish Response

Common causes:

- The loop usually has no derivative action
- The value of I_m (min/repeat) is long relative to the process response time
- The value of gain is too low



Visual Loop Tuning (5)

Sluggish Response (contd.)

Tuning procedure:

1. **Adjusting Gain**
   - Set the I_m as longest as possible and set *D* to zero.
   - Place the controller in manual mode, then step out the CO at a reasonable value
   - Immediately put the controller back in auto mode. Watch the process response to know what the controller action does
   - Repeat the process until we get one cycle of process output swinging

Swinging more than one cycle
Too much gain
→ lowering the gain | Swinging less than one cycle
Too little gain
→ increasing the gain | Swinging in one cycle
Reasonable gain achieved

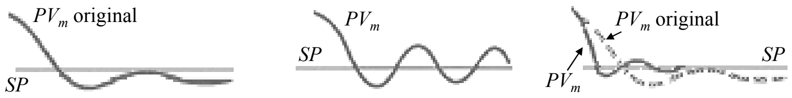
Visual Loop Tuning (6)

Sluggish Response (contd.)

Tuning procedure:

2. Adjusting Integral Action

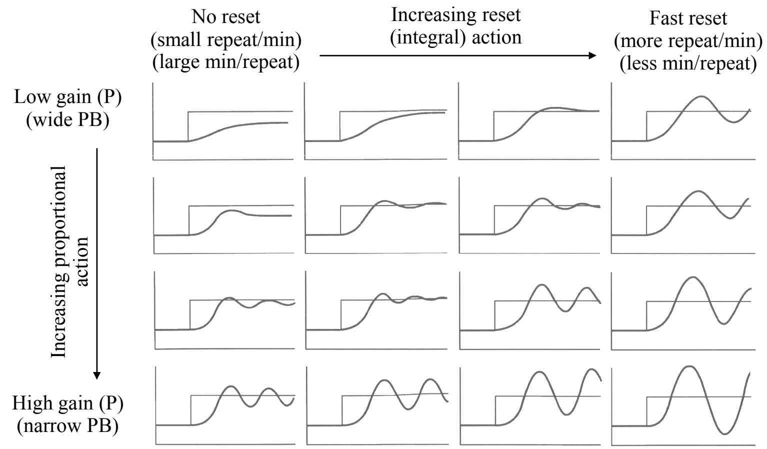
- Place the controller in manual mode, shorten the value of I_m, then step out the CO at a reasonable value
- Immediately put the controller back in auto mode. Watch the process response to know what the controller action does
- Repeat the process until we get the PV_m ramps back to setpoint about **half** as fast as it moved away from setpoint form the CO step



Response with reasonable gain (swinging in one cycle) | To little I_m (swinging at setpoint) | Reasonable I_m

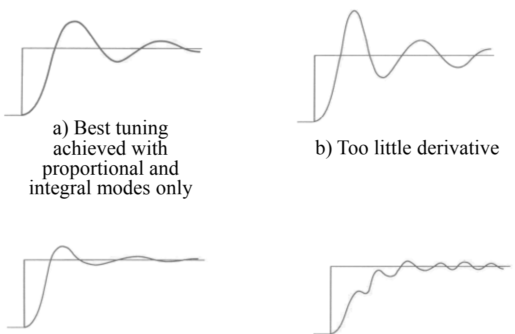
Visual Loop Tuning (7)

Tuning map for gain (P) and reset effect (I_m)



Visual Loop Tuning (8)

The Effect of Adding Derivative



a) Best tuning achieved with proportional and integral modes only

b) Too little derivative

c) Derivative added gain increased integral action faster

d) Too much derivative

What is Autotuning?

- Autotuning (also known as self-tuning) is a feature supplied by many controller, PLC and DCS vendors that allows the controller to “tune itself”
- It minimize the task of a control engineer in manually tuning the loops

Why Autotuning?

- The process is nonlinear or operated under widely varying conditions
  - → Need various combination of tuning parameters for different operating condition ← can be also accomplished by using operator's log
- The process characteristic change rapidly
  - → Frequent manual changing of the tuning parameters can not be expected to be able to produce satisfactory results
- The end user doesn't have the knowledge or experience for successful manual tuning

Autotuning Categories

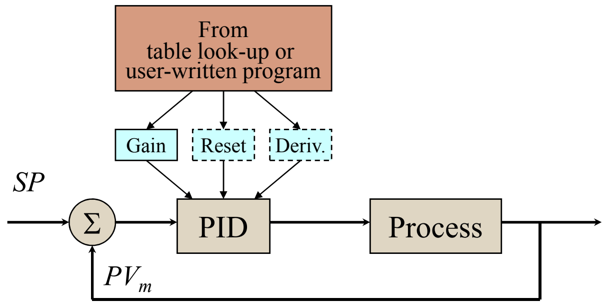
A variety of autotuning techniques found on the market:

- Scheduled tuning
- On-demand tuning
- On-line tuning

Scheduled Tuning (1)

- Merely, an automation of the "operator's log" concept
  - The users have to provide the correct value either by means of a table look-up or a user-written program
- Tuning parameters are changed automatically as operating points change
- No assessing and modifying of the controller performances by determining improved tuning parameters

Scheduled Tuning (2)



Region	Boundaries	Gain	Reset	Deriv.
1	0 – 30%	P_1	I_1	D_1
2	30 – 70%	P_2	I_2	D_2
3	70 – 100%	P_3	I_3	D_3

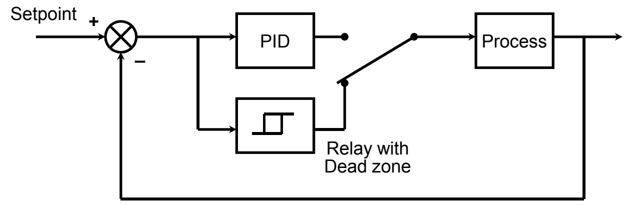
- Example: Fisher DPR900^{TM} Single-loop Controller

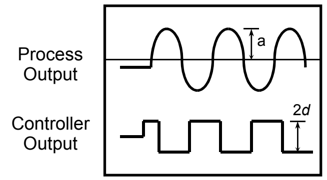
On-demand Tuning (1)

- It is simply an automation of the open- or closed-loop testing method
  - Open-loop tuning methods:
    - Ziegler-Nichols (most common)
    - Lambda tuning
    - ...
  - Closed-loop tuning method:
    - Relay Feedback autotuning ← motivated by Ziegler-Nichols closed-loop tuning method
- User presses a ‘tune’ button to start the tuning procedures which carry out automatically
- The tentative tuning values are display for confirmation. If confirmed, they are inserted into the control algorithm

On-demand Tuning (2)

Relay Feedback Autotuning Method





On-demand Tuning (3)

Relay Feedback Autotuning Method (contd.)

The Procedure

- Forces the system to oscillate by a relay controller
- Require a single closed-loop experiment to find the ultimate frequency information
- No *a priori* information on process is required
- Switch relay feedback controller for tuning
- Find P_{crit} and calculate τ_{crit} according to the formula

$$\tau_{crit} = \frac{4\,d}{\pi\,a}$$

- User specified parameter: *d*
  - Decide “*d*” in order not to perturb the system too much
- Mostly use Ziegler-Nichols tuning rules for PID tuning parameters

On-line Tuning (1)

- The tuning parameters are determined by an auxiliary program that automatically evaluates the closed-loop behavior and calculates and modifies the tuning parameters whenever necessary
- Methods:
  - Pattern recognition
  - Others use a more formal mathematical procedure

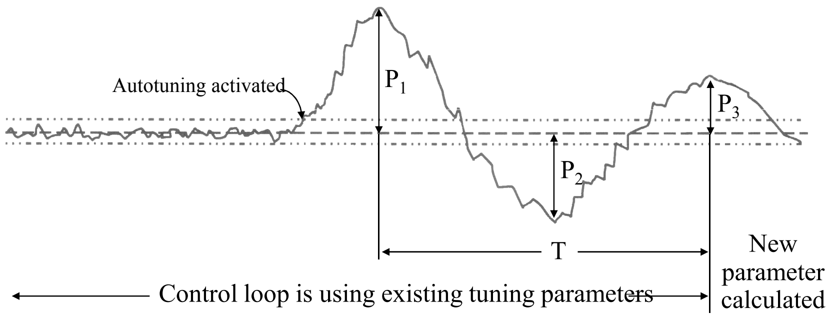
On-line Tuning (2)

Pattern Recognition Approach

- Example: The Foxboro EXACT™ (Expert Adaptive Controller Tuner)
  - It observers the pattern of the response, then invokes a set of rules for determining new tuning parameters that will drive the pattern closer to a desired response pattern
  - The technique:
    - Does not require artificial load upsets – instead it utilizes the normal process disturbances that occur; and
    - Does not attempt to impose an arbitrary mathematical model on the process

On-line Tuning (3)

- The Foxboro EXACT™



Session Summary

- Manual tuning of PID controller can be conducted in various ways by means of some plant test
- There simply is no way to analytically tune a controller if we do not know the type of algorithm and the units
- Autotuning simplifies the tuning procedure of PID controller