Bang-Bang Controllers: Two Ways to Stop a Switch from Chattering

Roy Ratcliffe[a]

[a]https://github.com/royratcliffe

Published in *Roy's Code Chronicles*, 2026

A bang-bang controller has just two states: on and off. That sounds pretty basic, but there are plenty of applications where you actually do need just that (a heater, a pump, a relay, etc.). The real engineering challenge is in working out when you want to change state and when you want to resist changing it.

To switch or not to switch, that's the question. The answer isn't always clear though.

This article looks at two small Structured Text function blocks; one adds a quiet band around the switching value and the other imposes a minimum time between switches.

CODESYS | Structured Text | Control

A controller with no middle gear

Imagine controlling a room heater. A proportional controller could ask for half power, then a quarter, then a bit more. A bang-bang controller can't do that. It can only go full heat or nothing at all. On or off, there's no in between. The controller's name is actually a description of the controller's action: it bangs on and it bangs off.

This simplicity works when you've got a binary actuator (i.e. just a relay) or when the variable output is simply not worth the cost. A naive control rule such as 'heat when below 20°C, stop when above 20°C' has a drawback though. A noisy reading (even a minor one) can easily cross the 20°C threshold repeatedly resulting in relay chattering. This chattering will of course wear the contacts and can lead to a very noisy and unstable system.

The cure is memory (specifically 'output-state hysteresis'). A bang-bang controller will store its previous output so that the current input alone doesn't necessarily dictate the answer. 'Hysteresis' means the output is a function of history as well as the present. It's from the Greek 'hysteros' (later, after) so it's a description of the output being a function of the past as well as the present.

Value hysteresis: make a quiet band. Use two thresholds rather than one. For a heater, turn it on when the temp is below 18°C and turn it off when it's above 22°C. In between those two values just carry on as you were.

```
IF lrInputValue < lrHysteresisValueToOn THEN
    xOutput := TRUE;
ELSIF lrInputValue > lrHysteresisValueToOff THEN
    xOutput := FALSE;
END_IF;
```

The four degree gap is the hysteresis band. As the temperature moves through that band the controller doesn't change its mind. If the heater was on it will stay on until the upper threshold; if it was off it will stay off until the lower threshold. The same temperature within the band can therefore be either output state. That is not ambiguity, that is a deliberate choice.

Comparisons are strict. At exactly either threshold the output stays in its previous state as it does within the band. That gives a clean on-off rule at the threshold and prevents an unintended switch due to equality.

This is the classic Schmitt trigger[1] idea (in software form) i.e. separate the switch-on and switch-off points so that noise very close to one point won't cause rapid switch back and forth. It works very well when the input itself requires a tolerance band.

Time hysteresis: make the actuator wait. A different issue arises when an actuator requires a time delay between starts and stops. A compressor for instance may require a minimum off time before it can be restarted. A time-based controller can use a single separation value but would require a minimum residence time in each output state:

```
(*
 * Scan the time-to-on and time-to-off timers during each cycle.
 * Time-to-on runs while the output is off.
 * Conversely, time-to-off runs while the output is on.
 *)
tonTimeToOn(
    IN := NOT xOutput,
    PT := UDINT_TO_TIME(udiTimeToOnMilliseconds)
);
tonTimeToOff(
    IN := xOutput,
    PT := UDINT_TO_TIME(udiTimeToOffMilliseconds)
);
```

[1] Schmitt triggers are the classic example of value hysteresis. They have separate switch-on and switch-off points which prevent noise very near a single threshold from causing the output to rapidly toggle.

```
(*
 * Decide whether to switch the output based on the timers and input value.
 * The output switches only if the corresponding timer has elapsed and the input value indicates a change.
 * Bang "on" if the on-timer has elapsed and the input value is below the separation threshold.
 * Bang "off" if the off-timer has elapsed and the input value is above or equal to the separation threshold.
 *)
IF NOT xOutput THEN
    IF tonTimeToOn.Q AND (lrInputValue < lrSeparationValue) THEN
        xOutput := TRUE;
    END_IF;
ELSIF tonTimeToOff.Q AND (lrInputValue >= lrSeparationValue) THEN
    xOutput := FALSE;
END_IF;
```

While the output is off the on-timer is running; while the output is on the off-timer is running. A request to change state only takes effect after the timer for that state has expired.

There's a very subtle but important point: the timers are measuring the duration that the output has been in its current state. It doesn't need the input to have remained on the other side of the separation value for the entire duration in order for the output to switch. The timer can in fact finish whilst the input is still requesting the current state; if the request subsequently changes the output can then switch immediately. It's anti-cycling rather than debounce.

Two different kinds of patience. Both techniques prevent unwanted switching, but solve different problems:

Table 1. Comparison of value and time hysteresis.

Technique	What it remembers	Best suited to
Value hysteresis	Which side of a threshold band the output last entered	Noisy measurements and tolerance around a setpoint
Time hysteresis	How long the output has remained in its current state	Minimum run-time or off-time requirements

They are not interchangeable. A wide value band does not in any way guarantee a minimum off time. A long dwell timer doesn't stop a noisy measurement from asking for a switch as soon as the timer has expired. Some systems will in fact require both; a value band to make the decisions and minimum run/off times to protect the equipment.

The compromise: a limit cycle

Neither version results in the measured value sitting perfectly at the set point. The heater will continue to add heat after it has switched on and the room will continue to warm up after it has switched off. There is a lag which carries the temperature past the switching point so the value goes up and down in a repeating pattern known as a limit cycle. The band and timer will shape that cycle but they won't eliminate it entirely.

This is the trade-off with a simple two state actuator. If you make the value band too narrow you're liable to get chatter due to noise. If you make it too wide the value will wander further. Increase the minimum run time and the actuator will switch less often, but the measured value will spend longer outside the desired range.

Conclusions

Two-position control is not a broken proportional (PID) control. It's just a practical alternative when the output is inherently on/off.

A value hysteresis controller will remember which side of the band you entered on, whereas a time hysteresis controller will limit how long the output stays at a given value.

Choose value hysteresis to dampen a noisy threshold and time hysteresis when you need a minimum time interval between transitions. Use both together if both of those constraints apply and of course retain your independent safety interlocks. Hysteresis is a control technique, not a safety system.