- **IMan** (initialization manual) – PID action is suspended because of an interruption in the forward path of the controller output. This is typically caused by a downstream block that is not in the cascade mode. The controller output is back-calculated to provide bumpless transfer.
- **RCas** (remote cascade) – The set point is remotely set, often by another computer. PID action is active (closed loop). This mode is also known in older systems as the *supervisory* mode.
- **ROut** (remote output) – The output is remotely set, often by a sequence or by another computer. PID action is suspended (open loop). In older systems, this mode is known as *direct digital control* (*DDC*).

1.4 Is That Your Final Response?

The contribution that the proportional action makes to the controller output is the error multiplied by the gain setting. The contribution made by the integral action is the integrated error multiplied by the reset and gain setting. The reset setting is repeats per minute and is the inverse of integral time or reset time (minutes per repeat). Foundation Fieldbus will standardize the reset time setting as seconds per repeat and the rate (derivative) time setting as seconds. The contribution made by derivative mode is the rate of change of the error or process variable in percent (%), depending upon the type of algorithm, multiplied by the rate and gain settings.

When derivative action is on the process variable instead of on the control error, it works against a set point change. (The control error is the difference between the process variable and the set point.) The reason for this is that it doesn't know the process variable should be changing initially and that the brakes should only be applied to the process when it approaches set point. Using derivation action that is based on the change in control error will provide a faster initial takeoff and will suppress overshoot for a set point change. This is particularly advantageous for set points driven for batch control, advanced control, or cascade control. The improvement can translate into shorter cycle or transition times, an enhancement of the ability of slave loops to mitigate upsets, and less off-spec product because overshoot has been diminished.

When derivative action is used with a time constant there is a built-in filter that is about one-eighth (1/8) of the rate setting. However, you should use set point velocity limits to prevent a jolt to the output when there is a large step change in error from a manually entered set point. This is particularly important when you are using large gains or derivative action based on control error.

The PID algorithm uses percentage (%) input and output signals rather than engineering units. Thus, if you double the scale span of the input (error or process variable), you effectively halve the PID action. Correspondingly, if you double the scale span of the output (manipulated variable), you double the PID action. In fieldbus blocks, both input and output signals can be scaled with engineering units. Using output signal scaling will facilitate the manipulation of the slave loop's set point for cascade control. For controllers that use proportional band, you need to divide the proportional band into 100 percent to get the equivalent controller gain. Proportional band is the percentage change in error needed to cause a 100 percent change in output.

Proportional mode is expressed by the following equation (note that adjustments are gain or proportional band):

$$P_n = K_c * E_n$$

When the derivative mode acts on the process variable in percent (%PV), the equation becomes as follows:

$$P_n = K_c * D_n$$

Integral mode is expressed by the following equation (note that adjustments are integral time or reset):

$$I_n = K_c * 1/T_i * (E_n * T_s) + I_{n-1}$$

When the derivative mode acts on %PV, the equation becomes as follows:

$$I_n = K_c * 1/T_i * (D_n * T_s) + I_{n-1}$$

The equation for derivative mode is as follows (adjustments are derivative time or rate):

$$D_n = K_c * T_d * (E_n - E_{n-1}) / T_s$$

Or for derivative mode on %PV:

$$D_n = K_c * T_d * (\%PV_n - \%PV_{n-1}) / T_s$$

Note the inverse relationships across controllers!

$$\text{Gain } (K_c) = 100\% / PB$$

Reset action (repeats/minute) = $60 / T_i$

Where:

E_n = error at scan n (%)
K_c = controller gain (dimensionless)
PB = controller proportional band (%)
P_n = contribution of the proportional mode at scan n (%)
$\%PV_n$ = process variable at scan n (%)
I_n = contribution of the integral mode at scan n (%)
D_n = contribution of the derivative mode at scan n (%)
T_d = derivative time or rate time setting (seconds)
T_i = integral time or reset time setting (seconds/repeat)
T_s = scan time or update time of PID controller (seconds)

Rule 2 – **If you halve the scale span of a controlled (process) variable or double the span of a manipulated variable (i.e., set point scale or linear valve size), you need to halve the controller gain to get the same PID action.** For controlled variables, the PID gain is proportional to the measurement calibration span. Often these spans are narrowed since accuracy is a percentage of span. For control valves, the PID gain setting is inversely proportional to the slope of the installed valve characteristic at your operating point. If a valve is sized too small or too large, the operating point ends up on the flat portion of the installed valve characteristic curve. For butterfly valves, the curve gets excessively flat below 15 percent and above 55 percent of valve position.

Figure 1 shows the combined response of the PID controller modes to a step change in the process variable (%PV). The proportional mode provides a step change in the controller output ($\Delta\%CO_1$). If there is no further change in the %PV, there is no additional change in the output even though there is a persistent error (offset). The size of the offset is inversely proportional to the controller gain. Integral action will ramp the output unless the error is zero. Since the error is hardly ever exactly zero, reset is always driving the output. The contribution made by the integral mode will equal the contribution made by the proportional mode in the integral time ($\Delta\%CO_2 = \Delta\%CO_1$). Hence, the integral time setting is the time it takes to repeat the proportional contribution (seconds per repeat). The contribution made by the derivative mode for a step change is a hump because of the built-in filter, which is about one-eighth (1/8) of the rate