Advances in PID control for processes with constraints

Prof. Julio Elias Normey-Rico

Automation and Systems Department

Federal University of Santa Catarina - Brazil

IFAC WC 2020 – Berlin – July 2018

Process control

Manipulated variables (MV) are constrained

- Amplitude constraints
- Slew-rate constraints

⟺ **Actuator limitations**

Process variables (PV) need to be constrained

- Operational limits
- Safety

⟺ **Process operation**

Most used in industry: PID + Saturation blocks in MV + AWP

Use new ideas to improve PID AWP performance

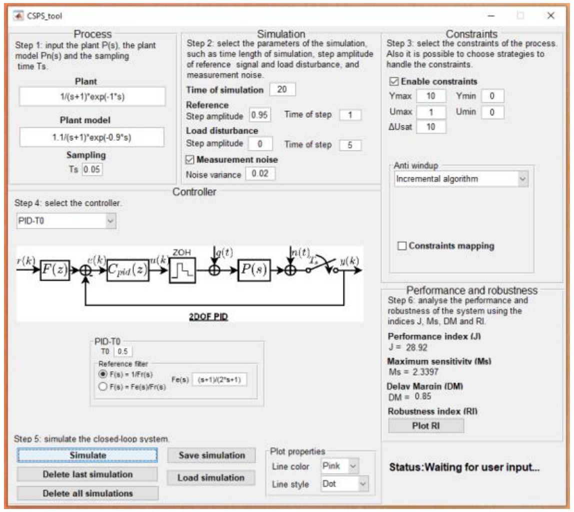
Agenda

1. Motivating example
2. PID control review and wind-up
3. PID control with AWP – Proposed solution
4. Conclusions

Simulation tool

http://rodolfoflesch.prof.ufsc.br/cspstool

CSPS - Constrained SISO Process Simulation tool



Motivating example

Ex1-Simple model with constraints

Simple model with delay

$$P(s) = \frac{e^{-s}}{s}$$

Several closed-loop simulations:

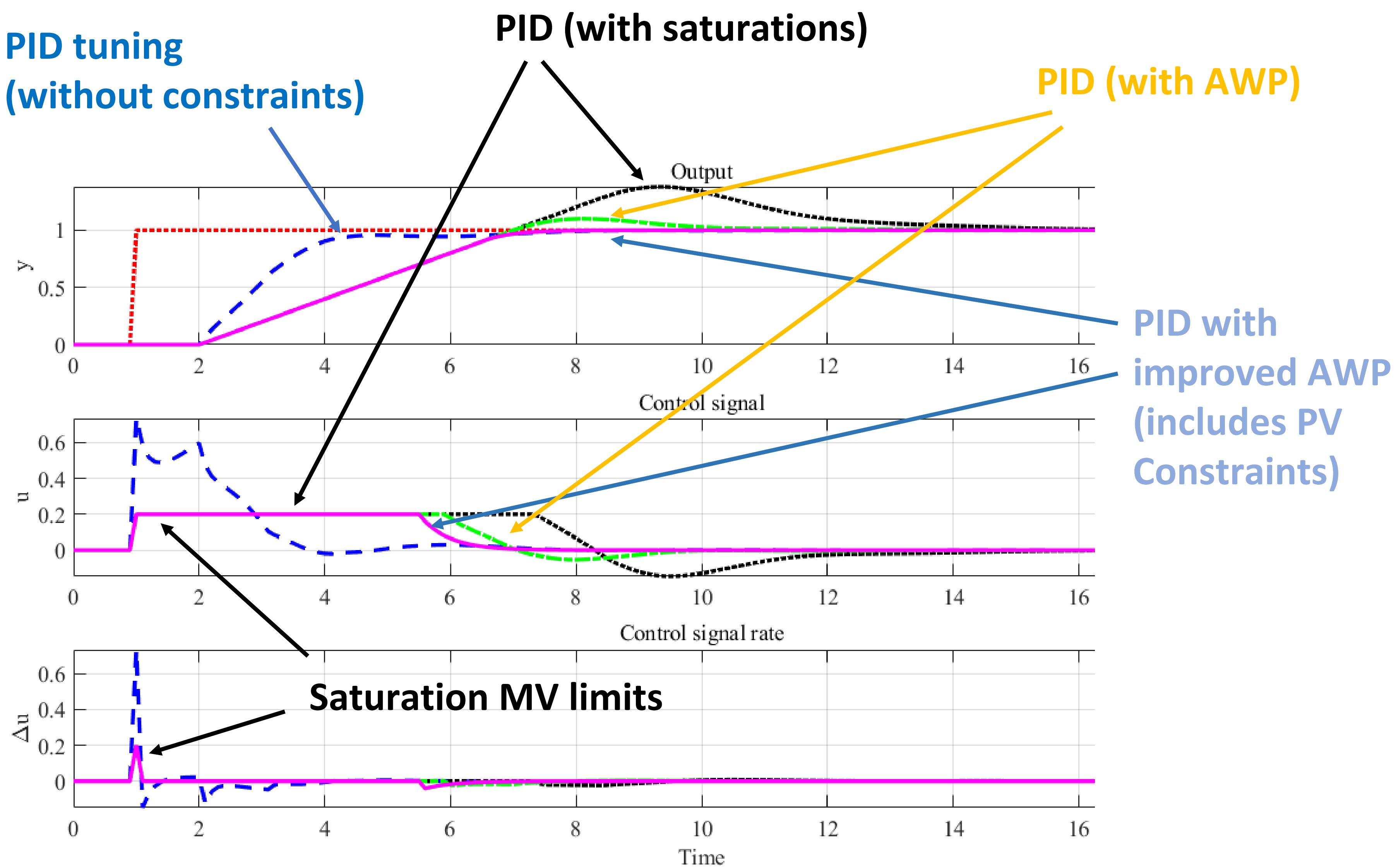
- PID tuning
- No constraints
- Constraints – no AWP
- Constraints – classical AWP
- Constraints – Improved AWP

$U_{max} = 0.2$ $U_{min} = -0.2$

$\Delta u_{max} = 0.2$

$Y_{max} = 1$

Ex1-Simple model with constraints



PID tuning

Unified PID formulation

Process models: FOPDT, IPDT, UFOPDT

$$\frac{K_p}{1+sT}e^{-Ls} \qquad \frac{K_p}{s}e^{-Ls} \qquad \frac{K_p}{sT-1}e^{-Ls}$$

PID form: Filtered derivative action

$$C(s) = K_p + \frac{K_i}{s} + \frac{K_d s}{1+s\alpha}$$

PID tuning: low frequency approximation of the Filtered Smith Predictor

Tuning parameter = closed-loop time constant To (trade-off)

↑ ***To*** ➡ ↑ **ROBUSTNESS** ↓ **PERFORMANCE**

Discrete controller

Discrete controller – proper sample-time selection *T_s*

Parallel form

$$C(z) = K_p + \frac{K_i T_s z}{z-1} + \frac{K_d}{\alpha + \frac{T_s z}{z-1}}$$

$$u(k) = u_p(k) + u_i(k) + u_d(k)$$

$$u_p(k) = K_p e(k),$$
$$u_i(k) = u_i(k-1) + K_i T_s e(k),$$
$$u_d(k) = \frac{K_d\left[e(k) - e(k-1)\right] + \alpha u_d(k-1)}{\alpha + T_s}$$

Discrete controller

Compact form

$$C(z) = \frac{n_0 + n_1 z^{-1} + n_2 z^{-2}}{(1 - z^{-1})(1 - \beta z^{-1})}$$

$$u(k) = (1+\beta)u(k-1) - \beta u(k-2) + n_0 e(k) + n_1 e(k-1) + n_2 e(k-2)$$

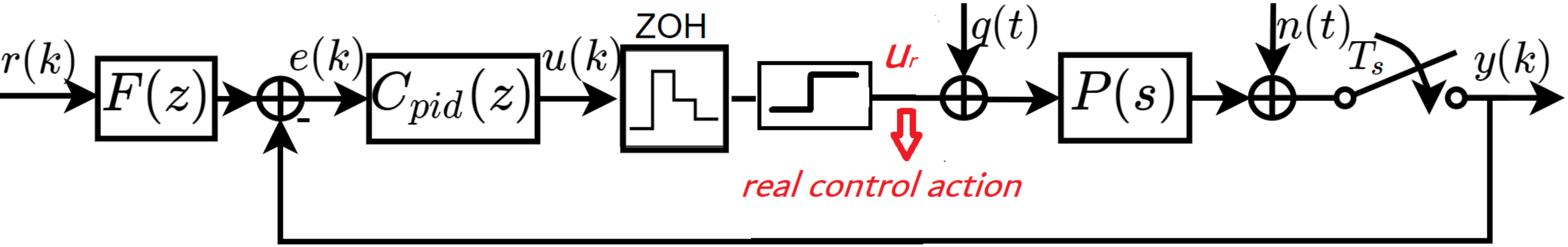
- β, n_i are computed with the given $K_p, K_i, K_d, \alpha, T_s$
- $\beta = 0$ ideal case (without derivative filter)

Incremental form: $\Delta u(k) = u(k) - u(k-1)$

$$\Delta u(k) = \beta \Delta u(k-1) + n_0 e(k) + n_1 e(k-1) + n_2 e(k-2)$$

PID control with AWP

Saturation problems

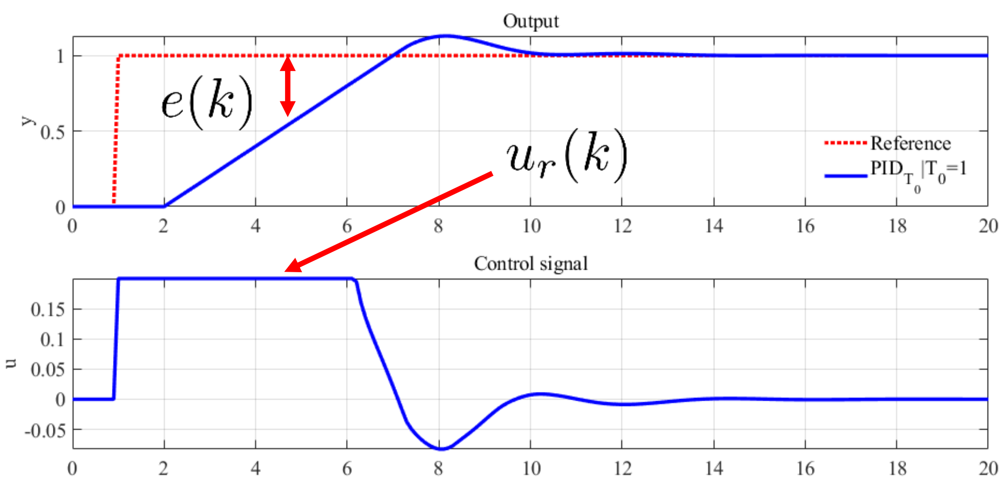


- Saturation when $u(k) \neq u_r(k)$
- Integral action increases

$$u_i(k) = u_i(k-1) + K_i T_s e(k)$$

$$e(k) > 0$$

$$u_i(k-1) > u_r(k)$$



$u_i(k)$ only decrease when *$e(k)$* changes its signal

But, as *$u_i(k) >> u_r(k)$,* needs some time to go out from saturation

Causes WIND UP

Anti Windup techniques

Incremental algorithm

$$\Delta u(k) = \beta \Delta u(k-1) + n_0 e(k) + n_1 e(k-1) + n_2 e(k-2)$$

Algorithm : Incremental algorithm

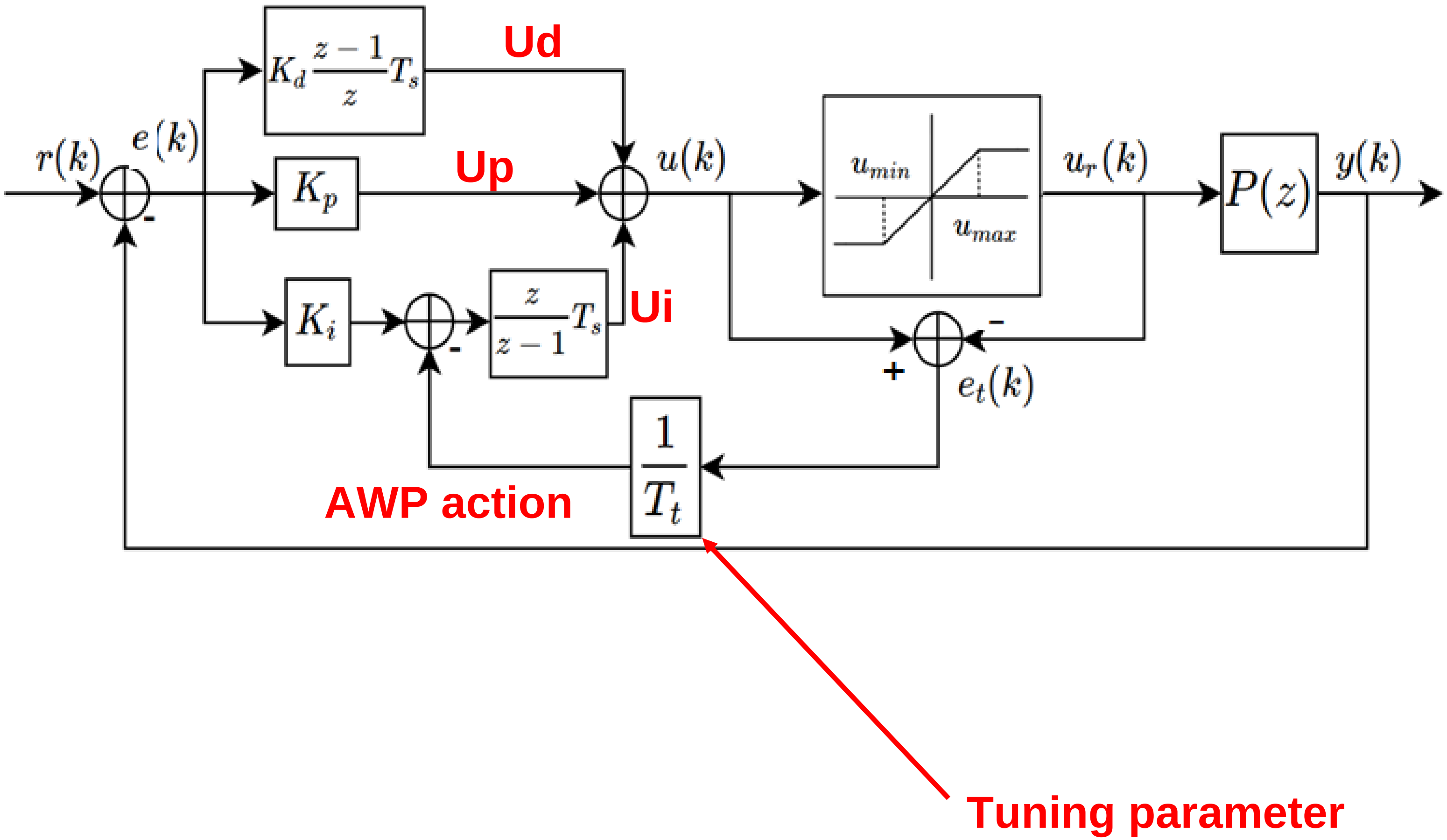
1. Compute $\Delta u(k)$
2. **if** $\Delta u(k) + u(k-1) \leq u_r(k)$ **then**
3. $\quad u(k) = \Delta u(k) + u(k-1)$
4. **else**
5. $\quad \Delta u(k) = u_r(k) - u(k-1)$
6. $\quad u(k) = u_r(k)$

No tuning parameter

code *update:* $\Delta u(k-1) \leftarrow \Delta u(k)$

Anti Windup techniques

Back-calculation



Anti Windup techniques

Error Recalculation (ER)

Recalculate the error signal at every sample to maintain the consistence between *u(k)* (computed) and $u_r(k)$ (applied)

$$u(k) = (1+\beta)u(k-1) - \beta u(k-2) + n_0 e(k) + n_1 e(k-1) + n_2 e(k-2)$$

$$u(k) > u_{max} \rightarrow u_r(k) = u_{max}$$

Consider:

$$u_r(k) = (1+\beta)u(k-1) - \beta u(k-2) + n_0 e(k) + n_1 e(k-1) + n_2 e(k-2)$$

?

$$\Rightarrow \quad e^*(k) = e(k) + \frac{u_r(k) - u(k)}{n_0}$$

No tuning parameter

Used in the code to *update the error:*

e(k-1)=e(k)*

Ex2- Anti Windup techniques

Comparing AWP methods

Simple example using the simulation tool

$$P(s) = \frac{1}{s+1} e^{-s};$$

- Simulation for a 0,95 step
- PID tuning with To=0,5
- Only MV saturation U_{mx}=1, U_{min} =0.

Improved AWP

Classical AWP works with simple saturation

$$\underline{U} \leq u(k) \leq \overline{U} \quad \forall k \geq 0,$$

Typical process constraints consider other variables

$$\begin{gathered} \underline{u} \leq u(k) - u(k-1) \leq \overline{u} \quad \forall k \geq 0, \\ \underline{y} \leq y(k) \leq \overline{y} \quad \forall k \geq 0. \end{gathered}$$

Solution: To map the constraints to the manipulated variable!

How? → Using some prediction ideas (MPC concepts)

Improved AWP

Mapping constraints (Max case)

$$\boxed{u(k) < U_{max}}$$

Direct

$$\Delta u(k) < \Delta u_{max}$$

$$\Delta u(k) = u(k) - u(k-1) < \Delta u_{max}$$

$$\boxed{u(k) < \Delta u_{max} + u(k-1)}$$

$y(k) < y_{max}$ Depends on past values of control action!

Using model $y(k) = f(u(k-i), y(k-i))$

u(k) will affect the future values of *y(k)* ⇒ Predictions!!!

Improved AWP

Using prediction ideas $$y(k+j) < y_{max} \quad \forall \quad j = 1 \ldots N_y$$

Horizon

$y(k-i)$
$u(k-i)$
$u(k+j)$

MODEL

Predictions

$y(k+d+j/t)$

Process delay

We do not know the future and we are going to apply only *u(k)* at $t = kT_s$

$$u(k+j) = u(k) \quad \forall j \quad \Rightarrow \quad y(k+d+j/k) = f(u(k), y(k-i), u(k-i))$$

The same idea in MPC , Control horizon $N_u = 1$

Improved AWP

Problem to be solved $y(k+j) < y_{max} \quad \forall \quad j = 1 \ldots N_y$

$$f(j, u(k), y(k-i), u(k-i)) < y_{max}$$

⇓

A set of inequations to obtain *u(k)* $u(k) < u_j \quad \forall j$

$$u(k) < min_j[u_j] = U_j \quad \forall j$$

Final constraint

$$u(k) < \min\{U_{max}; \Delta u_{max} + u(k-1); U_j\} = \overline{U}_{sat}$$

PID AWP control is then computed using this $\overline{U}_{sat}$ (Repeat for min condition)

Ex3- Improved AWP

Example: $$P(z) = \frac{bz^{-d}}{z-a} \Rightarrow y(k) = ay(k-1) + bu(k-d-1)$$

$$\begin{gathered} 0 \leq u(k) \leq 1 \quad \forall k \geq 0, \\ -0.1 \leq u(k) - u(k-1) \leq 0.1 \quad \forall k \geq 0, \\ -1.5 \leq y(k) \leq 1.2 \quad \forall k \geq 0. \end{gathered}$$

Max constraints

$$\left[\begin{array}{l} u(k) \leq 1 \\ \Delta u(k) \leq 0.1 \rightarrow u(k) \leq 0.1 + u(k-1) \\ y(k+d+j) \leq 1.2 \qquad j = 1, 2 \end{array}\right.$$

$$y(k+d) = \underbrace{a^d y(k) + ba^{d-1}u(k-d) + ... + bu(k-1)}_{\text{Known past values}}$$

Ex3- Improved AWP

$$y(k+d+j) = a^j y(k+d) + \underbrace{\boxed{(a^{j-1} + a^{j-2} + ... + 1)b}}_{K_j} u(k)$$

$$y(k+d+j) \leq 1.2 \Rightarrow u(k) \leq \frac{1.2 - a^j y(k+d)}{K_j} \qquad j = 1, 2$$

$$u(k) \leq \min\{1; 0.1 + u(k-1); \frac{1.2 - ay(k+d)}{K_j}; \frac{1.2 - a^2 y(k+d)}{K_j}\}$$

Past data

Min constraint

$$u(k) \geq \max\{0; -0.1 + u(k-1); \frac{-1.5 - ay(k+d)}{K_j}; \frac{-1.5 - a^2 y(k+d)}{K_j}\}$$

Ex3- Improved AWP

Implementation equations for max and min values:

$$y(k+d) = a^d y(k) + ba^{d-1}u(k-d) + \ldots + bu(k-1)$$

$$\overline{U}_{sat} = \min\{1; 0.1 + u(k-1); \frac{1.2 - ay(k+d)}{K_j}; \frac{1.2 - a^2 y(k+d)}{K_j}\}$$

$$\underline{U}_{sat} = \max\{0; -0.1 + u(k-1); \frac{-1.5 - ay(k+d)}{K_j}; \frac{-1.5 - a^2 y(k+d)}{K_j}\}$$

Apply the AWP scheme for PID control with $\overline{U}_{sat}$ $\underline{U}_{sat}$

Simulation Tool Example

Conclusions

Conclusions

- Using MPC ideas PID performance can be improved for process with constraints
- Improved AWP PID algorithms can be the solution in modern real-time distributed control systems for simple constrained systems
- Simulation tools are very important for analysis and design

Thanks!

For your attention

IFAC 2020 organizers

UFSC

CNPq
Conselho Nacional de Desenvolvimento Científico e Tecnológico

julio.normey@ufsc.br

References

Astrom, K.J. and Hagglund, T. (1995). PID controllers: theory, design, and tuning.

Camacho, E.F. and Bordons, C. (2013). Model Predictive Control. Springer, London.

da Silva, L.R., Flesch, R.C.C., and Normey-Rico, J.E. (2018). Analysis of anti-windup techniques in PID control of processes with measurement noise. In 3rd IFAC Conference on Advances in Proportional-Integral- Derivative Control, 948-953. Ghent, Belgium.

da Silva, L.R., Flesch, R.C.C., and Normey-Rico, J.E. (2019). Controlling industrial dead-time systems: When to use a PID or an advanced controller. ISA Transactions. doi:10.1016/j.isatra.2019.09.008.

Fertik, H.A. and Ross, C.W. (1967). Direct digital control algorithm with anti-windup feature. ISA Transactions, 6(4), 317.

Flesch, R.C., Normey-Rico, J.E., and Flesch, C.A. (2017). A unified anti-windup strategy for SISO discrete deadtime compensators. Control Engineering Practice, 69, 50-60.

Guzman, J.L., Astrom, K.J., Dormido, S., Hagglund, T., and Piguet, Y. (2006). Interactive learning modules for PID control. IFAC Proceedings Volumes, 39(6), 7-12.

Hippe, P. (2006). Windup in Control: Its Effects and Their Prevention. Springer Science & Business Media.

Normey-Rico, J.E. and Camacho, E.F. (2007). Control of Dead-time Processes. Springer, London.

Normey-Rico, J.E. and Guzman, J.L. (2013). Unified PID tuning approach for stable, integrative, and unstable dead-time processes. Industrial & Engineering Chemistry Research, 52, 16811-16819.