Relax Controller :

ℜelaxc

$$U_n = \Re_{(U_{n-1},\tau_{re})} + k_s(\tau_g \dot{e} + e)$$

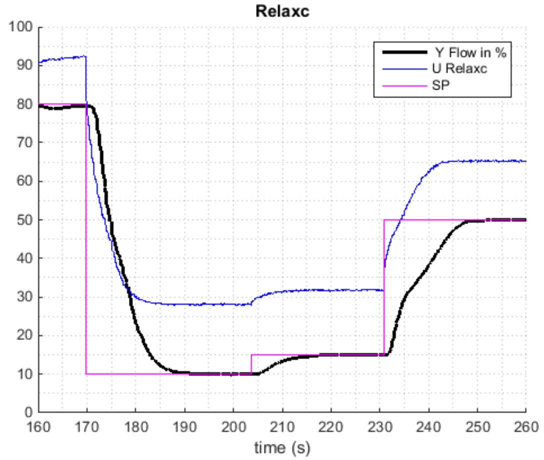
Relaxc Equation : ℜ the reactivity function, e=(yref-ym) with yref the reference trajectory and ym the process

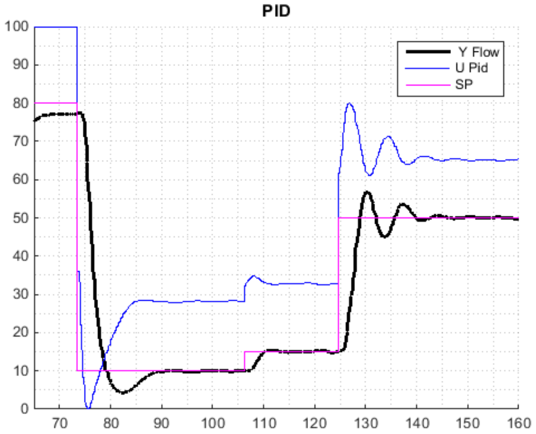
- **Change of paradigm:** In contrast with the other approaches, Relaxc is an implicit algebraic controller. The control U is evaluated by numerical convergence. This is why U does not depend of the mathematical structure of the process, so that we can apply it for tuning (almost) all the kind of processes, without modification or complex parameter rules .
- Thus by his mathematical nature, Relaxc allows to control **effortless and without overshoot** complex processes including Non minimal phase, small and large pure delay, unstable processes, variable static gain, strongly non-linearities, constraints on U, speed saturations, discontinuities, load disturbances, etc. **while minimizing the operational costs, the energy peaks of U with a more reliable control.**
- **Plug and control: Relaxc is available in** Unity Pro software (Schneider Electric). It can be embedded in small controllers with less than 10 arithmetic operations by cycle. **The tracking mode** functionality is available too and allows to switch in confidence your controller to Relaxc.
- **Example : Split range with pneumatic valves**

It is a strongly non-linear process (Cavitation phenomenon , 5000 liter/h). From scratch, in only 15 minutes after two open-loop responses, we estimates $\tau_{re} = \frac{t_d}{3} = 0.8s$, $\tau_g = 1.5s = 1/v$ and k_s =0.25. We obtain an adequate closed-loop response for different operating ranges. Notice how Relaxc follows the variation of the process dynamic with robustness. It is not the case for the PID (*Kp*=0.8 and Ti=2s). There are excessive oscillations and overshoots even if the response seems better for a step between 10% and 15%.

- **A Few words about Relaxc tuning.**

To work with Relaxc, you have to identify the pure delay and the "reactivity" time(1) of the process to build τ_{re}(named relax time too) and the max speed of your process (v). With this information, we deduce the time constant of the reference trajectory τ_g to get the best dynamic that the process can secure. Then , we just use the k_s pretty formula to compute it. $k_s = \frac{1}{v*\tau_g*(\frac{t_d}{\tau_{re}}+1)}$. This methodology seems close to that of Ziegler & Nichols. But Relaxc is **not a *PID*** and in particular all its parameters have a physical meaning, are **clearly discernable and easily identifiable** in the mathematical structure of Relaxc with respect to open-loop or impulse response or by auto-tuning identification. For this reason, the parameters of **Relaxc predict directly** the response time in closed-loop and the time response of the active disturbance rejection(2) without involving numerous trial-and-error tests or complicated calculus





More information about Relaxc:

https://www.researchgate.net/publication/335652545_Relaxc_vs_Real_processes

https://www.researchgate.net/publication/326649670_RelaxC_and_the_other_controllers

https://www.researchgate.net/publication/321918636_RelaxC_Controller_The_Ultimate_controller_The_origin

(1) "reactivity" time(1) of the process is not exactly the "apparent" time delay that we see in the literature.
(2) Relaxc provides a better rejection behavior than many other controllers.

❑ Appedge 18-22 rue d Arras 92000 Nanterre France
Email : Relaxc@appedge.com J.Masse 13/09/2019

Relaxc technology can be used in all sectors of the industry where processes control is needed: chemistry, aerospace, drone, pharmaceutical, automotive, petrochemical, electric or combustion or rocket engine, etc