

Understanding P, I, and D

R. Russell Rhinehart

Develop Your Potential Series in CONTROL magazine, Vol. 31, No. 2, February 2018, pp. 40-44.

It is important to understand what the proportional, integral, and derivative terms do within the PID controller. That understanding is essential to choose appropriate action, trouble shoot controllers, chose appropriate modifications, or set up advanced controllers. Unfortunately, the controller synthesis approaches, in which PID magically appears within Laplace Transform analysis, does not provide that functional understanding. Hopefully, this more intuitive development of PID will be helpful.

Chosen as an example is a commonly understood process – level control in a tank of liquid. See Figure 1. The inflow is a wild variable, or disturbance that will upset the level, which is an indication of liquid inventory. The slide valve on the outflow will open or close to release more or less fluid to keep the level at the desired set point. As notation the controlled variable, CV, is the liquid level in the tank, h , and the manipulated variable, MV, is the valve stem position, U . Recognize that your region or community may use alternate terminology. The nominal, initial steady state values are h_0 and U_0 .

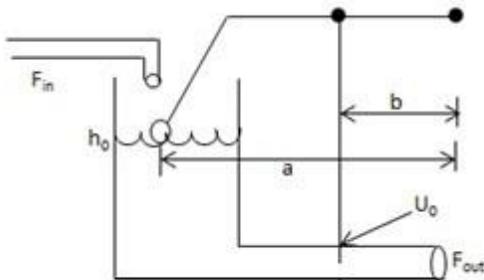


Figure 1. Physical sketch of process and lever-type controller

The controller, also illustrated in Figure 1, is a mechanical lever, a proportional-only controller. If the liquid level rises somewhat, then the float rises the same amount, which raises the lever, which opens the valve an additional amount by the lever proportionality, which releases fluid faster, which seeks to counter the rising liquid inventory. If the liquid level falls somewhat, the lever closes the valve proportionally.

The liquid level may rise or fall for any number of reasons. The inflow rate, F_{in} may change, the viscosity of the fluid may change affecting its outflow speed, or the downstream pressure or in-pipe flow restrictions may change affecting F_{out} . The reason for a rise or fall in level is immaterial for the controller; this lever action moves the valve in the appropriate direction, and in a manner that is proportional to the level change.

Proportional Action

The lever arm length ratio, b/a , is the gain of the controller. By changing the relative lengths of the lever arms, by perhaps changing the position of the fulcrum, the controller can be more or less aggressive. If

the level, h , starts at a base case of h_0 , which is also the set point, h_{SP} ; using simple relations the equation for the change in the valve stem position w.r.t. level is:

$$\Delta U = \frac{b}{a} \Delta h = -\frac{b}{a} (h_{SP} - h) = K_c e \quad (1)$$

Where $K_c = -\frac{b}{a}$ represents the controller gain and $e = (h_{SP} - h)$ represents the actuating error, the deviation of the CV from its set point.

Since $\Delta U = U - U_0$, then the equation representing the lever-type control is

$$U = U_0 + K_c e \quad (2)$$

Figure 2 reveals a block diagram of a controlled process using generic symbols for the MV, U , and CV, Y and disturbance, d . Contrasting the illustration of the physical process of Figure 1, this represents the path of cause-&-effect information exchange. The controller is shown acting on the actuating error, e .

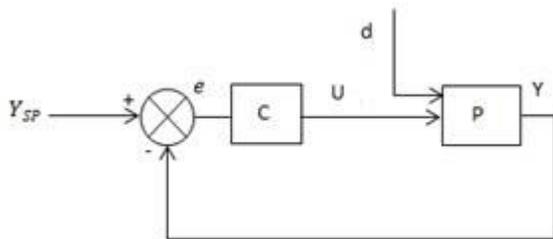


Figure 2. Block diagram representation of a controlled process

Note: Although termed the process output, Y , the level of the liquid does not come out of the tank. The material liquid goes in or comes out, and level is a measure of the inventory response of the in-tank contents. The block diagram reveals the information transfer between controller and process, not material exchange. The lines in Figure 2 are not pipes. The block labeled C is the controller, which represents the lever, but could be a calculator that executes Equation (2). It is simple arithmetic: The controller multiplies K_c times e then adds it to U_0 .

Equation (2) seems very much like what is often presented as a proportional controller, $U = K_c e$. However, if $U = K_c e$ was the relation, then if the CV were at the set point, if $e = 0$, the controller would set $U = 0$, and close the valve, which would make the level rise, and cause it to deviate from the set point. In Equation (2), the term U_0 is the controller bias, it is the value of U for which the initial MV position is required to hold the CV at set point. As illustrated in Figure 1, this is about 50%.

The user chooses the controller gain. Normally, the controller starts in manual mode (MAN) with the user deciding the MV value, then when the CV is at the set point, the user switches the controller to automatic (AUTO) mode. For bumpless transfer, the bias is usually set by the controller as the MV value when the controller is switched from MAN to AUTO.

A block diagram of the arithmetic operations in the P-only control logic of Equation (2) is shown in Figure 3. The actuating error is multiplied by the controller gain and then added to the bias to determine the controller output. These are simple arithmetic operations (not calculus or Laplace transformed magic).

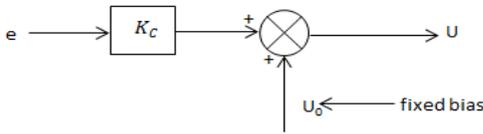


Figure 3. Block diagram of the P-only control calculation

Note: It does not matter whether the controller in Figure 1 is an actual physical float-&-lever device, or whether it is a digital calculation of Equation (2) in a computer that sends the valve stem position target to an i/p device to move the valve stem. The logic and action are identical.

Steady State Offset and Integral Action

P-only control is often good enough, but its problem is steady state offset. Consider what happens if F_{in} increases and holds at a new value, when the level is initially at the set point. Initially, F_{out} remains the same because h has not yet changed, and the valve stem position is at the initial U_0 . Then, since the new inflow is greater than the outflow, level rises. As h rises, this increases U , which increases F_{out} . Eventually F_{out} will match F_{in} and the level will stop rising; but at this new steady state, h is not at the set point. It must be above the set point for the valve to be open enough to let the outflow be higher. This h deviation is steady state offset.

It is immaterial whether the disturbance is the inflow rate or some other aspect that affects either the inflow or outflow, or whether the level falls or rises as a response to the disturbance. If the disturbance persists, the process will not settle at the set point.

A method to eliminate steady state offset is to add integral action. But, that calculus terminology has little physical meaning; so, to add understanding consider the injection of a turnbuckle to the valve stem as illustrated in Figure 4.

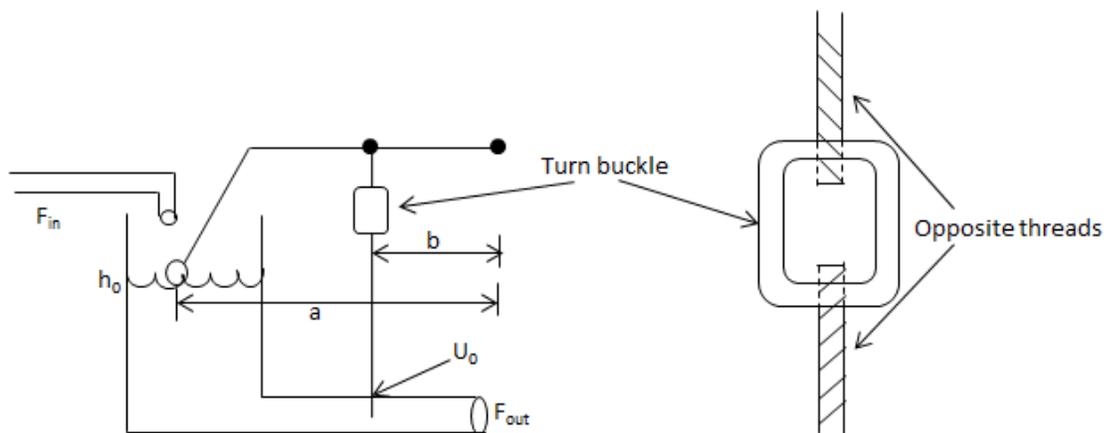


Figure 4. Including an adjustable valve stem feature to the controller

In a turnbuckle, the ends of the rods are threaded to fit into the threaded holes of the buckle. The threads go in opposite directions. So, if the buckle is turned in one direction the two sections of the valve stem are pulled together, the stem is shortened, and the valve opens. If turned in the other direction, the stem is lengthened and the valve closes. This permits opening and closing of the valve without a change in tank level.

If you notice that the liquid level has risen, you know that some disturbance is acting, and the buckle needs to be turned to shorten the stem, to open the valve a bit more than the original, pre-disturbance position for the tank level. If the level rises a little bit, there is only a small upset, and the buckle only needs to be turned a little. In contrast, a large level deviation indicates a large upset has occurred, which justifies a large turnbuckle readjustment. And, of course, a level rise or drop would direct turns in the opposite direction.

So, let's have an observer follow this rule: *At each sampling, observe the level deviation from set point, and make an incremental change in the turnbuckle angle that is proportional to the level deviation.* In Equation (3), α represents the thread pitch (axial distance per angle), β is the proportionality rule to change angle due to level deviation from set point (angle per h deviation), and $c = \alpha \beta$ is their product.

$$\Delta U = \alpha \Delta \theta = \alpha \beta e = ce \quad (3)$$

After the most recent sampling, the i^{th} observation, control action changes the valve stem length from the previous length.

$$U_i = U_{i-1} + ce_i \quad (4)$$

The sequence of the past two adjustments is

$$U_i = U_{i-2} + ce_i + ce_{i-1} \quad (5)$$

Continuing to include past adjustments generates an equation that indicates how all of the adjustments have contributed to the current valve stem length change from the initial value, U_0 .

$$U_i = U_0 + c \sum_{i=1}^N e_i \quad (6)$$

In Equation (6) the number of items in the sum is $N = \frac{t}{\Delta t}$, where t is the total time that the controller has been in AUTO, and Δt is the controller sampling interval. Multiplying and dividing the sum by Δt reveals that the sum of rectangles (e -height times Δt -base) is just the rectangle rule of integration, which can be represented by the calculus notation.

$$U(t) = U_i = U_0 + \frac{c}{\Delta t} \sum e_i \Delta t = U_0 + \frac{c}{\Delta t} \int_0^t e dt \quad (7)$$

Note: Even though the integral symbol is used in Equation (7), no calculus procedure was used by the turnbuckle adjuster. If this were to be implemented by a computer, the Equation (4) adjustment in the valve stem length is not calculus, but a simple algebraic multiplication and addition. Further in the computer, the subscripts are not needed. The assignment statement representing the Equation (4) action is $U := U + ce$. Don't let the integral symbol misdirect your understanding. There is no calculus to the doing of control.

A common form of the controller calculation is to incrementally sum (integrate) the scaled error, $K_C e$, which means that the $\frac{c}{\Delta t}$ coefficient needs to be divided by K_C . Since the term $\frac{\Delta t K_C}{c}$ only has dimensions of time, τ_i one can represent the PI controller function as the block diagram of Figure 5. It shows the integral value (really, it is just the sum of incremental changes) is added to the initial bias to make the controller bias adjust at each sampling. The block diagram notation indicates the function (inside the box) and the function argument (the box input value); but again, don't be thinking calculus, the integral operation is simply an arithmetic incremental accumulation.

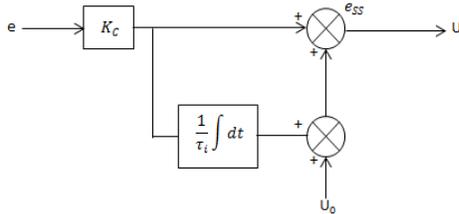


Figure 5. Block diagram of the standard form of the PI controller

In the standard form, K_C is the controller gain that multiplies both the P and I terms, and the integral time, τ_i , divides the integral (which is actually calculated by incremental summation).

Recall, the purpose of the proportional action was to immediately counter the effect of a disturbance, but its problem is that it leaves a steady state offset. The integral purpose was not to be the primary control action, but to remove the offset left by the P-action. Accordingly, tune the P first (K_C) to set the aggressiveness of the controller, then adjust the I-action (τ_i) to remove the residual offset at a desirable rate.

Anticipated Error and Derivative Mode

In that description of P-action, the controller acts on the initial impact of the disturbance on the CV. However, the initial reveal that a disturbance has happened might be a small CV deviation; but, if the deviation is allowed to fully develop over time it might evolve to a large value. The eventual value of the fully developed CV deviation, not the initial indication, represents the magnitude of the disturbance that is causing the CV to start deviating. With Derivative action, the controller will take proportional action based on the anticipated error, not just on the initial reveal of the CV deviation. The question is how to forecast the anticipated error, the fully developed result of a disturbance.

If the process is linear and has a first-order response to a disturbance, then the model of how e would respond to a change in a disturbance, Δd , is

$$\tau_d \frac{de}{dt} + e = K_d(\Delta d) = e_{anticipated} \quad (8)$$

The value of $e_{anticipated}$ is the steady, fully developed, anticipated value.

Note: If the disturbance could be measured and the process gain to the disturbance were known, then $e_{anticipated}$ could be calculated from $K_d(\Delta d) = e_{anticipated}$. However, those are often unmeasured and unknown. Fortunately, Equation (8) reveals that one can estimate the anticipated future error based on the current actuating error and its rate of change, the values of which are already known by the controller. (Assuming the process response to the influence is first-order.)

$$e_{anticipated} = \tau_d \frac{de}{dt} + e \quad (9)$$

Equation (9) does not specify what the disturbance is. The deviation could indicate a confluence of several disturbances, including the MV. Equation (9) is called a “lead”, which should be familiar. It represents what a ball thrower must do to have a running target catch the ball. The ball must be thrown to where the receiver will be when the ball gets there, not to where the runner is now. The value of e is the present location, and the runner velocity is $\frac{de}{dt}$. The value of τ_d is the time for the ball to get to the runner, so the product $\tau_d \frac{de}{dt}$ is how far the runner advances while the ball is in the air, which is how far in front you must aim the ball. So, $e_{anticipated} = \tau_d \frac{de}{dt} + e$ is the location to aim the ball to intercept the running receiver.

The PI controller with the P-action based on $e_{anticipated}$ is

$$U = K_C e_{anticipated} + \frac{K_C}{\tau_i} \int edt + U_0 \quad (10)$$

When Equation (9) is substituted into Equation (10) and rearranged, the PI controller with P-action on the anticipated error is the classical PID relation.

$$U = K_C e + \frac{K_C}{\tau_i} \int edt + K_C \tau_d \frac{de}{dt} + U_0 \quad (11)$$

Although Equation (11) looks like calculus with its $\int edt$ and $\frac{de}{dt}$ representation, the integral is actually just the incrementally updated sum, and the derivative will be calculated from a numerical approach $\frac{e_{new} - e_{old}}{\Delta t}$, which, again, is simple arithmetic subtraction and division.

PD-action is equivalent to P-action on the anticipated error. Whether D action is used or not, one still needs incremental adjustment to the bias, I-action, to remove steady state offset.

If the process measurement is noisy, the numerical derivative amplifies the noise impact. And, if the process is relatively quick to respond, there is no need to use the anticipated error concept. So, only use D-action on noiseless and slow-to-evolve processes.

In block diagram notation, the PID controller (a PI controller with P based on the anticipated error, and the incremental adjustment to remove steady state offset) is represented in Figure 6. The figure uses the Laplace transformed notation, where s indicates the operation to take the derivative of the input to the function block, and $1/s$ indicates to integrate the input. But, again, regardless of the symbols or calculus words to describe the functions all are simple arithmetic operations.

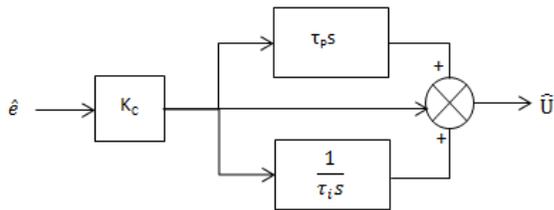


Figure 6. Block diagram of the standard PID controller using Laplace notation

A View of PID

- Proportional control is the simple concept of taking immediate proportional action on the actuating error, but P-only control, $U = K_C e + b_0$, with a fixed bias, leaves steady state offset. The user chooses the value for K_C to set controller aggressiveness.
- Integral action incrementally adjusts the bias to remove steady state offset. Note: Although called “integral” there is no calculus in the action in the incremental accumulation. The user chooses the value for τ_i to set the speed at which offset is removed.
- Derivative action forecasts what the actuating error will be, as a result of past influences, if they are left uncontrolled. It is the lead commonly used in hitting a moving target. PD-action is equivalent to P-action on the anticipated error, and leaves steady state offset. Note: Although called “derivative” there is no calculus in the action of numerically estimating the CV rate-of-change. The user chooses the value for τ_d to lead the aim of the controller.

We communicate the PID procedure with calculus, or Laplace or z-transforms, or other advanced mathematical symbols. With a bit of sarcasm, it seems the reason to use fancy mathematics is to make people think it is difficult, so they need to hire an expert. But, the reality is that the PID calculations are simple arithmetic procedures. Even a computer can do the calculation. By contrast, an expert’s focus on the mathematics distracts those intellects from the important aspects of control; such as structuring ratio, cascade, and override, or choosing appropriate modifications and anti-windup and initialization procedures. The real experts are not necessarily the mathematicians of control theory. They are the ones that can implement control.

There are many modifications to the PID equation. Reset Feedback, for example, is an alternate method to incrementally adjust the bias, which prevents integral windup and is especially useful in override and constraint strategies. Another common modification is the rate-before-reset or interacting controller, which can be created if incremental changes to the bias are also based on the anticipated error. For a discussion of such modifications, see Rhinehart, R. R., H. L. Wade, and F. G. Shinsky, in Instrument Engineers' Handbook, Vol II, Process Control and Analysis, 4th Edition, B. Liptak, Editor, Section 2.3, "Control Modes - PID Variations," pp. 124-129, Taylor and Francis, CRC Press, Boca Raton, FL, 2005.

For a procedure to tune the controller, see CONTROL Magazine, Rhinehart, R. R., "Criteria and Procedure for Controller Tuning", Develop Your Potential Series in CONTROL magazine, January, 2017, Vol. 30, No. 1, pp 54-55.