

Using a Logix Controller for Barrel Temperature Control on Plastic Injection Molding and Extruding Machines

The Purpose of this Document: To assist machine builders in utilizing the PIDE instruction for extremely accurate closed loop control. This document pays particular attention to controlling temperature (heat/cool or heat-only) loops with slow response, commonplace in barrel temperature control applications found on plastic extrusion and injection molding machines.

Application Description: Extruders and injection molding machines must keep plastic at a specific temperature as it is forced through the barrels of these machines. This is accomplished by constantly monitoring and controlling barrel temperature. The barrels themselves are made of metal, and heat up slowly, usually with significant thermal lag which adds to the difficulty of reaching and holding the desired setpoint. Often, applications also require cooling as well, which is often accomplished by controlling coolant flow through the barrel's coolant jacket, or by controlling the speed of a fan blowing air over the barrel.

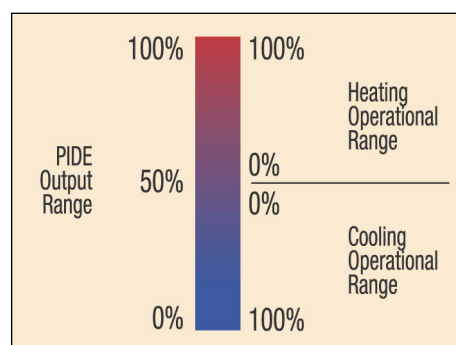
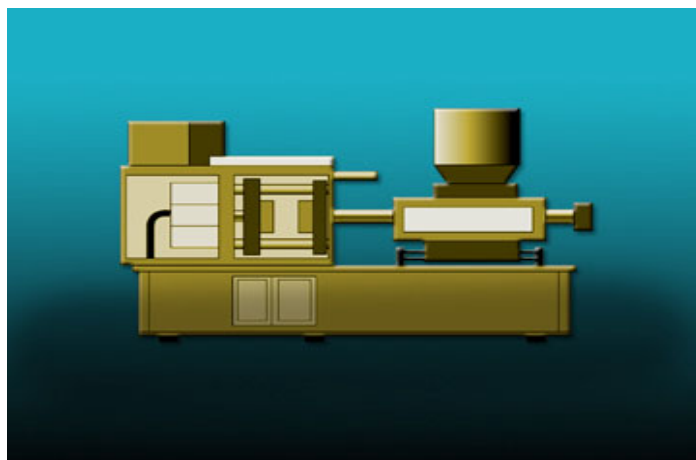
Control System Description: In this document, the control system is made up of (a) a Logix controller, where one or more PIDE instructions with built-in autotuning are executed, (b) multiple analog inputs that provide temperature readings to the controller, (c) multiple discrete outputs which are used to pulse the barrel heaters and cooling system off and on, and (d) an operator interface (HMI). Typically, control is of the 'time proportioned' type. As an example, let's assume that all the heater bands along the barrel are to be controlled over a 10 second duty cycle. On startup, the 'on' percentage of the duty cycle might be as high as 100%, applying power to the heaters for the full 10 second duty cycle. However, as the barrel approaches setpoint, the PIDE instruction begins reducing the percentage of "on" time, and increasing the percentage of 'off' time, perhaps to 30% on (3 seconds) and 70% off (7 seconds) each duty cycle.

The Challenge: The Logix controller is capable of providing multi-zone barrel temperature control every bit as precise as a dedicated barrel temperature control module. However, since the Logix controller is a general purpose device, there are a few 'tricks of the trade' one must understand prior to programming and configuring control loops for barrel temperature control. Once understood, the benefits of running temperature control loops, machine sequencing, and interlocking within the same controller are huge, and worth the small amount of time required to learn them.

Details of the Rockwell Automation Solution:

The Basic Configuration

For most slow temperature loops, you should execute the PIDE instruction in a periodic task with a periodic task update rate of one-half to one second. Since these loops tend to be very slow, there is no point in executing the loop any faster. Make sure that the PIDE instruction is set up for Periodic timing mode. You can check this by making sure that the Timing Mode parameter is set to 0. (Note this is the default value.) This will allow the PIDE instruction to get its update time directly from the Periodic Task's defined update rate.



Heat/Cool loops are split-ranged for operation with two outputs from a single PIDE. This illustration shows how TP outputs relate to PIDE functions.

Most typically, the heating will be performed by pulsing resistive heaters and the cooling will be performed by pulsing a solenoid valve running coolant around the barrel. The Split Range Time Proportional (SRTP) instruction was designed to perform this output pulsing. It is important first of all, that the SRTP instruction is executed much more quickly than the PIDE instruction. Since the SRTP is actually performing the pulsing of the heating and cooling outputs, your output resolution is a function of the CycleTime of the SRTP and how often the SRTP executes. For example, if you defined a CycleTime for the SRTP of 10 seconds, and then executed the SRTP in the same periodic task as the PIDE at once a second, your output resolution would actually only be 10%! It would be impossible to control your loops with this resolution. Therefore, what you want to do is execute the SRTP instruction in a faster, higher priority periodic task typically running every 10 or 20 milliseconds. You can then use a controller scoped tag to send the data from the PIDE output to the input of the SRTP. For a typical loop then, your CycleTime might be 10 seconds, and if your SRTP instruction was running in a 20 millisecond periodic task, then your output resolution is 0.2% which is plenty of resolution to handle most of these types of loops. For heat/cool loops, you typically configure the SRTP instruction such that 100% PIDE output provides full heating, 50% PIDE output provides no heating or cooling, and 0% PIDE output provides full cooling. For a heat-only loop, configure the SRTP such that 100% PIDE output provides full heating, and 0% PIDE output provides no heating. NOTE: This is the default SRTP configuration. Additionally, for a heat/cool loop, you will typically want to set the .CVInitValue parameter of the PIDE instruction to 50. This will cause the PIDE loop to start up with an output of 50% when the controller first goes to run mode. Please note that when you have a system that exhibits a noisy temperature input, you may want to turn on the .DSMOOTHING parameter in the PIDE instruction. This will filter the derivative action so that the derivative gain will not amplify the noise signal through to the loop output. If noise is a serious application problem, you can also filter the temperature signal with a Moving Average (MAVE) or Low Pass Filter (PPF) instruction.

Startup Operation

One of the common issues with extrusion control is the desire to quickly come up to temperature when doing a cold startup at the beginning of the day, for example. Often this is done by entering the desired temperature setpoint and then putting the loop into Auto. A general purpose PID instruction will take a long time to reach setpoint in this manner. This is because these types of instructions are designed to always support a bumpless transfer from Manual to Auto mode. In other words, if the loop output is 0 and the loop is then placed from Manual to Auto, the loop output will still be 0, even if the temperature is far below setpoint. The loop will then slowly increase its output as a result of integral action. In addition, once the temperature does start increasing, the proportional contribution to the output actually starts decreasing (since the error is decreasing), thereby decreasing the output at the same time that the integral action is trying to increase the output. If you try to compensate for this by tuning the PIDE instruction to perform a faster startup, you often end up with gains which do not work well when trying to control around the setpoint. Quite often, an extremely large proportional gain is tried which ends up just turning the loop into an on-off controller and guaranteeing that you will not get stable temperature control around setpoint.

A much better way to start these loops from a cold start is to add a small amount of logic to perform some special actions for cold start. Your logic should check that if the loop is in Auto, and the temperature is more than, for example, 40 degrees below setpoint, then put the loop into Manual with an output of 100% (full heating). Once the temperature is within 40 degrees of setpoint, then drop the manual output down to, for example, 5% heating (ie. 52.5% PIDE output for a heat/cool loop) and put the loop back into Auto. This way, you can tune the loop to obtain the best control around setpoint, and let your cold startup logic handle the cases when you are first starting up in the morning or changing to a different product which requires a new setpoint. You can try different values of your startup cutoff to see if some value other than 40 degrees below setpoint is more appropriate for your extruder or injection molding machine.

Alternate Startup Method: As an alternate startup method, you may just set the PV>TRACKING parameter. This will force the SP (Setpoint) to = the Process Variable (temperature) when the loop is in manual mode. When you do a cold startup, you will put the loop in 'auto' and change the setpoint. This will provide a large proportional term which will cause the loop to reach setpoint more quickly.

Tuning

The PIDE instruction has a built-in autotuner which you can use to generate a suggested set of tuning constants. Note that any autotuner only gives an approximate set of tuning constants. Often these are good enough, but usually can be tweaked somewhat to give better control if desired. To use the PIDE autotuner, the loop must be in manual. You will define a step change size for the manual output in % (5 or 10% is a typical amount) and a process type. For slow temperature loops, you should choose a process type of "Non-Integrating." This will typically give better results than selecting a process type of "Temperature." When you start the autotuner, it will step the output by the amount you configure and monitor the temperature response to that step change. It will then give you a suggested set of tuning constants to use. Usually, the medium or fast response tuning constants will give best results.

For heat-only loops, this set of tuning constants is all you need. For heat/cool loops, you can often use these tuning constants for both heating and cooling, and the control will be good enough. If the cool side is not providing effective control, then you should obtain a second set of tuning constants for cooling. To do this, you will need to artificially provide a heating source to the zone so that your step change in cooling will provide a realistic response. One way to accomplish this would be to run the adjacent zones at your typical operating temperatures to provide a heat source to the zone you are trying to tune. Another way would be to run product through the barrel and rely on shear heating to provide heat. If all else fails, you could temporarily add logic to independently pulse the heating contact for, say, 5% heat for the duration of your cooling autotune. Once the zone is comfortably sitting well above room temperature, start the autotune with a manual output of 0, and a configured step change of -5% or so. This will cause an increase in cooling, and the autotuner will watch the corresponding decrease in temperature and provide a set of suggested tuning constants

Once you have your suggested heating and cooling gains, you would then use some logic to swap these gains. Simply, if the PIDE output is greater than 50% (ie. you are heating) then move the heating gains into the .PGain, .IGain, and .DGain parameters. Otherwise, if the PIDE output is less than 50%, then move the cooling gains into the .PGain, .IGain, and .DGain parameters. This can be done in the Function Block Diagramming editor by just using a GRT instruction comparing the PIDE output to 50%, and using the output of the GRT to switch a SEL (Select) instruction for each of the .PGain, .IGain, and .DGain parameters. Alternatively, a simple routine of Ladder Logic or Structured Text could also perform this gain swapping. The PIDE instruction is designed to bumplessly start using the new gains without requiring any special reset logic.

Important User Information

Solid state equipment has operational characteristics differing from those of electromechanical equipment. Safety Guidelines for the Application, Installation and Maintenance of Solid State Controls (Publication SGI-1.1 available from your local Rockwell Automation sales office or online at <http://www.ab.com/manuals/gi>) describes some important differences between solid state equipment and hard-wired electromechanical devices. Because of this difference, and also because of the wide variety of uses for solid state equipment, all persons responsible for applying this equipment must satisfy themselves that each intended application of this equipment is acceptable.

In no event will Rockwell Automation, Inc. be responsible or liable for indirect or consequential damages resulting from the use or application of this equipment.

The examples and diagrams in this document are included solely for illustrative purposes. Because of the many variables and requirements associated with any particular installation, Rockwell Automation, Inc. cannot assume responsibility or liability for actual use based on the examples and diagrams.

No patent liability is assumed by Rockwell Automation, Inc. with respect to use of information, circuits, equipment, or software described in this document.

www.rockwellautomation.com

Corporate Headquarters

Rockwell Automation, 777 East Wisconsin Avenue, Suite 1400, Milwaukee, WI, 53202-5302 USA, Tel: (1) 414.212.5200, Fax: (1) 414.212.5201

Headquarters for Allen-Bradley Products, Rockwell Software Products and Global Manufacturing Solutions

Americas: Rockwell Automation, 1201 South Second Street, Milwaukee, WI 53204-2496 USA, Tel: (1) 414.382.2000, Fax: (1) 414.382.4444

Europe/Middle East/Africa: Rockwell Automation SA/NV, Vorstlaan/Boulevard du Souverain 36, 1170 Brussels, Belgium, Tel: (32) 2 663 0600, Fax: (32) 2 663 0640

Asia Pacific: Rockwell Automation, 27/F Citicorp Centre, 18 Whitfield Road, Causeway Bay, Hong Kong, Tel: (852) 2887 4788, Fax: (852) 2508 1846

Headquarters for Dodge and Reliance Electric Products

Americas: Rockwell Automation, 6040 Ponders Court, Greenville, SC 29615-4617 USA, Tel: (1) 864.297.4800, Fax: (1) 864.281.2433

Europe/Middle East/Africa: Rockwell Automation, Brühlstraße 22, D-74834 Elztal-Dallau, Germany, Tel: (49) 6261 9410, Fax: (49) 6261 17741

Asia Pacific: Rockwell Automation, 55 Newton Road, #11-01/02 Revenue House, Singapore 307987, Tel: (65) 6356-9077, Fax: (65) 6356-9011