Split parallel control
- a little known control structure

Krister Forsman*†, Mohammed Adlouni*, Sigurd Skogestad†

Perstorp Specialty Chemicals, 284 80 Perstorp, Sweden

†Norwegian Institute of Science and Technology (NTNU), 7491, Trondheim, Norway

Abstract: We describe a control structure that is commonly used in the process industry, e.g. chemical and petrochemical industries, for switching between manipulated variables (MVs), but which has received little attention in academia. It has one controller for each MV, typically PID-controllers, that control the same process value (y) but with different manipulated variables (u_i) and different setpoints (r_i). The scheme is sometimes called "separate controllers with different setpoints", but we suggest that a better name is "split-parallel control" (SPC), since the two controllers are placed in parallel in the block diagram, but the active control action is split between the two controllers, similar to in split-range control (SRC). SPC is an alternative to SRC, but it does have some advantages compared to SRC, including ease of implementation and the possibility to have different PID tunings for each MV. We also state some yet unresolved questions regarding the SPC structure, especially in regard to stability. Split-parallel control (SPC) uses setpoint separation to perform the switching, which is an advantage in some cases, for example, for bidirectional inventory control.

Keywords: Process control, multivariable systems, control structures, control architecture, decentralized control, optimizing control, split-range control, MV-MV-switching.

1. INTRODUCTION

Classical advanced control structures (architectures) are described in text books, e.g. in Marlin (2000), Smith and Corripio (2006), Skogestad and Postlethwaite (2005), as well in the recent review by Skogestad (2023). Some structures can be seen as "performance boosters" compared to PI-control, while others handle truly multivariable or non-linear control problems.

The processes that we study here have one controlled variable (CV; y), and (for simplicity) two manipulated variables (MVs; u_1, u_2). This means that we have an extra degree of freedom, which here is used to specify the sequence in which the MVs are used, that is, we want to use one MV at a time, in a predefined order, for example, based on using the cheapest MV first. A classical way of solving this problem is so-called split-range control (SRC), as described in Reyes-Lúa et al (2019). This is a case of what is known as MV-MV switching and two less known alternatives are "valve position control" and "separate controllers with different setpoints" (Skogestad, 2023).

In this paper, we focus on the last control architecture, which is commonly used in industry, but not much described in the literature. This structure has previously been called "separate controllers with different setpoints" (Forsman, 2005; King, 2011; Skogestad, 2023), but we suggest that it instead be called "**split-parallel control" (SPC)** because it is a parallel control structure which splits the control action and is an alternative to split-range control (SRC). Compared to SRC, the SPC structure has the advantage of being simpler to implement and allowing for separate controllers for each MV.

A typical application is when we two sources of heating in a house, for example, hot water (u_2) and more expensive electric heating (u_1). With split range control (SRC), we have one controller c and it is set up such that it first manipulates the hot water (u_2) to control the temperature (y), and when it saturates (at u_2=100%) (because it's a cold day), the controller switches to using electricity (u_1). With split parallel control (SPC), we have two controllers c_1 and c_2 with different setpoints, say $r_2=23^oC$ for hot water and $r_1=22^o$C for electricity. If it is not so cold, we only use hot water (u_2), so $u_1=0$ and the temperature is 23^{o}C. However, on a cold day, u_2 (hot water) will saturate at 100% and controller c_2 is not able to maintain the temperature at the setpoint $r_2=23^oC$. The temperature y will start dropping and when it approaches 22^{o}C, controller c_1 will activate and use the electric heat u_1 to keep y at the lower setpoint r_1=22^{o}C.

One potential disadvantage with SPC is that the switching is based on setpoint separation, which means the CV (y) must deviate from the original setpoint before the MV switching (between u_1 and u_2) occurs. In many cases, this is not a problem or it may even be an advantage, for example, for the application to bidirectional inventory control (Shinskey, 1981; Zotică et al, 2022). In other cases, the setpoint should remain constant, but if a temporary (dynamic) deviations from setpoint is acceptable, and it is possible to add an outer cascade loop to SPC as discussed in section 8.

2. SPLIT PARALLEL CONTROL (SPC)

A block diagram for SPC is shown in Figure 1. This control architecture is commonly used in industry. The two controllers should have different setpoints (SP; r_1, r_2), but use the same process or controlled variable (PV; CV), denoted y in the diagram.

In the block diagram the process is represented by a single block P with two process inputs (u_1, u_2) which are equal to the controller outputs. We could equally well have written it as two processes with individual inputs, and the PV as the sum of the outputs of those processes:

$$y = P_1(s)\, u_1 + P_2(s)\, u_2 + \text{disturbances}$$

If both controllers in Figure 1 have integral action, e.g. are PI-controllers, then the system would be **internally unstable** in a linear framework (Appendix A3). A simple way of understanding this is that there are infinitely many combinations of u_1 and u_2 that give the same y at steady state. However, in practical applications there are always controller output saturations (the valve positions (and thus u_i) are limited to 0% to 100%) or there are overrides from other controllers that limit the allowable range for u_i. The internal instability is actually an advantage as it drives one of the inputs towards its constraint limit (0% or 100%) and from equation (A2) we see this will happen faster when the setpoint difference $|r_1-r_2|$ is large This means that the two controllers in Figure 1 will be operated sequentially (except possibly for shorter periods), that is only one u_i is active at a time, and this makes the nonlinear system (with the saturations) internally stable.

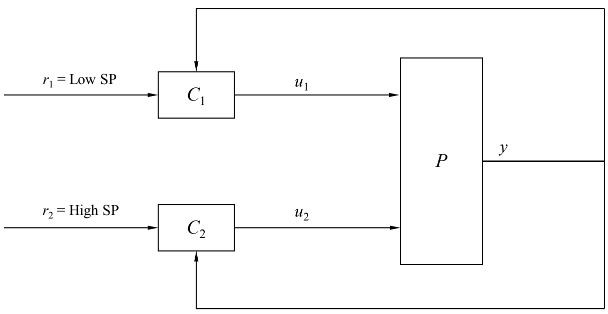

Fig. 1. Block diagram for split-parallel control (SPC). Except during transitions, only one of the two controllers (C_1, C_2) is active.

During normal operations, one of the controllers will be inactive, with u_1 or u_2 being not manipulable, either because one of the MVs (u_1 or u_2) is at its maximum or minimum saturation limit or because an override selector has made another controller take over one of the MVs. Obviously, the integral anti-windup functionality plays an important role in how well the structure works.

This SPC structure is not explained or analyzed in much detail in the literature. It is briefly mentioned e.g. in King (2011), Forsman (2005, 2016), and also in Reyes-Lúa et al. (2018, 2019,2020) and Skogestad (2023). In Jagtap et al. (2013) it appears as a part of override control without being devoted special attention. Shinskey (1981) describes it in the context of bidirectional inventory control where the difference in setpoints is used to take advantage of the buffer capacity. The economic advantage of using different setpoints for heating a residence is discussed by Reyes-Lua and Skogestad (2019).

In Balchen and Mumme (1988) there is a mention of "parallel control", which has similarities to SPC, but the two controllers have the same setpoint. To avoid internal instability, only one controller has integral action. Here, the objective of the extra MV u_2 is to improve the dynamic response and both controllers are always active, so there is no split between the manipulated variables.

3. BASIC PROPERTIES OF SPC

For now, we will assume that both controllers have integral action. We consider the case where MV saturation (0% - 100%) is the reason for switching and we assume that the two controllers are tuned so that the entire system is stable (relying on the MV saturation).

Example 1. As noted above, since the two controllers have different setpoints, one of them will drive its MV (say u_1) to a limit. At steady state, the PV (y) for the process will be equal to the setpoint (reference) of the controller (r_2) which has a non-saturating MV (u_2). Figure 2 illustrates this scenario. Here, the initial values were chosen so that the system was not in steady state. Initially, both controllers are active and are trying to drive the system to their setpoints of $r_1=50$ and r_2=52, but they are competing, so y remains in the middle at 51. At about time t=80 controller 1 (C_1) saturates at $u_1=0\%$, and controller 2 (C_2) drives the PV (y) to its setpoint (r_2=52). In this simulation both processes P_1 and P_2 were first-order processes with deadtime, having the same dynamics. The controllers were PI controllers with the same tuning parameters (see Appendix A1).

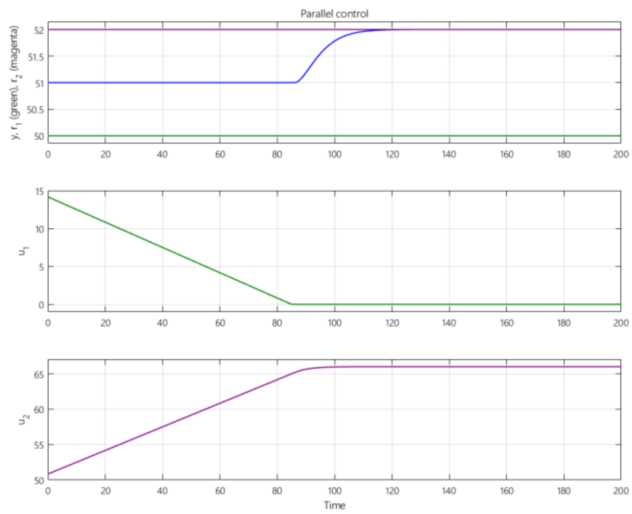

Fig. 2. Transient behavior of a SPC system (Example 1). Top graph: PV (y) for the process with SP (r_1, r_2) for the two controllers. Middle graph: Output of controller 1 (u_1). Bottom: Output of controller 2 (u_2).

4. APPLICATION: STEAM HEADER PRESSURE

Example 2. In this application there are two manipulated valves on individual pipes connected to a high-pressure steam header. One of the valves (V2, u_2) is a vent valve which allows steam to be vented to the atmosphere. The other (V1, u_1) sends high pressure steam through a turbine to the lower pressure header. The specifications for the control solution is that the high pressure header should normally be controlled to a setpoint of r_1=50 bar(a) with the vent valve completely closed. The vent valve should only open when there is an excess pressure, that is, when the pressure reaches r_2=54 bar.

Figure 3 shows a split-parallel control architecture that satisfies the specifications. PC2 in the figure is the "rescue controller" that is only active when venting is needed. It has a higher setpoint than PC1, which is the controller maintaining the pressure in the header when the plant is running normally. Most of the time we wish the output of PC2 (u_2) to be zero, since venting is not desired.

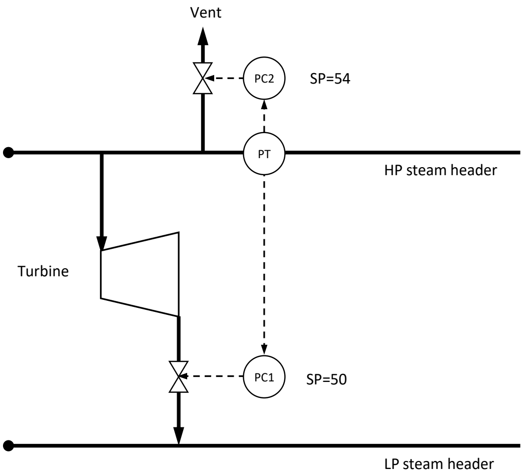

Fig. 3. Split-parallel control for pressure in steam header (Example 2).

5. APPLICATION: BI-DIRECTIONAL CONTROL

Example 3. Consider a chain of buffers (liquid inventories) connected serially by pipes. This is a simplistic representation of a plant with "linear" topology. If we wish to maximize the throughput of the plant, then the choice of master flow (throughput manipulator) depends on the location of the bottleneck. Using the control structure shown in Figure 4, we get automatic reconfiguration of level control – flow control pairing. Details are described in Zotică et al. (2022).

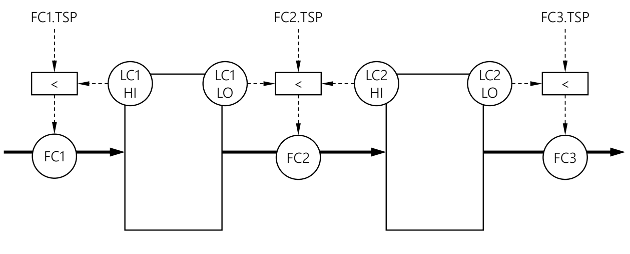

Fig. 4. Bi-directional control (Example 3). Each tank has two level controllers with different setpoints, and the selectors determine which one is active.

This is an example of a combination of split parallel controllers that switch between the MVs and minimum selectors that switch the pairing of MVs and CVs. Each tank has a level controller with high setpoint (LC1HI, LC2HI) and a level controller with low setpoint (LC1LO, LC2LO).

6. COMPARISON TO SPLIT-RANGE CONTROL

In a general, there are three alternative structures for MV switching (Skogestad, 2023), namely split-parallel control (SPC), split-range control (SRC) and valve position control. SRC is the oldest and most common structure. However, the less well-known SPC alternative studied in this paper, has some advantages:

- With SPC, you can tune the two controllers differently so that the parameters, like integral time, are fit for the subprocess manipulated with that controller. In SRC, one may indirectly achieve different controller gains for the two subprocesses by tuning the split location in the table, but it is not possible to have different integral times.
- When using SRC, the two valves are algebraically locked to each other. The operator cannot easily choose to run the valves independently, whereas this is immediately obvious how to do with SPC.
- In some applications it is an advantage to have two different steady states (setpoint separation). In the bidirectional control application example above, we could have used a single SRC for each level and combined those with selectors. However, the advantage of using SPC is that we automatically get varying steady state buffer levels which enables production to be continued when there is a new bottleneck, as explained in Zotică et al (2022).
- In SPC, a disturbance of short duration does not necessarily cause a switch to the other MV. There is a "grace period", especially if we have a large setpoint separation. In some applications this may be an advantage. This is illustrated in Figure 5. The smaller disturbance appearing at time t=10 is managed by controller C_1 without activating controller C_2.
- With SPC, for large disturbances or during startup, both manipulated variables (u_1 and u_2) may be used simultaneously during dynamic transients. This is illustrated in Figure 5 for the larger disturbance at time t=60, which causes C_2 to be active for a short time. This speeds up the response for y towards the new steady state. This will not happen with conventional split-range control which only can manipulate one MV at the time.

The latter is also a disadvantage with SPC because stability problems may arise during transients when the two parallel controllers are active simultaneously as the total loop gain increases. However, this is believed to be more a theoretical than a practical problem (see Section 7).

The SPC architecture obviously has the potential drawback that it depends on the process state, e.g. disturbances, which setpoint (r_1 or r_2) will be reached at steady state. However, this can be addressed by adding another "master" controller on top of the other two, as explained in Section 8 below.

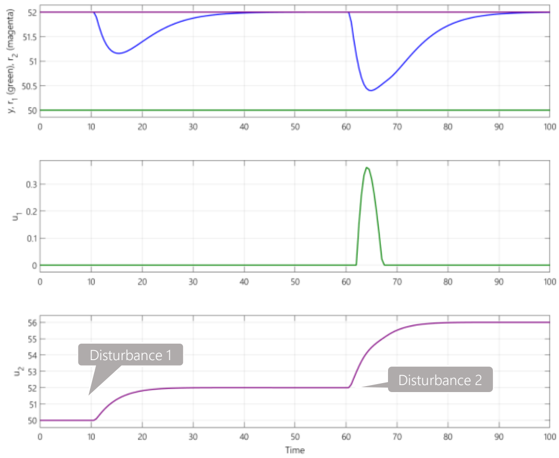

Fig. 5. SPC simulation illustrating grace period for small disturbance and fast response for large disturbance. Plot layout the same as in Fig 2 (Example 1).

7. STABILITY CHALLENGES

When studying SPC, a question posing itself immediately is: What are the conditions on system dynamics and controller tuning for the two controllers not to start interacting in such a way that the entire system becomes unstable?

The internal instability mentioned in Section 2 may result in situations which industrial practitioners often refer to as the controllers "fighting" each other. This can be clearly seen during the initial period in the simulation in Figure 2. Until time is about 80, both controllers C_1 and C_2 are active at the same time and the output y stays at 51, between the two setpoints of $r_1=50$ and $r_2=52$. During this period, the system is internally unstable with u_1 decreasing and u_2 increasing because of the integral action in the controllers (see Appendix A3). The initial value of 51 for y was chosen in the midpoint between r_1 and r_2 such y remains constant at 51 in spite of both u_1 and u_2 changing. However, when u_1 reaches saturation (0%), the fighting stops and loop 2 operates alone in a stable manner.

Another instability (which comes in addition to the internal instability) may occur if the controllers are tuned too aggressively. Assume that we are in a situation when both controllers are active at the same time and are "helping" each other by driving y in the same direction. This may occur if the setpoints r_1 and r_2 are close to each other and y is outside the setpoint range. In this case, the combined action of the parallel controllers may lead to instability with limit cycles (see Example 4 and Figure 7). We don't have a general theoretical analysis of when stability will be achieved, because stability analysis is difficult for constrained systems with switching. However, in terms of the unconstrained (linear) stability, a simple analysis for the case with two inputs and with $r_1=r_2$, says that if each individual controller-process system is tuned so that the gain margin is larger than 2, then it is unlikely that the combined action of both controllers makes the system become unstable (Appendix A2). If the gain margins are smaller than 2, then the analysis (Appendix A2) shows that we may get (linear) instability during dynamic transients, and simulations (Figures 6-8) show that this may initiate limit cycles (what a practitioner would call instability), depending on the initial state, magnitude of disturbances and amount of separation between the setpoints. This is a situation that practitioners want to avoid, so our recommendation is to always tune the controllers so that they have gain margin at least 2, and preferably larger.

Another theoretical challenge is to prove our conjecture of uniqueness of the steady state, that is, that the SPC system always ends up in the same final steady state for given disturbances, setpoints and MV limits, independent of the initial state.

Example 4: Figure 6 to Figure 8 show simulations of three scenarios for the process (P) studied in Example 1 (Fig 2). The two subprocesses (P_1 and P_2) have the same parameters (first-order with time delay), and C_1 and C_2 are both PI-controllers with identical tunings. Both individual control loops are tuned with a gain margin of only 1.31, corresponding to a sensitivity function peak value M_s of 4.9. This is surely an aggressive tuning, so we may expect stability problems if both controllers are active at the same time and driving the system in the same direction, which may happen if the setpoint difference is small.

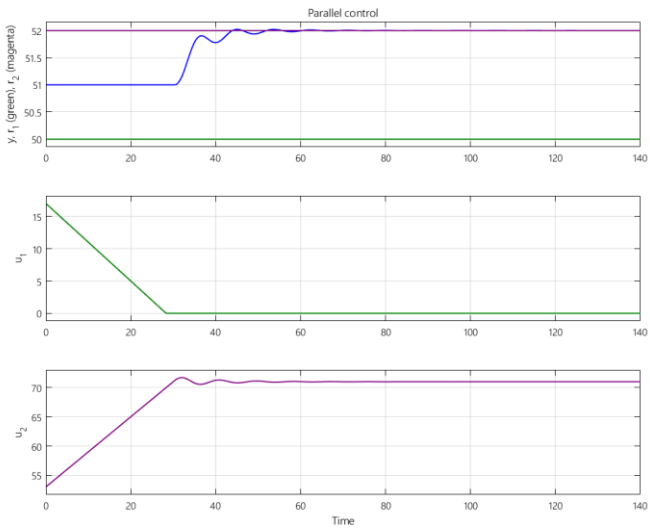

Fig. 6. Simulation scenario: r_1 = 50, r_2 = 52. (Example 4; similar to Fig. 2 in Example 1 but more aggressive PI tunings).

The difference between the scenarios is only the setpoint separation magnitude. Setpoint r_1 is 50 in all cases, but r_2 varies between the scenarios, as described by the figure captions. Linearly, when both controllers are active (in the first 80 minutes), the system is unstable (see Appendix A1), but this instability may die out when saturation is reached in loop 1 (Figures 6 and 7 with r_2=52 and r_2=51.95) because loop 2 by itself (with $c_1=0$) is stable.

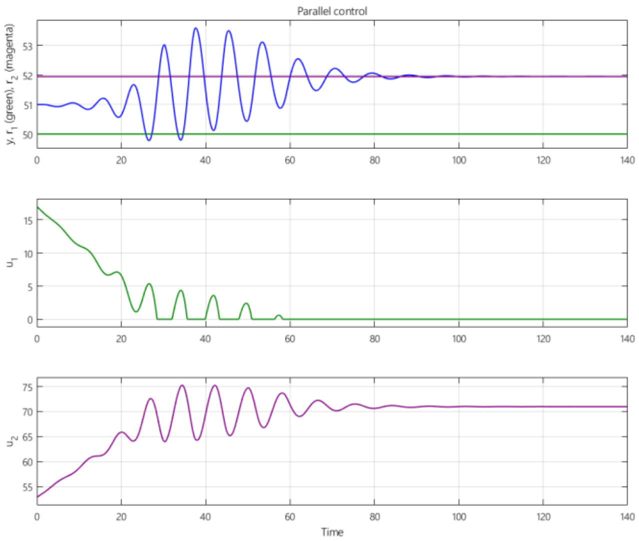

Fig. 7. Simulation scenario: r_1 = 50, r_2 = 51.95 (Example 4).

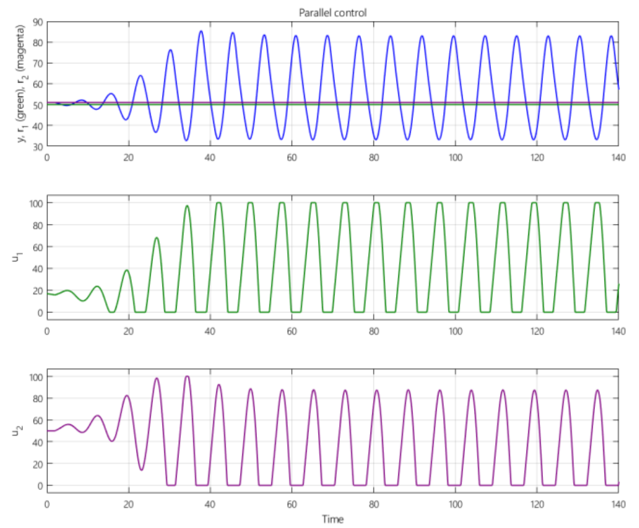

Fig. 8. Simulation scenario: r_1 = 50, r_2 = 51 (Example 4)

However, with the smallest setpoint difference (r_2=*51*, Fig. 8) the system settles into "stable limit cycles" which a practitioner would call "unstable". As can be seen, the stability of this system cannot be described using linear theory. Note that the scales of the vertical axes are different in the figures.

8. CASCADE CONTROL FOR UNIQUE STEADY STATE

Figure 9 shows an extension of the SPC scheme, where an extra controller C_0 has been added (Skogestad, 2023). This "master controller" gives the setpoint for C_1 and C_2, with a fixed bias Δr in between them. In this architecture the process value will converge to the setpoint of C_0 in steady state, if the total system is stable.

As discussed above, this may be desirable in some applications, but a disadvantage in others.

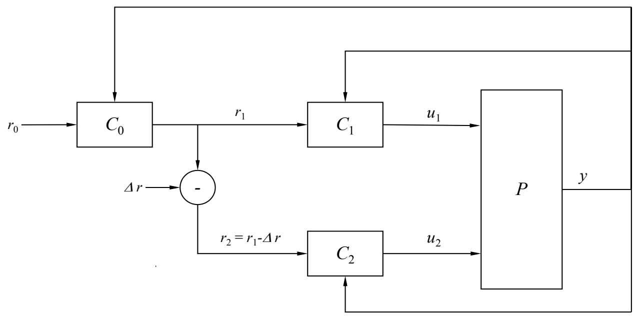

Fig. 9. SPC extended with a master controller C_0. The process value y will adhere to the setpoint r_0 in steady state, given that the controllers are tuned so that the entire system is stable.

10. CONCLUSION

Split-parallel control (SPC) is commonly used in industry for cases where MV-saturation makes it necessary to use two or more MVs (u_1, u_2) to control the CV (y) under all steady-state conditions. SPC is internally unstable when seen from a linear point of view, but this is actually an advantage as it drives one of MVs to its constraint limit. Compared to the more well-known split-range control, SPC has the advantage that it is simpler to implement and that one can have different tunings (c_1, c_2) for each MV. It requires the use of different setpoints (r_1, r_2) which may be an advantage or disadvantage, depending on the situation. Another advantage is that SPC may use more than one MV using dynamic transients, which may speed up the response (see disturbance 2 in Figure 5). However, as discussed in Section 7 this may cause problems with instability and limit cycles (Figure 8) if aggressive tunings are used for the individual controllers (c_1, c_2). Nevertheless, by using reasonable tunings this is not believed to be a problem in practice. A more careful theoretical study of such switched systems would be useful to answer these questions in more detail.

REFERENCES

J.G. Balchen and K.I. Mumme: *Process Control: Structures and Applications*. Kluwer, 1988

K. Forsman: *Implementation of advanced control in the process industry without the use of MPC*. DYCOPS conference, Trondheim, IFAC papers online, 49, pp 514–519. 2016.

K. Forsman: *Reglerteknik för processindustrin*. Student-litteratur, 2005. (In Swedish)

R. Jagtap, A.S. Pathak and N. Kaistha: Economic Plantwide Control of the Ethyl Benzene Process. AIChE Journal, Vol. 59, No. 6, pp 1996-2014. June 2013.

M. King: *Process Control. A Practical Approach*. Wiley, 2011.

T.E. Marlin: *Process Control*. 2nd edition. McGraw Hill, 2000.

A.Reyes-Lúa and S. Skogestad: Multiple-Input Single-Output Control for Extending the Steady-State Operating Range—Use of Controllers with Different Setpoints. Processes, 7, 941, 2019.

A. Reyes-Lúa and S. Skogestad: *Systematic Design of Active Constraint Switching Using Classical Advanced Control Structures.* Ind. Eng. Chem. Res., 59, pp 2229-2241, 2020.

A. Reyes-Lúa, C. Zotică, and S. Skogestad: *Optimal operation with changing active constraint regions using classical advanced control. IFAC* Adchem *Symposium*, China*, 2018*

A. Reyes-Lúa, C. Zotică, K. Forsman and S. Skogestad: *Systematic Design of Split-range Controllers*. 12th IFAC Symposium on Dynamics and Control of Process Systems, including Biosystems DYCOPS. Florianópolis, Brazil. 2019.

F.G. Shinskey: *Controlling Multivariable Processes.* Instrument Society of America, 1981.

S. Skogestad and I. Postlethwaite: *Multivariable Feedback Control: Analysis and Design.* 2nd edition. Wiley*,* 2005.

S. Skogestad, *Advanced control using decomposition and simple elements*. Annual Reviews in Control, vol. 56, Article 100903 (44 pages). 2023

C.A. Smith and A. Corripio: *Principles and Practice of Automatic Process Control*. 3rd Edition. Wiley, 2006

C. Zotică, K. Forsman and S. Skogestad: *A simple bidirectional inventory control structure with optimal use of intermediate storage*. Computers and Chemical Engineering, 159, 107677. 2022.

APPENDIX

A1. Parameters in simulations

Example 1:

$$P_1(s) = P_2(s) = \frac{\exp(-s)}{5s+1}, \quad C_1(s) = C_2(s) = 0.83\frac{5s+1}{5s}$$

Example 4:

$$P_1(s) = P_2(s) = \frac{\exp(-2s)}{5s+1}, \quad C_1(s) = C_2(s) = 3\frac{5s+1}{5s}$$

Anti-windup with tracking (e.g., Skogestad (2023)) was implemented on both controllers with tracking time constants $\tau_{T1} = 2$ and $\tau_{T2} = 5$.

A2. Stability analysis (for Section 7).

Assume that both parallel controllers are active so that the system is operating in the linear region (with no constraints). The closed-loop response of the linear system in Figure 1 is then

$$(A1) \quad y = \frac{1}{1+L_1+L_2}(L_1 r_1 + L_2 r_2)$$

Here $L_1=P_1C_1$ and $P_2=P_2C_2$. Linear stability is determined by the parallel loop transfer function $L = L_1 + L_2$. From the Bode stability condition we have that the system is closed-loop stable if and only if $|L(j\omega_{180})| < 1$ where ω_{180} is the frequency where $\angle L(j\omega_{180}) = -180^o$. Also note that $|L(j\omega_{180})| = \frac{1}{GM}$. *We assume for simplicity that the loop transfer functions have the same dynamics*. Then $L_1 = k_1 L_0(s), L_2 = k_2 L_0(s)$ and $L(s) = (k_1 + k_2)L_0(s)$ so all transfer functions have the same ω_{180}. Then $|L(j\omega_{180})| = |L_1(j\omega_{180})| + |L_2(j\omega_{180})| = \frac{1}{GM1} + \frac{1}{GM2}$ and a necessary and sufficient condition for linear stability is $\frac{1}{GM1} + \frac{1}{GM2} < 1$. If both subsystems have the same gain margin then this gives the necessary and sufficient stability condition GM1 = GM2 > 2. If this condition is satisfied then no instability is expected for the system in Figure 1.

This also means that if GM1=GM2 is less than 2 and the two loops are identical ($L_1=L_2$) then we will have linear instability (during dynamic transients with no input saturation). However, this does not necessarily mean that the instability results in "nonlinear" instability (that is, limit cycles) when constraints are encountered. This is clearly seen in Example 3 where GM=1.31 which is much less than 2. Thus, locally we have instability, but it does not necessarily lead to limit cycles as seen in the simulations in Figures 6 and 7.

A3. Internal instability (for Section 2).

The above stability analysis is a not related to the internal instability mentioned in Section 2. In fact, the internal instability (and non-uniqueness in the inputs) does not appear in equation (A1). To see the internal instability, we need to consider the transfer function to the inputs. For example, the transfer function from the setpoints to u_1 is given by (when operating in the linear region with no saturation):

$$u_1 = C_1(r_1 - y) = C_1(r_1 - \frac{1}{1+L_1+L_2}(L_1 r_1 + L_2 r_2) = \frac{C_1}{1+L_1+L_2}(r_1 + L_1 r_1 + L_2 r_1 - L_1 r_1 - L_2 r_2)$$

That is, we get

$$(A2) \quad u_1 = \frac{C_1}{1+L_1+L_2}(r_1 + L_2(r_1 - r_2))$$

We get a similar expression for u_2 (but with indices 1 and 2 interchanged). The internal instability appears in the last term in equation (A2) where there is an (unstable) integrator in L_2 which is not stabilized by the feedback. The internal instability drives one of the inputs towards its constraint limit (0% or 100%) and from equation (A2) we see this will happen faster when the setpoint difference $|r_1$-$r_2|$ is large. Which constraint that is encountered depends on the value of the disturbances.