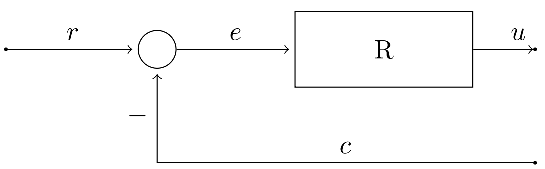
Controllers

1 PID controllers

The most widespread 85% are PIDs, 15% are others.

- easy of use (easy to tune)
- robustness (works even if badly tuned)



$e = r - y$

$u(t) = F[e(t)]$

Terms in industry:

SV Set Value (r)

SP Set Point

PV Process/ Present Value(c)

CV Control variable (u)

MV Manipulated Variable

$$u(t) = K_c \cdot e + K_i \cdot \int e \, \mathrm{d}t + K_d \cdot \frac{\mathrm{d}e}{\mathrm{d}t} \tag{1}$$

Parameters K_c, K_i, K_d

Components P, I, D

1.1 Controller's components

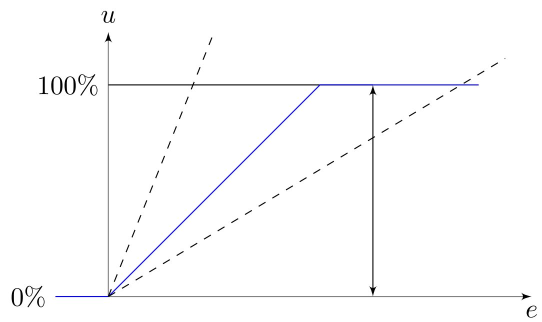
Proportional control

control law

$$u = K_c \cdot e$$

control signal proportional to the error, where K_c is a proportional `gain`/võimendustegur/ коэффициент усиления

Controller static characteristic is $u(e)$ with respect to error e. Output value is limited $0 \ldots 100\%$



three parts:

$$u = \begin{cases} 0\% & - \ \text{min} \\ K_c \cdot e & - \ linear \\ 100\% & - \ \text{max} \end{cases}$$

Output signals types: $(4 - 20)\ mA$, 0-4000, 0-1, $0\% \ldots 100\%$.
The gain K_c units are (if we deal with a thermal process) $[\%/^\circ C]$.
Proportional action responds only to a change in the magnitude of the error.
The controller has an error between PV and SP called
`proportional-only offset` / staatiline viga / статическая ошибка.

> Proportional action will not return the PV to set point ($PV \neq SP$). It will, however, return the PV to a value that is within a defined span around the PV [6].

To minimize the proportional-only offset we need to increase the gain of the controller gain (decreasing its proportional band). That makes controller more "aggressive". However, too much controller gain and control system becomes unstable (oscillations).
Another way: human operator places the controller in manual mode and move the controlled actuator just a little bit more u_0, so $PV = SP$, and then place the controller back into automatic mode. Otherwise, we need more sophisticated control techniques [2].

Integral control, reset action

The purpose of I action is to eliminate offset. Unlike proportional action, which simply moves the output an amount proportional to any change in PV or SP, integral control action does not stop moving the output until all error has been eliminated.

$$u = K_i \int e \ \mathrm{d}t = \frac{1}{T_i} \int e \ \mathrm{d}t,$$

K_i - transfer gain (repeats per minutes);
T_i - integral time constant (minutes per repeat).

The integration symbol tells us the controller will accumulate ("sum") multiple products of error (e) over tiny slices of time (dt)[2, 3].
It makes the system less stable.

Derivative control

$$u = K_d \cdot \frac{\mathrm{d}e}{\mathrm{d}t}$$

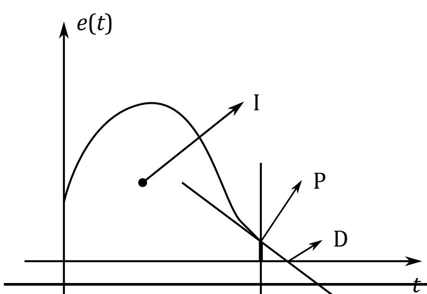
- Takes into account an error change rate;
- Used with other components P, I
  - if speed is a problem, so-called "boost",
  - if object is with a large time constant,
  - in case of PID allows to increase K_c, K_i values;
- In case of lags (delays) usage is not practical;
- Real derivative component:

  $$T_d \cdot s \quad \rightarrow \quad \frac{T_d \cdot s}{1 + \frac{T_d}{N} \cdot s} = \frac{T_d \cdot s}{1 + \alpha \cdot T_d \cdot s}$$

  $N = 1/\alpha$ gain of the derivative component.

  Thus, PID controller has one more pole.

PID controller output signal u is formed from from the error signal e of the past (I), present (P) and future (D).



$$u(t) = K_c \cdot e + K_i \cdot \int e \ \mathrm{d}t + K_d \cdot \frac{\mathrm{d}e}{\mathrm{d}t}$$

If error decreases fast ($\mathrm{d}e/\mathrm{d}t < 0$) there is no need to control with a large signal ($u \gg$).

1.2 Controller's parameters

$$K_c, K_i, K_d \quad \rightarrow \quad K_c, T_i, T_d$$

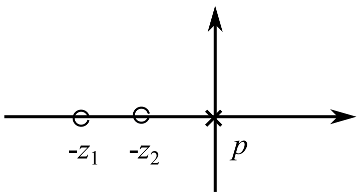
control law

$$u(t) = K_c\Big(e + \frac{1}{T_i}\int e\ \mathrm{d}t + T_d \cdot \frac{\mathrm{d}e}{\mathrm{d}t}\Big)$$

transfer function

$$W_c(s) = K_c\Big(1 + \frac{1}{T_i \cdot s} + T_d \cdot s\Big) = K_c\frac{1 + T_i \cdot s + T_i \cdot T_d \cdot s}{T_i \cdot s}$$

NB! Not realizable - two zeros and one pole



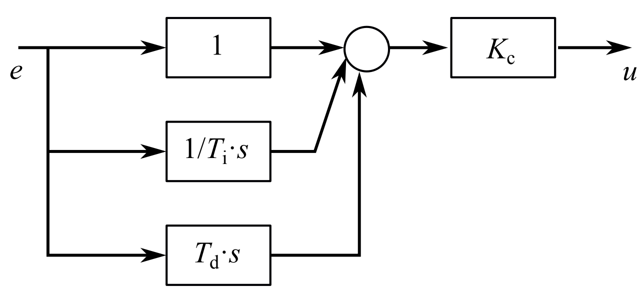
$z_1, z_2 \neq 1/T_i, 1/T_d!$,
$z_1 \cdot z_2 = 1/(T_d \cdot T_i)$,
$z_1 + z_2 = 1/T_d$

1.3 PID controller structures

A combination of all three of the actions described above is more commonly referred to as PID action. There are three major variations how PID equations implemented in modern PID controllers: the *parallel*, *ideal* and *series*.

Parallel PID

$$u = K_c \cdot e + \frac{1}{T_i}\int e\ \mathrm{d}t + K_d\frac{\mathrm{d}e}{\mathrm{d}t} \tag{2}$$



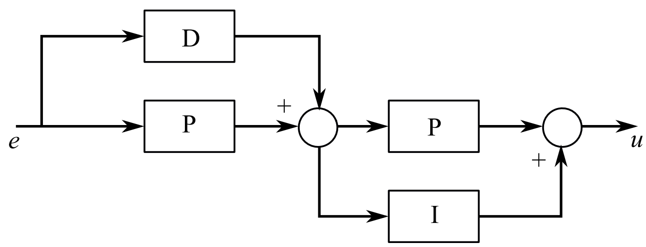
In the parallel equation, each action parameter (K_c, T_i, T_d) is independent of the others. Equation can be broken up three parts, each one describing its contribution to the output u.

Series PID

PID $\{K_c, T_i, T_d\} \Leftrightarrow$ PD + PI $\{K_c', T_i', T_d'\}$

The gain K_c affects all the three actions

$$u = K_c\left(e + \frac{1}{T_i}\int e\ \mathrm{d}t + T_d\frac{\mathrm{d}e}{\mathrm{d}t}\right) \tag{3}$$



$$= K_c'(1 + T_d' \cdot s) \cdot \left(1 + \frac{1}{T_i' \cdot s}\right)$$

The gain constat K_c equally affecting all three control actions. Increasing K_c makes the P, I, D actions equally more aggressive [2].

Transformation PID $\rightarrow$ PD + PI, if we assign $B = T_d/T_i$, then

$$K_c' = K_c\frac{1}{2}\left(1 + \sqrt{1 - 4B}\right);$$
$$T_i' = T_i\frac{1}{2}\left(1 + \sqrt{1 - 4B}\right);$$
$$T_d' = T_d\frac{1}{2B}\left(1 - \sqrt{1 - 4B}\right),$$

that is possible if $0 < B < 1/4$.

Components' inputs

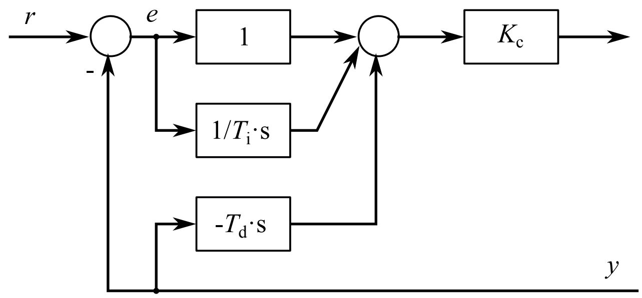
$$(e) \rightarrow (r, y)$$

As input of component D - y signal is used, so controller reacts on disturbance, not the change of r.

Implemented on most industrial controllers!

- As I component input only error can be used
- As P component input change we can use

$$u(s) = K_c\left[\frac{1}{T_i \cdot s}e(s) - \left(1 + \frac{T_d \cdot s}{1 + \alpha \cdot T_d \cdot s}\right)\right]$$

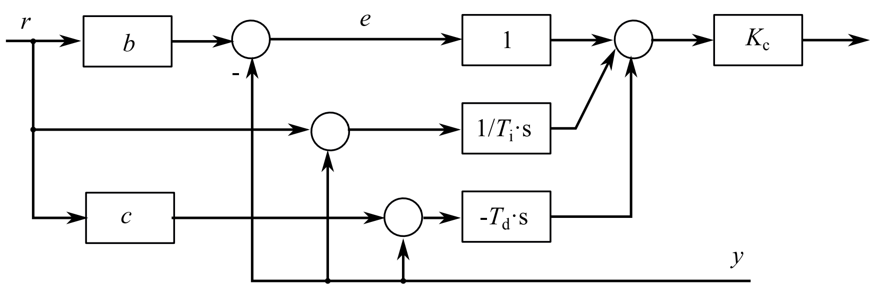


ISA PID

Generalizes above mentioned

$$u(s) = K_c \left[(b \cdot r - y) + \frac{1}{T_i \cdot s} e - \left(1 + \frac{T_d \cdot s}{1 + \alpha \cdot T_d \cdot s} (c \cdot r - y) \right) \right]$$

Inputs of the components can be changed by the coefficients b, c. Why is that? It gives additional



opportunities to tune controller.

Standard PID control tuning determines closed system properties for both reference and disturbance transfers.

ISA-PID could be considered as double controller: C_1 with r input and C_2 with e input.

1.4 Parallel versus Series

The series PID equations and the parallel PID equations are not identical. However, the differences are small. The tuning coefficients in the parallel form of the PID control equation can be algebraically related to the tuning coefficients in the series form of the PID, see Table 1.

Where coefficients for parallel form are K_c, T_i, T_d and K'_c, T'_i, T'_d are series form coefficients [5].

Table 1: Tuning Coefficient Conversion Equations

Mode	Series to Parallel	Parallel to series
Proportional	$K_c = K_c' \cdot \dfrac{T_i' + T_d'}{T_i'}$	$K_c' = \dfrac{1}{2} K_c \left[1 + \left(1 - \dfrac{4T_d}{T_i}\right)^{1/2}\right]$
Integral	$T_i = T_i' + T_d'$	$T_i' = \dfrac{1}{2} T_i \left[1 + \left(1 - \dfrac{4T_d}{T_i}\right)^{1/2}\right]$
Derivative	$T_d = \dfrac{T_i' T_d'}{T_i' + T_d'}$	$T_d' = \dfrac{1}{2} T_i \left[1 - \left(1 - \dfrac{4T_d}{T_i}\right)^{1/2}\right]$

2 Controller Tuning

To implement continuous control we should assemble a control loop which consists of the process/object, controller, sensors and actuators.

Information about the control loop Find, read or write documentation which describes work of the control loop: Process Control Philosophy. Where the next information can be found:

- process description and relationship with other parts;
- goals and requirements;
- staff: knowledge, skills, roles, responsibilities, duties, training;
- loop architecture: signals, parameters, sampling time, database management, rules, software tools;
- control techniques, procedures, time interval.

Control loop should be designed for a specific project, to implement control a controller should be tuned.

NB! Tuning of the SISO control is a basic knowledge in automation. Simply experimenting with P, I, and D parameter values is tedious at best and dangerous at worst! Do not do if you have no understanding of what each type of control action if useful for, and the limitations of each control action.

2.1 Controller tuning

To tune a controller you need carry out the next procedures

1. Check loop devices: sensors, actuators, etc.

- range, calibration, dynamics;
- find a problem and solve it

  do not tune controller in worthless loop!

2. Derive a process model
   - trial-error method also gives some results;
   - autotuning also needs some initial parameters.
3. Describe needs, *requirements*, goals
   - accuracy, speed, robustness.
4. Choose the algorithm: PI, PID, etc.
5. Tune the controller
   - there are a lot of acceptable methods, choose the best;
   - take into account that feedback loop has its own limits that cannot be exceeded.
6. Simulate the loop, make sure it works with SV change, different loads and disturbances.
7. Observe work of the control loop
   - discover: differences, unexpectedness;
   - document the results: test, parameters, etc;
   - observe control loop in the future

     process, as equipment changes.

2.2 PID controller tuning

Controller has several free parameters (tuning parameters) changing them controller can be prepared for work with a

- given process,
- according to requirements.

Sad Statistics

- 50% of controllers badly tuned, 1/3 oscillates,
- just 4% of tuned parameters are changed during last two years.

Badly tuned controller still works...

Control performance can be evaluated.
What are important features of controller work?

Well-tuned controller saves energy and materials, increases quality of the product.

How to tune a controller?

1. Use your knowledge and experience from the similar projects

   empirical equations, guidance;

2. Use model of the process/object

   set a goal, synthesize a controller;

3. Autotuning.

Different methods give similar but not matching results.

2.3 Tuning equations and rules

If process properties is not known do the test:

- step response

  test with a stable object

- frequency response

  assemble control loop, observe oscillations

How are rules and equations obtained?

A lot of tests and simulations have been done with different objects and controllers (P, PI, PID), thus closed system properties were found out.

Rules and equations are derived from the obtained data, which associates the controller parameters (K_p, T_i, T_d) with test or model parameters (K, T, τ) and system properties.

Those equations are approximate and can be applied to parameters with a limited range. Controller has free parameters (tuning parameters) which fit for the

- given process and
- according to requirements.

Some loops cannot be tuned by any approach trial-and-error, traditional tuning techniques, or automated tuning. The most common problems in order obtain good performance of the control are:

- Process non linearities.
- Drawbacks on the process and P&I diagrams.
- Problems with the final control element (especially valves).
- Large dead times.
- Noise.
- Loop interaction.
- Improper nesting of cascade loops.

3 Open-Loop Methods

The open-loop tuning methods execute the process test with the controller on manual. The test data consist of the response in the process variable to a known change in the controller output. The most common problem in applying an open - loop tuning method is that the process test is not executed properly.

3.1 Ziegler-Nichols method: Reaction Curve Method

First systematic approach to tune PID controllers. The Ziegler-Nichols methods (open-loop and closed-loop) provide quarter wave decay tuning for most types of process loops. This tuning does not necessarily provide the best ISE or IAE tuning but does provide stable tuning that is a reasonable compromise among the various objectives.

Because of their simplicity and because they provides adequate tuning for most loops, the Ziegler-Nichols methods (1942) are still widely used.

After making the step-change in output signal with the controller in manual mode, the process variable trend is closely analyzed for two salient features: the *reaction lag* and the *reaction rate*. Reaction lag is the amount of time delay between the output step-change and the first indication of process variable change. Reaction rate is the maximum rate at which the process variable changes following the output step-change (the maximum time-derivative of the process variable).

Substitute the values of the reaction lag and reaction rate into the tuning equations in Table 2.

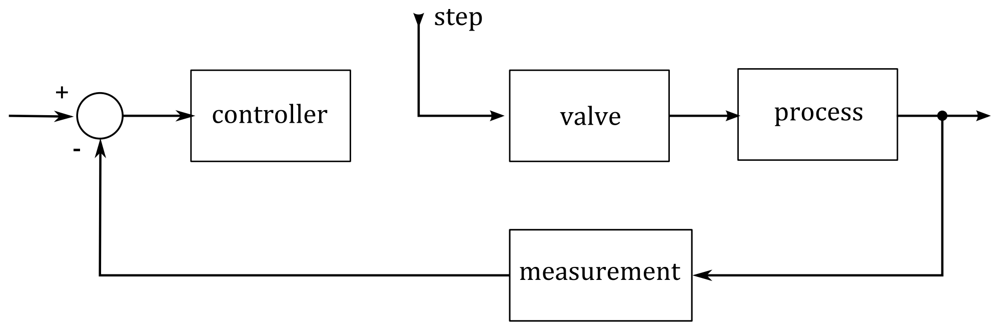


Figure 1: The open loop reaction curve method

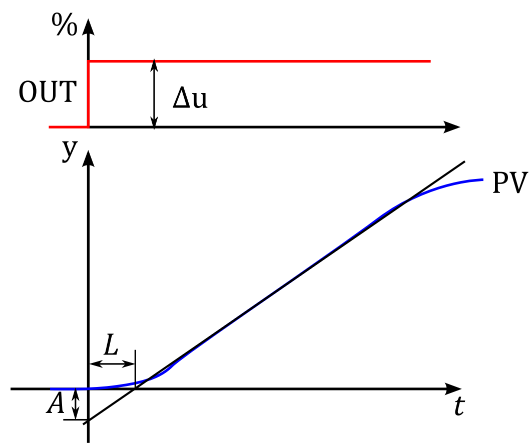


Figure 2: Characteristic "S-shaped" reaction curve

Table 2: Open-loop Ziegler-Nichols tuning method

controller	K_c	T_i	T_d
P	$\frac{\Delta u}{A}$	-	-
PI	$0.9\frac{\Delta u}{A}$	$3.33L$	-
PID-series	$1.2\frac{\Delta u}{A}$	$2.0L$	$0.5L$

Where Δu is a controller output step-change magnitude while testing in open-loop mode and if FOPDT model is known, then $\dfrac{1}{A} = \dfrac{T}{K_p \cdot \tau}$. To give a response with a quarter decay ratio, Ziegler–Nichols proposed the tuning equations in Table 2. Ziegler–Nichols only provided the coefficients for the series form of the PID (the parallel form could not be implemented in the pneumatic controllers available in 1942).

Some comments

- applicable to stable object with no oscillations;
- easy to use;
- some processes do not permit step response tests or it gives a little information about the process, the step input applied should be small enough for the response to stay within the bounds of linearity;
- tuning criterion is a speed-oriented, aggressive, strongly oscillating process, not robust process, sensitive to changes;
- reaction on disturbances.

This method [2, 4, 5] was a basis for developing of the following methods.

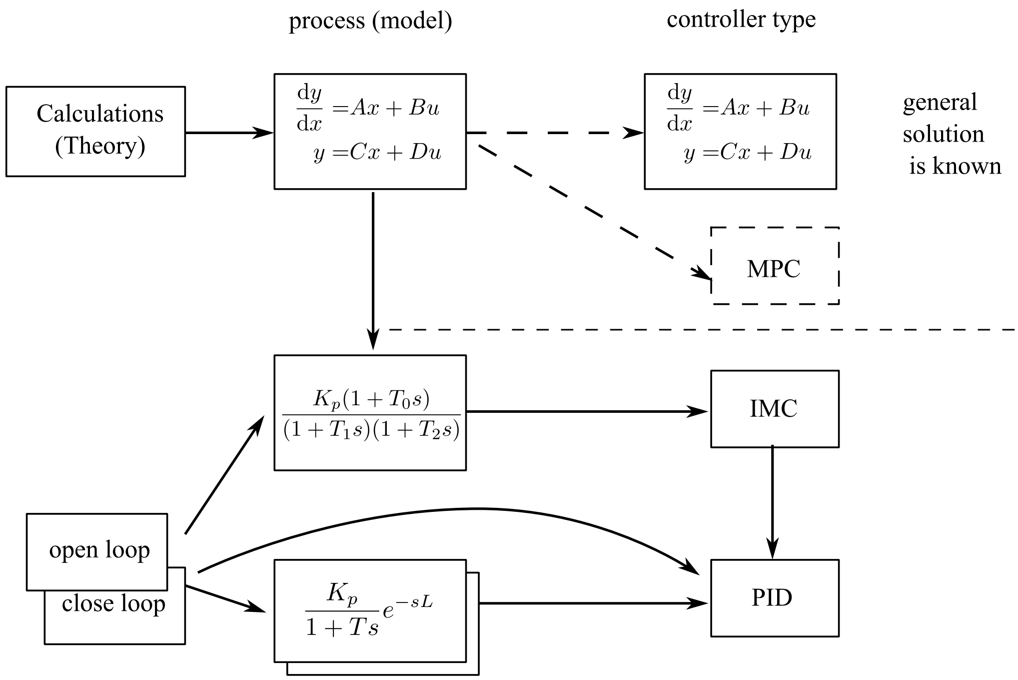


Figure 3: Overview of the control techniques

Bibliography

[1] Don Green and Robert Perry, *Perry's Chemical Engineers' Handbook*, The McGraw-Hill Companies, Inc, 8-th Edition, 2008.

[2] Tony R. Kuphaldt, *Lessons In Industrial Instrumentation*. URL:`http://www.pacontrol.com/industrial-instrumentation.html`, 2012.

[3] Jonathan Love, *Process Automation Handbook: A Guide to Theory and Practice*. Springer-Verlag London Limited, 2007.

[4] John A. Shaw, *The PID Control Algorithm How it works, how to tune it, and how to use it*, Process Control Solutions, 2nd Edition, 2006.

[5] Cecil L. Smith, *Practical process control: tuning and troubleshooting*, John Wiley & Sons, Inc., Hoboken, New Jersey, 2009.

[6] *Instrumentation & Control: Process Control Fundamentals*, URL:`http://www.PAControl.com`