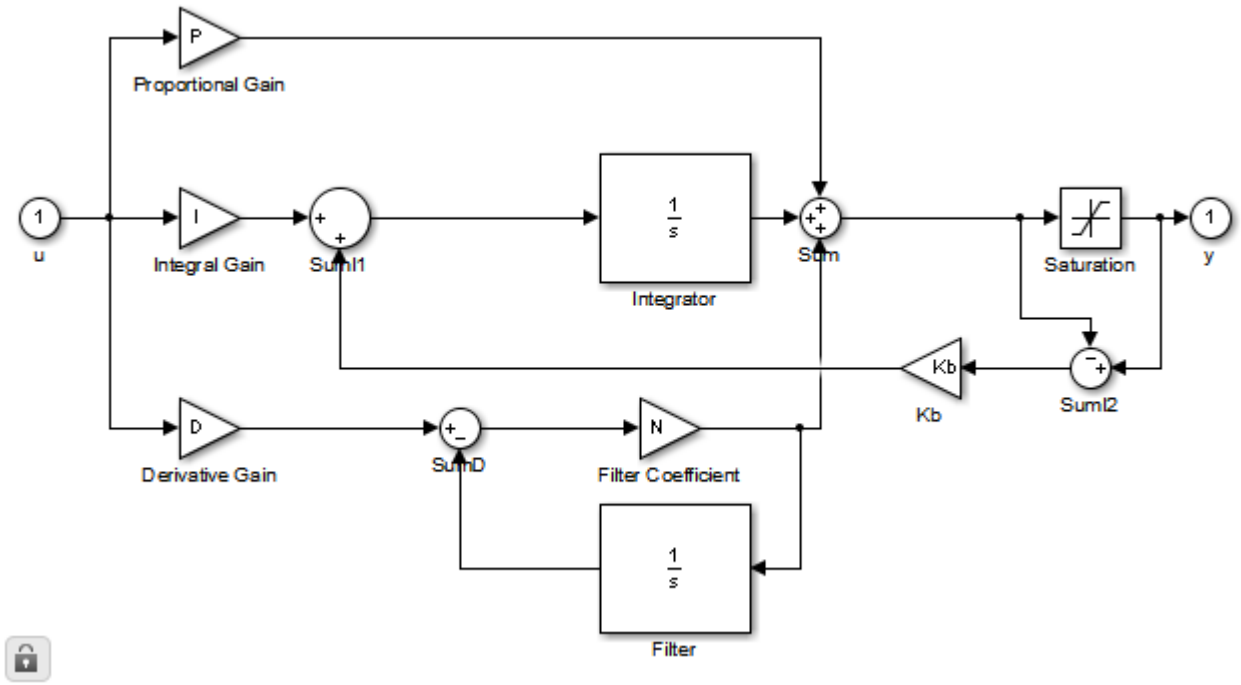
Simulink: PID Controller - difference between back-calculation and clamping for anti-windup?

Asked 7 years ago Active 4 years, 9 months ago Viewed 18k times

6

3

I need to implement an anti-windup (output limitation) for my PID controller. Simulink is offering two options: `back calculation` and `clamping` (documentation) which seem to deliver equal results. I know what back calculation is doing mathematically. It requires to define the back-calculation gain `Kb` . This gain is dependent on how long my controller is saturated, therefore it is actually a dynamic value (because I may have a high variation of saturation times). Do you see a way to control this value? (in this case it probably would be necessary to build my own PID Controller as shown in the documentation above or in the picture below.



Which brings me to the question, what is `clamping` actually doing? And what are other differences? Which one is faster, which one is more robust against stiff slopes? Does anybody has experiences using both?

matlab controller limit simulink integrator

asked Sep 6 '13 at 9:48

thewaywewalk

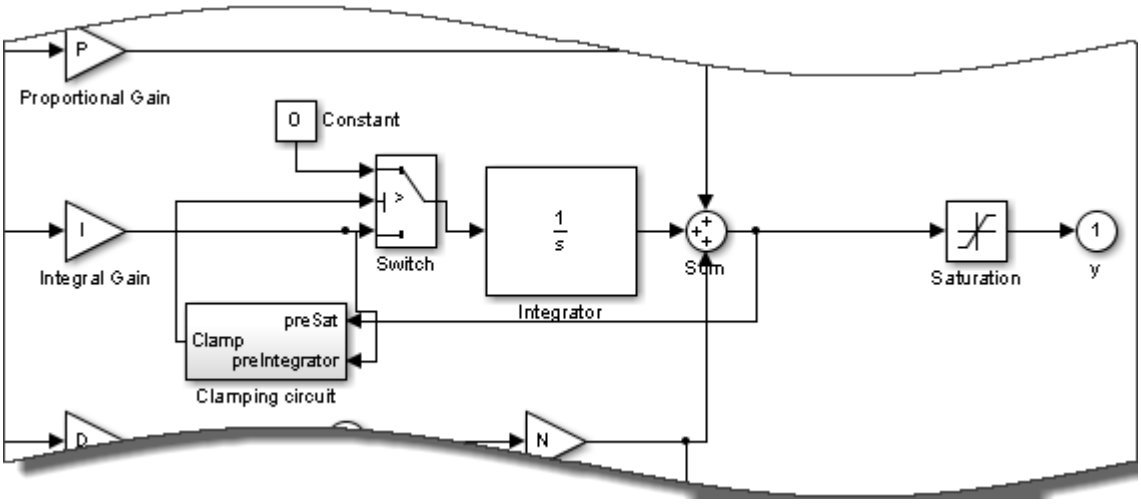
24.3k 8 62 107

Not sure if this fully answers the question, but the PID Controller documentation page, explains a bit more about clamping:

3

`clamping`

Stops integration when the sum of the block components exceeds the output limits and the integrator output and block input have the same sign. Resumes integration when the sum of the block components exceeds the output limits and the integrator output and block input have opposite sign. The integrator portion of the block is:



The clamping circuit implements the logic necessary to determine whether integration continues.

If you select the clamping option and look under the mask, you can probably see the details of the clamping circuit.

answered Sep 6 '13 at 9:57

am304

13.4k 2 17 34

3

Additionally to **am304's answer** there are some more things to consider.

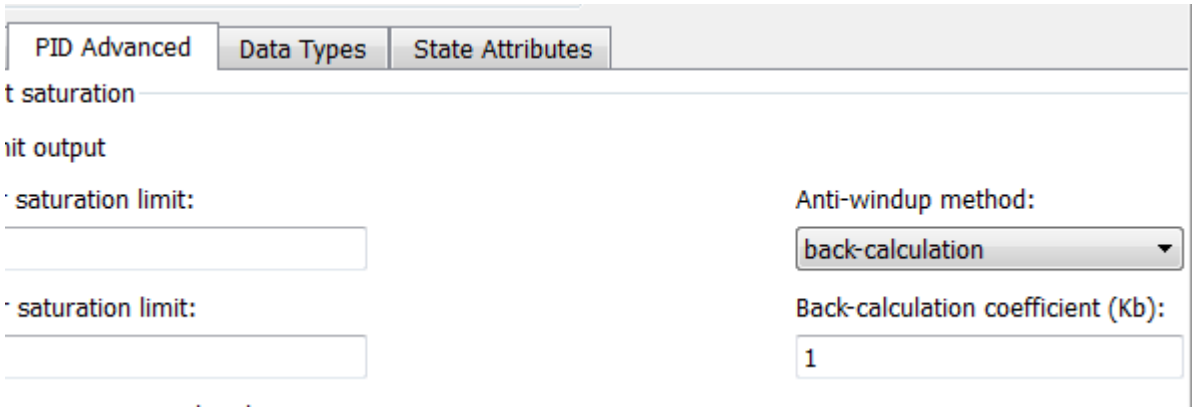
Clamping

Clamping will always work. It detects when there is integrator overflow and sets the integral path of the PID-controller to zero to avoid windup by using a simple switch.

input as well. This mechanism must me implemented manually.

Back Calculation

Back Calculation highly depends on the back calculation coefficient `Kb` . If you don't know how to actually calculate the parameter `Kb` *don't use back-calculation*. This method calculates the difference between the actual controller output and the saturated output and subtracts it from the I-Gain path, amplified by `Kb` . In most of cases the default value `Kb = 1` will lead to worse results than clamping, it is even possible that it has no effect at all. `Kb` should be calculated based on the sampling time or in case a D-Gain is involded, based on D- and I-Gain. Appropriate literatur should be consulted to calculate the coefficient. *Back calculation with* ***a properly set coeffient*** *enables better dynamics than clamping!*



edited May 23 '17 at 12:13
Community ♦
1 1

answered Nov 30 '15 at 9:24
thewaywewalk
24.3k 8 62 107