ControlLogix PID Execution Time

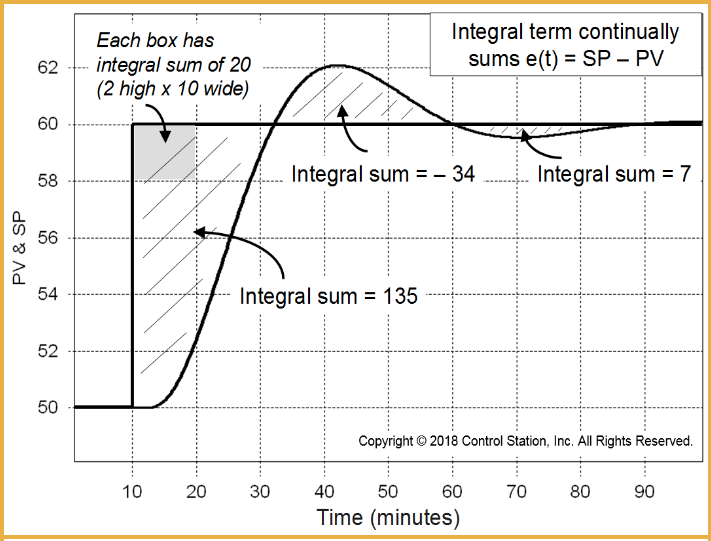
Overview

The ladder logic within Rockwell Automation's ControlLogix® processor can be configured to execute tasks either continuously or periodically. Continuous tasks run in a continual, serial fashion in the background. Once a continuous task is completed it is restarted immediately by the processor. In contrast periodic tasks run on a scheduled basis. When triggered a periodic task will interrupt the execution of other, lower priority tasks in order to perform its function. Only after the periodic task is completed are the other tasks permitted to resume their operation. The ability to program tasks for either continuous or periodic operation provides flexibility and it allows engineers to configure the ladder logic based on the unique requirements of their application. However, improper configuration can severely impact the performance of select functions such as the PID (i.e. Proportional Integral Derivative).

Challenge

Processors like ControlLogix are used to execute a wide array of function blocks that include motion, communication and PID. It is considered best-practice to configure the PID function block for operation as a periodic task. Doing so assures that the time associated with executing the task remains in synch with the PID's sample time. When the PID block is configured and run as a continuous task, however, a timing mismatch commonly occurs. The mismatch is the result of a difference in the execution time of the continuous task and the PID's sample time. That mismatch directly influences the effectiveness of the PID's integral and derivative terms, and it can result in either poor control loop performance or worse.

The primary function of the PID controller is to correct for Error in a process. Of the PID's three terms it is the integral term that is responsible for correcting for offset. Offset is the difference between a control loop's Set Point and its primary input – the Process Variable (PV). Offset is acondition which often results from a sustained disturbance. Rather than responding to the value of Error that stems from the disturbance at a specific time, the integral term sums the associated Error in a continuous fashion. It either adds Error to the Controller Output (CO) when the PV is below Set Point or it subtracts Error from the CO when the PV is above Set Point. In this way the integral term steadily works to close the gap between the PV and Set Point.



As shown in the graphic above the Integral term sums Error in a continuous fashion, adding or subtracting the value from the Controller Output (CO).

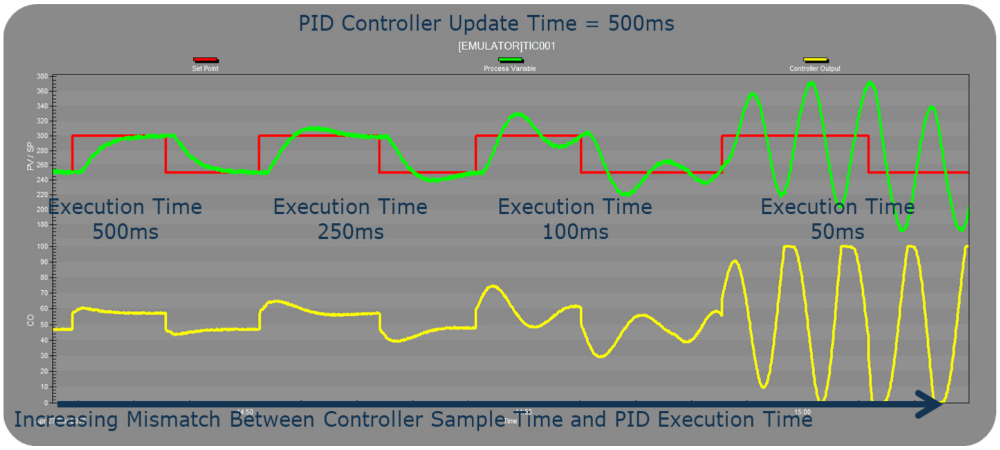
Depending on the number and complexity of tasks executed by the processor the update rate for a continuous task will typically vary by 20 – 50 milliseconds. While the processor's task execution time may vary, the operation of the PID function block has no such inconsistency. The sample time of the PID is fixed. It is for that specific reason that the periodic task is recommended for use with the PID function block. An alternative solution is to apply a timer that executes the PID on a scheduled interval.

Example

Consider a PID controller with a sample time (i.e. .UPD) of 500 milliseconds. The controller calculates the value for integral for this PID using the trapezoid/rectangle method. Now consider that the PID function block is placed in a continuous task. While the time to perform the continuous task varies slightly based on processor loading, the execution time is approximately 20 milliseconds.

The negative impact of applying a continuous task on such a controller's performance is twofold. First, the integral would add up being much faster than expected and any integral tuning parameter entered would provide 25 times more integral than appropriate (i.e. 500 ms / 20 ms = 25). Second, as the processor's load fluctuates the continuous task would either increase or decrease the amount of integral action driving the controller's behavior, resulting in even more variability.

As noted previously a mismatch between a PID controller's sample time and the associated task execution time can have a profound effect on control loop performance. The graphic below showcases the performance of a controller that has a sample rate of 500 milliseconds. The impact of a mismatch becomes evident as the execution time decreases from 250 milliseconds to 50 milliseconds. Whereas performance is acceptable on the far left when sample time matches the execution time, the process shows poor performance once those values are mismatched. Ultimately the process approaches instability on the far right when the execution time is set to 50 milliseconds.



Solution

While Rockwell Automation has a broad portfolio of Programmable Logic Controllers (PLCs) and Programmable Automation Controllers (PACs) the timing mismatch challenge described in this Tech Note is limited to the ControlLogix (i.e. Ladder PID Block) and CompactLogix PACs as well as the company's older PLC-5® PLCs. This challenge is not associated with either the SLC5/02 PLCs or the ControlLogix (i.e. Function Block PIDE), PlantPAx® (i.e. P-PID Blocks) and MicroLogix™ PACs from Rockwell Automation.

There are two options for resolving a mismatch between the controller's sample time and the associated task's execution time. The first is to configure the PID function block to operate as a periodic task instead of a continuous task. The ControlLogix processor supports both types. The other option for resolving such a mismatch is to implement a timer that executes the PID on a scheduled basis. Following is an excerpt from the LOGIX5000 Controllers General Instructions Reference Manual from Rockwell Automation that stipulates the use of a periodic task to correct for timing mismatch:

PID Instruction Timing:

The PID instruction and the sampling of the process cariable need to be updated at a periodic rate. This update time is related to the physical process you are controlling. For very slow loops, such as temperature loops, an update time of once per second or even longer is usually sufficient to obtain good control. Somewhat faster loops, such as pressure or flow loops, may require an update time such as once every 250ms. Only rare cases, such as tension control on an unwinder spool, require loop updates as fast as every 10ms or faster.

Becasue the PID instruction uses a time base in its calculation, you need to synchronize execution of this instuction with the sampling of the process variable (PV).

The easiest way to execute the PID instruction is to put the PID instruction in a periodic task. Set the loop update time (.UPD) equal to the periodic task rate and make sure that the PID instruction is executed every scen of the periodic task.

For a more detailed explanation of the continuous and periodic task options within the ControlLogix processor, please refer to the documentation provided by Rockwell Automation.