Breaking from Tradition:

Model Based Control vs. PID

AUTHORS:

STEVE NIXON AND MEGAN KORNOWA-WEICHEL

ADVANCED CONTROL SOLUTIONS, INC.

8750 RESOURCE PARK DRIVE

SYLVANIA, OHIO 43560

WHY MODEL-BASED NOT PID?

New ACSI model-based controllers are outperforming PID controllers in glass applications. The ACSI-integrated process controller quickly responds to process disturbances, minimizes job change time and reacts quickly to stabilize temperature variations. The benefits to you are lower production costs, improved product quality, and ultimately increased profits.

HOW CAN I IMPROVE PROCESS CONTROL?

Controlling glass temperature is key to achieving optimum glass viscosity and gob weight. Even the smallest temperature variations can negatively impact the quality of the finished product and result in lost production time. If we can model zone temperatures and reduce job change time, we can achieve tighter control.

In order to understand the process, we must first know the basics. The *Process Variable* (PV) is the property being measured by the control system, such as temperature, pressure, or level measurements. The *Setpoint* (SP) is the desired value for the PV. The *Output* is the action the PID loop takes to correct an error, such as changing valve positions, damper positions, charger speeds, or motor speeds.

Benefits of Model-Based Control

- Reduced process settling time on startups or process changes
- Increased product consistency and quality
- Reduced energy use
- Operation closer to specifications
- Simplified operation and maintenance

The ACSI model-based controller effectively controls melter and forehearth temperatures by modeling the existing process. The controller creates models for each control/process variable and feed forward input. These ideal models allow the system to anticipate changes needed to maintain consistent glass temperature.

What is the Difference Between PID and Model-Based Control?

Traditional PID Control relies on an Engineer determining the proper values to be used for the Proportional (P), Integral (I), and Derivative (D) constants. The engineer must then continuously evaluate the process and tune the constants. Model-Based Controllers are able to learn how a process responds to changes, and in turn, they can automatically make the tuning adjustments that would traditionally require an engineer. The table below illustrates key differences between these two types of control.

PID Control	Model-Based Control
Does not recognize dead time	Recognizes and manages dead time
Requires manual control	Adapts automatically to process changes
Uses feedback only	Uses feed forward as well as feedback

Consider the following everyday example to compare and contrast PID and Model-Based Control.

You have a very delicate recipe that requires the liquid to remain at 90°. The PID controller may be compared to a new trainee, and the Model-Based controller can be compared to an experienced chef. Each gas stove alerts the chef when they exceed the temperature limit.

If the trainee follows the process used by a PID controller, he pours in the liquid and begins to turn up the gas on the stove. He continues to turn up the gas until he receives feedback that the temperature has exceeded 90°. Because he does not understand the concept of dead time (the time before a change is observed) he overshoots the desired temperature. He then reacts to the situation and manually adjusts the gas higher and lower while the temperature oscillates, moving above and below the setpoint. He eventually reaches 90°.

In the mean time, the experienced chef realizes that there will be a period of time between when he turns up the gas and the time the change is measured by a thermometer. He accounts for the dead time and prevents oscillation, arriving at 90° with ease.

Now the recipe calls for the temperature to be raised to 100°. The trainee has tuned his controller to maintain 90°; therefore, he will have to slowly raise the temperature in order to avoid overshoot. The chef has years of experience and has learned how the stove responds. He knows exactly how much heat must be added to raise the temperature 10°; therefore, he arrives there much faster.

The next step is to add more water to the pot. The trainee will again guess how fast the water should be added and manually adjust the amount of gas. The chef will take into account feed forward inputs, such as the temperature of the water being added, density of the water, and ambient temperature in the room to predict exactly how quickly he can add the water and how much additional heat is needed. He will always be able to automatically adapt to changes in the rate at which he pours or the ambient temperature and still maintain 100°.

The experience of the chef allows him to predict the behavior of his stove and the contents in the pot. He can quickly and effectively finish his recipe, and in turn, the restaurant is able to serve customers in a timely manner. In much the same way, effective processes allow glass plants to run with fewer upsets and less lost production time.

HOW IS THIS APPLIED TO GLASS MANUFACTURING?

The everyday situation described above can be translated to glass manufacturing using the example of a forehearth zone.

During installation an ACSI engineer uses a detailed procedure to "teach" the controller how to model the process. For example, the controller learns that for every 10% it puts in, a certain reaction occurs. This information is used to take the best series of actions to control the temperature. The better the controller is "trained," the better the model, and the more effectively the controller will manage the temperature. The controller continues to learn and refine itself in order to have the most reliable "roadmap" when the next change is required. It is then able to understand the process, adapt to changing conditions or production rate changes, and predict the effect a certain situation will have on the process all without requiring an engineer to tune the loop.

Managing Dead Time

PID controllers do not understand dead time and cannot take it into account when controlling temperature. Often, the result is oscillation, where the temperature overshoots the setpoint, corrects, goes below setpoint, corrects, and goes back above. Model-based control does understand dead time and takes it into account when making adjustments. During the "learning process," a model-based controller learns what the typical dead time is and uses that knowledge to know how long to wait before continuing with adjustments.

Feedback vs. Feed Forward

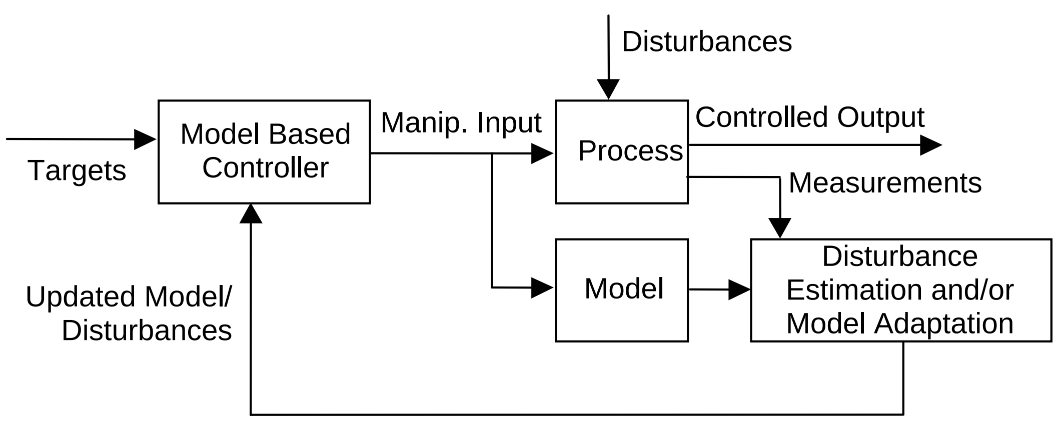
PID controllers use *feedback* to control glass temperature, where model-based controllers use *both feedback and feed forward* input. Using the example of a forehearth, a PID Control system will have sensors at the exit of each forehearth zone. Each sensor *feeds* information *back* to the PID controller, which then adjusts the heating or cooling to bring the temperature back to setpoint. Because glass has already been flowing through the forehearth, the system must always try to catch up with the desired temperature. This process has a long recovery time during which production values decrease. With PID the error is not detected until the glass reaches the zone exit. At this point, a problem already exists, and the controller must try to catch up to the correct temperature. In addition, the engineer must learn how the system reacts to different conditions and tune the loop accordingly.

In contrast, Model-Based Control can use both feedback and feed forward input to predict what will happen and make the adjustments needed to remain at setpoint. Feed forward provides the controller with information that allows it to anticipate the changes that will be needed to maintain consistent glass temperature. The controller can look at the incoming temperature as feed forward and analyze how it will affect the process. If it knows the temperature will be too hot when it gets to the zone entrance, it can reduce the heating to avoid the error. Instead of waiting for feedback from an instrument, model-based control analyzes the feed forward input and makes the adjustment before receiving the feedback.

Managing Job Change Time

In order to maximize profit potential, it is essential to minimize job change time. New glass temperature setpoints must be achieved as quickly as possible with minimum overshoot. It is difficult to achieve both objectives simultaneously with a standard PID controller. One of two scenarios is likely to occur. The first scenario is quickly raising the glass temperature. In this scenario, the temperature often overshoots the optimum SP and must be adjusted back to SP. The second option is to gradually raise the temperature, which requires a long period of time. Both scenarios typically require several hours to stabilize the temperature during which production operates at less than optimal parameters.

With model-based control, once the optimum process is modeled, the controller can predict what actions are needed to reach setpoint quickly without overshoot.



Model-Based Control	PID Control
Models the existing process, learns, adapts, and predicts.	Works on error from setpoint. Requires an operator to learn what action needs to be taken, take a reading, then act.
Models the existing process by creating models for each control/PV and feed forward input. The ideal model allows the system to anticipate changes needed to maintain consistent glass temperature. First, the ACSI engineer “teaches the controller while onsite at startup. It then continues to learn.	A sensor at the outlet of each forehearth zone measures glass temperature as it exits the zone. The sensor then relays data to the PID controller, which adjusts heat to bring temperature back to SP. As glass travels through each chamber, the controller continues to play catchup. There is a long recovery time during which production values decrease.
Adapts to process and production rate changes automatically for better control without loop tuning. Models feed forward inputs and updates control actions to quickly stabilize temperature variation.	Requires the operator to react to readings and tune the loop.
The controller can predict control actions required to drive the glass temperature to SP quickly without overshoot. Continues to refine itself, so that it has the most efficient “roadmap” the next time a change is needed.	Glass temperature can be raised quickly, but the temperature overshoots the optimum SP and must be adjusted back to SP. Glass temperature is achieved with gradual rise, which requires a long time to reach SP.
Calculates dead time and takes it into account when making adjustments.	Does not understand dead time.
Can look at incoming temperature as feed forward and analyze how it will affect the process. For example, if the glass temperature will be too hot when it arrives, it can back off and avoid the error.	Without feed forward, the error is not detected until the glass reaches the zone exit. At this point, a problem already exists.