Advanced control using simple elements - An important and challenging research area

Sigurd Skogestad*

Department of Chemical Engineering, Norwegian University of Science and Technology (NTNU), 7491, Trondheim, Norway

Abstract

The paper describes the most important standard control elements used in industry to design advanced control systems, including cascade, ratio, feedforward, decoupling, selectors, split range and many more. These industrial methods are in this paper called "advanced regulatory control" (ARC). Many examples are given in the paper, with focus on process control. The shortcomings of model-based optimization methods (e.g., MPC) are highlighted, because many academics think that MPC can solve all problems. They therefore see no need to study the industrial ARC methods which they regard as ad-hoc, difficult to understand and outdated. This paper makes the point that this is not true. With the knowledge of control elements presented in this paper, it should be possible to understand most industrial solutions and also to propose alternatives and improvements. The academic community is challenged to start teaching these methods and to focus a lot more research on developing theory and improving design methods.

Keywords: control structure design, feedforward control, cascade control, PID control, selective control, override control, time scale separation, decentralized control,

*Corresponding author
Email address: Sigurd.Skogestad@ntnu.no (Sigurd Skogestad)

Contents

1. Introduction

Today, the process industry makes use of two main approaches for advanced control:

- Advanced regulatory control (ARC): Use of simple standard control elements.
- Model predictive control (MPC): On-line optimizing control using a dynamic process model.

This paper focuses on "advanced regulatory control" or more generally on how one maay put together simple "classical" control elements to control real complex multivariable nonlinear constrained processes. In addition to give an overview of these elements, the objective of the paper is to point out the need for academia to significantly increase teaching and research in this important area.

The background and focus of the paper is on process control (including thermal power and bioprocesses), but most of" the "advanced" control elements presented in this paper are used by engineers in other application areas, including automotive, robotics, manufacturing, marine, aerospace, power, medical, and agriculture. Of course, "advanced" is a relative term, but at least for engineers in the process industry it is any control scheme or element that comes in addition to the basic single-input single-output feedback PID loop. MPC is also discussed in some detail, and this is mainly to demonstrate that, even if a model is available, MPC should not replace the simple control elements; rather it should be a complement and addition to the engineer's toolbox.

Process control started developing as a discipline around 1920. An important reason for the introduction of automatic control was the appearance of large-scale continuous processes (including ammonia, refining and petrochemical plants). Initially, these processes where controlled manually (with one operator for each valve) but this soon became impractical. The first automatic controllers were on-off feedback controllers, but these had the disadvantage that they generated oscillations. Therefore, during the 1920s, the process industry started using continuous feedback controllers based on proportional action. However, there was a problem with steady-state offset, and one needed to manually update the bias term of the proportional controller. To deal with this, methods for "automatic reset" of the bias were introduced, which later became the integral mode. For some processes there was also need for some "preact" (derivative) action. Minorsky introduced a three-term PID controller for steering of ships already in 1922, but according to Bennett (1988) this development was not known in the process industries. John Ziegler says in an interview with Blickley (1990) that Foxboro came out with the first standard proportional plus reset (PI) controller (Model 40) in about 1934-35. It was mainly used for flow control in the petroleum industry. Taylor Instrument Company came out with a similar product in 1936. In 1939, Taylor introduced the first general purpose three-term PID controller (Model 100 Fullscope) and soon after the other control manufacturers followed with similar products.

The PID controller has three tuning parameters and already in 1942, John Ziegler and Nathaniel Nichols (both from Taylor Instrument Co.) published their famous paper on "Optimum settings for automatic controllers" (Ziegler & Nichols, 1942). They write that *in spite of the multitude of air, liquid and electrically operated controllers on the marked, all are similar in that they incorporate one, two, or at most three simpler control efforts. These can be called* *"proportional", "automatic reset" and "pre-act".* The development of the tuning rules was based experiments combined with analog simulations. To speed up the process of analyzing the results from the analog simulations, Nichols rented the differential analyzer at MIT (Blickley, 1990). The paper had an enormous impact and despite being rather aggressive and having no adjustable tuning parameter, the Ziegler-Nichols-settings were for at least 50 years, up to about 1990, the by far most common rules used in academia and industry for systematic PID tuning.

The 1930s was a very active period for new ideas in automatic control, and during this period the following three elements became widely used in the process industry:[1]

- PID control, and in particular the use of integral action to reset the bias
- Cascade control
- Ratio control

In addition, to handle constraint changes, also selective (limit) control and split range control came into use. Ratio, cascade, selective and split range control are described in the book of Eckman (1945) on "Principles of industrial process control". He uses the term "metered control" to describe cascade control and "multiagent control" to describe the idea behind split range control. Later, additional features came into use, so that in the 1960s the following standard "advanced" control elements were used in the process industry (in addition to PID and on/off controllers);

E1*. Cascade control [2]

E2*. Ratio control

E3*. Selective (limit, override) control (to control many CVs with one MV)

[1] In the opinion of the author, these are the three main inventions of process control.
[2] The control elements with a * are discussed in more detail in this paper.

E4*. Input (valve) position control (VPC) to improve the dynamic response (using extra MVs).

E5*. Split range control (to control one CV with many MVs, but use only one MV at a time)

E6*. Separate controllers (with different setpoints) as an alternative to split range control

E7*. VPC as an alternative to split range control

All the above seven elements have feedback control as a main feature and are usually based on PID controllers. Ratio control seems to be an exception, but the desired ratio setpoint is usually set by an outer feedback controller. There are also several features that may be added to the standard PID controller, including

E8*. Anti-windup scheme for the integral mode

E9*. Two-degrees of freedom features (e.g., no derivative action on setpoint, different proportional gain for setpoint, setpoint filter)

E10. Gain scheduling (Controller tunings change as a given function of the scheduling variable, e.g., a disturbance, process input, process output, setpoint or control error)

In addition, the following more general model-based elements are in common use:

E11*. Feedforward control

E12*. Decoupling elements (usually designed using feedforward thinking)

E13. Linearization elements

E14*. Computation blocks (including nonlinear feedforward and decoupling)

E15. Simple static estimators (also known as inferential elements or soft sensors)

Finally, there are a number of simpler standard elements that may be used independently or as part of other elements, such as

E16. Simple nonlinear static elements (like multiplication, division, square root, dead zone, dead band, limiter (saturation element), on/off)

E17. Simple linear dynamic elements (like lead-lag filter, time delay, etc.)

E18. Standard logic elements

If we look more closely at these standard control elements, then we see that most elements have one input and one output variable, but there are a few elements that link a specific set of inputs (e.g., CVs) to a specific set of outputs (e.g., MVs). Thus, the control engineer needs to make structural pairing decisions to use the standard elements. This makes it difficult to handle very interactive processes where pairing is not obvious (and here model-based methods like MPC may be preferred), but on the other hand, the advantage with a fixed pairing is that the engineer can specify more directly how the system responds in a given situation.

This above list of control elements makes up what we can call the "industrial advanced process control world". It is sometimes called "classical advanced control" or "advanced PID control" and it is what we in this paper refer to as *advanced regulatory control*.

Almost in a different world, we have what may be called the "academic control world". These two worlds have been separated from the beginning. For example, in 1945, two control books were published. One was the industrial book by Donald P. Eckman on "Principles of Industrial Process control" (Eckman, 1945). The other was the academic book by Hendrik Wade Bode on "Network Analysis and Feedback Amplifier Design" (Bode, 1945). Although both books deal with mainly with feedback control, there are essentially no overlap between the two. Bode's book deals with analysis of linear control systems, including robustness and frequency analysis. Frequency analysis has had a large impact on understanding feedback systems and on the teaching of feedback control, but it is not used much for controller design in the process industry. Bode's book also includes Laplace transforms and transfer functions which under the name of "servo control", became very popular tools in the academic community around 1950. Transfer functions remain important for teaching and are still used in the industrial world, for example, in the design of lead-lag elements. However, since the 1970s, most academic researchers have switched from Laplace transforms to the time domain for research, both for numerical reasons and for generality, for example, to be handle nonlinear systems. This is closer to the approach in the industrial world, but otherwise the two worlds have remained largely separated.

The only academic control approach which has made its way into the industrial process control world is Model predictive control (MPC). The present state-space version of MPC is a result of a fusion between two heuristic (at least originally) industrial approaches from the 1970s, namely Dynamic Matrix Control (DMC) of Cutler & Ramaker (1980) and Model Predictive Heuristic Control of Richalet et al. (1978), and the academic optimal control theory (LQG control) of Kalman and others of the 1960s. MPC has been in industrial use since the 1970s and it came into common use in the petrochemical and refining industry at the end of the 1980s. However, in spite of a large academic focus on MPC as the preferred method for advanced control, both in terms of teaching and research, its adaptation into other process industries has been significantly slower than was anticipated in the 1990s.

In summary, the advanced regulatory control elements listed above, remain the main tool for advanced control in most process industries. Nevertheless, they have been largely ignored by the academic control world. Even the PID controller was for a long time considered obsolete by the academic community, and only after about 1980 did academic researchers (e.g. Morari, Astrom and their coworkers) develop improved methods to replace the Ziegler-Nichols tuning rules from 1942. What is the reason for this? Why has the academic control community, since it appeared as an academic discipline around 1950, largely neglected the control approaches being used in practice, in particular in the process industries? The main reason has probably been the belief that the control approaches used in industry were simplified and outdated and would soon be replaced by more modern and general approaches, for example, the optimal control and state space theory of the 1960s (LQG control), which is now implemented using MPC. The second reason is that the industrial control approaches seem ad-hoc because they are not presented within a systematic framework. Also, some of problems are challenging theoretically, such as the pairing problem and the stability of switched systems. The third reason, as pointed out by Foss in his famous paper from 1973, with the title "Critique of Chemical Process Control Theory" (Foss, 1973), is that the academic community has largely neglected the structural issues, that is, the decision on what to control (outputs, CVs) and how to decompose the system into decentralized controllers by pairing inputs (MVs) and outputs (CVs). Foss (1973) writes:

> *The central issue to be resolved by the new theories of chemical process control is the determination of control system structure. ... Which variables should be measured, which inputs should be manipulated, and what links should be made between these two sets? ... There is more than a suspicion that the work of genius is needed here, for without it the control configuration problem will likely remain in a primitive, hazily stated, and wholly unmanageable form.*

In some systems, for example for operation of multiple cars in traffic (vehicle formations), an important reason for decentralized control is that there is only limited information exchange between the subsystems (cars). However, in process control applications, the information about all process variables is usually centralized, so the main motivation for applying decomposition and decentralized control is mainly that it is simpler and that it usually is good enough. It allows for independent controller tuning without the need for a process model describing the detailed dynamics and interactions in the process. Multivariable controllers may always outperform decentralized controllers (at least in theory), but this performance gain must be traded off against the cost of obtaining and maintaining the process model needed for multivariable control.

Following the paper of Foss (1973), some research was initiated on control structure design and "chemical plant(wide) control", in particular, the three-part series Morari et al. (1980), Morari & Stephanopoulos (1980a) and Morari & Stephanopoulos (1980b)). They introduced the concept of feedback-optimizing control. The main idea is to always control (select as CVs) the active constraints, and for the remaining unconstrained degrees of freedom, to control what Skogestad (2000) later called "self-optimizing" variables. However, about at the same as time (in the 1980s), MPC became popular and many academic researchers expected that it would soon replace all the seemingly ad-hoc and complex industrial "advanced PID" structures. Therefore, with a few exceptions, the academic research efforts in this area died away around 1995. A review of some academic research in control structure design and advanced regulatory control is found in Chapter 10 of Skogestad & Postlethwaite (2005). Good overviews of the industrial status on advanced regulatory control are found in the books "Basic and advanced regulatory control" by Wade (2004) and "Advanced process control - beyond single-loop control" by Smith (2010). A good source of practical process control case studies are the many papers and books by Bill Luyben, e.g., Luyben et al. (1998).

Do we really need a theory for advanced regulatory control when it seems to be working well already? Yes, we do. First, the fact that it is working, does not mean that it could work much better. Second, without theory, it is difficult to improve the methods and suggest alternatives. Third, without some theory, teaching becomes difficult.

Forsman (2016) from the Perstorp chemical company writes that "traditional expositions of classical control structures often lack a systematic and holistic perspective. The step from control specifications to choice of control structure is seldom obvious, and it is often unclear if the problem at hand could be solved by other structures than the one presented. As a consequence it is not easy for an inexperienced user to design a new control structure that solves a given problem, or to combine several structures. In comparison, MPC design is definitely more systematic."

Hägglund & Guzman (2018) conclude that the regulatory control layer is an almost neglected area when it comes to research and development, with the exception of PID controller tuning. They say that "very little work has been presented related to the basic control structures that connect the PID controllers" and that "the impact of advances in this field has a great potential, since these structures appear in so many places in so many process industries".

Notation

The most important notation is summarized in Figures 1 and 2. The feedback controller has as inputs the controlled variable (CV) and its setpoint (CV_s), and as output the manipulated variable (MV). The process input is denoted u, and the process outputs are denoted y (with reference value or setpoint y_s) and w (extra measurements).

In some cases, the process input u may not be equal to the actual (physical) input ($\tilde{u}$) which is applied to the process, for example, because of saturation in the final control element (which is usually a valve or sometimes a variable speed pump in process control), or because of the use of selectors.

Figure 1: Block diagram of general feedback controller (usually dynamic and possibly nonlinear). In the multivariable case, the feedback controller may consist of several simpler control elements.
- CV = controlled variable (with setpoint CV_s) = controller input.
- MV = manipulated variable = controller output.

Figure 3 shows a simple feedback control system which acts on the control error e and where we have MV=u and CV=y_m (measured process output). This is called "one degree-of-freedom" control because the controller acts on only one variable, namely the setpoint error $e = y_s - y_m$.

The more general two degrees-of-freedom controller in Figure 1, makes independent use of CV= y_m and $CV_s = y_s$. A two degrees-of-freedom control system can be realized in many ways. One common implementation with a

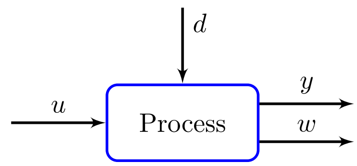


Figure 2: Block diagram of general process (usually dynamic and nonlinear).
- u = process input (often valve position z)
- d = disturbance
- y = process output (with setpoint y_s) (usually measured)
- w = extra measured process variable

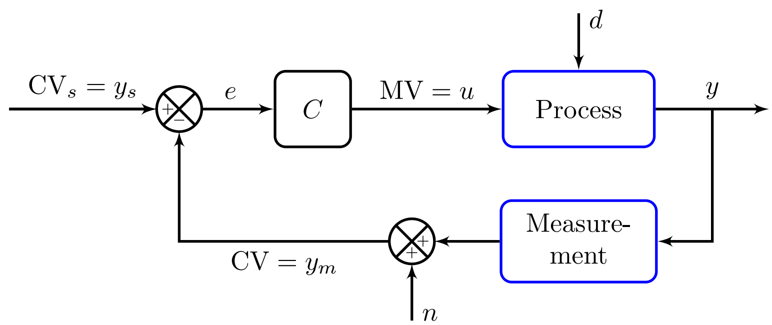


Figure 3: Block diagram of common "one degree-of-freedom" negative feedback control system. All blocks are possibly nonlinear. The objective of the control system is to keep the process output y close its setpoint y_s in spite of disturbances d.
- y_m = measured value of y
- n = measurement noise/error/bias
- C = feedback controller with input $e = y_s - y_m$.

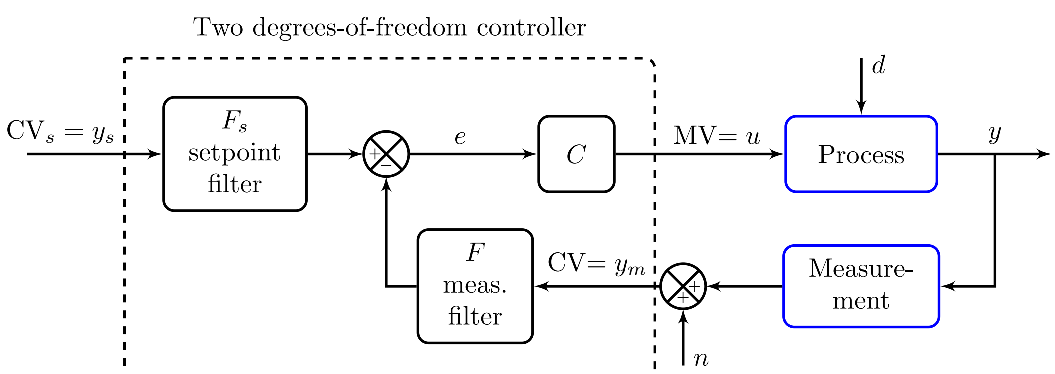


Figure 4: Two degrees-of-freedom control system with setpoint filter F_s and measurement filter F. All blocks are possibly nonlinear.

setpoint prefilter F_s for y_s is shown in Figure 4. Here, we have also added a measurement filter F for y_m. Another approach is to add, in parallel to C, a feedforward element C_{Fy} from the setpoint y_s to MV=u (Figure 5).

In process control, the measurement block is often represented by a time delay or a first-order process. However, in this paper, we will usually not include the measurement block or the measurement noise/error (n), that is, we assume perfect measurement with $y_m = y$. Of course, this is not correct but it simplifies the block diagrams. In addition, we will usually not include the measurement filter F in the block diagrams, although it is an important design parameter in many cases. With these simplifications, we have CV=y.

In summary, we will often write y and u for the controller input and output signals, although it strictly speaking (with reference to Figure 1) would be more correct to write CV and MV.

In the linear case, the one degree-of-freedom feedback controller in Figure 3 then becomes (with Laplace transforms and deviation variables)

$$u = C(s)(y_s - y) \tag{1}$$

and the two degrees-of-freedom feedback controller in Figure 4 becomes

$$u = C(s)\left(F_s(s)y_s - F(s)y\right) \tag{2}$$

Here C is the feedback controller (e.g., PID), whereas F_s and F typically are lead-lag transfer functions, with a steady-state gain of 1. In process control, we often use $F = 1$ (no filter) or a first-order measurement filter

$$F(s) = \frac{1}{\tau_F s + 1} \tag{3}$$

As mentioned, an alternative two degrees-of-freedom realization is to complement the one degree-of freedom controller C in (1) with a “feedforward” element

C_{Fy}. For the linear case, we then have with feedforward from the setpoint:

$$u = C(s)(y_s - F(s)y) + C_{Fy}(s)y_s \tag{4}$$

Feedforward control is more commonly used for measured disturbances, and a linear feedforward control system (with no feedback) is shown in Figure 5. Here, we have

$$u = C_{Fd}(s)d + C_{Fy}(s)y_s$$

where d is a measured disturbance. The system is linear because the independent contributions from d and y_s are added together.

A more general (possibly nonlinear) control system with combined feedforward and feedback control is shown in Figure 6. Here, disturbance d_1 is measured and d_2 is unmeasured. The feedback controller C should have integral action to give zero steady-state offset for unmeasured disturbances d_2, whereas the feedforward element for d_1 is based on a inverting the process model. Many other control schemes can be rewritten in this form, for example Internal Model Control (IMC), MPC (where the block "feedback controller" is actually the estimator) and the use of transformed inputs v Skogestad et al. (2023). In the latter case, the "feedforward element" block is static and may also include decoupling and linearization.

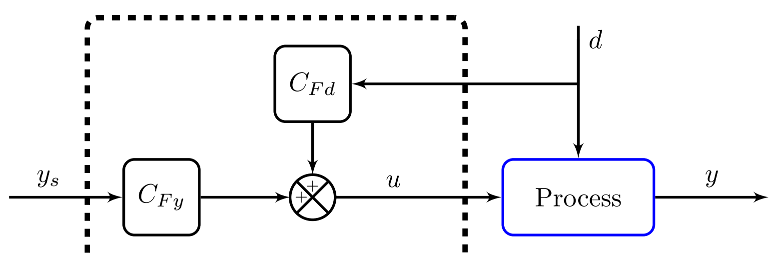


Figure 5: Block diagram of feedforward control system with linear combination of feedforward from measured disturbance (d) and setpoint (y_s).

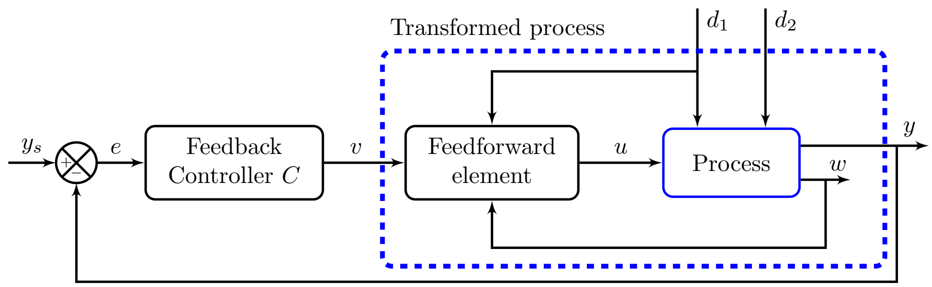


Figure 6: Block diagram of control system with combined feedforward element (often nonlinear) and feedback controller (often linear). The "transformed process" is used for designing the feedback controller C which uses v=MV to provide a feedback correction to the feedforward part.

2. Feedback versus feedforward control

Control makes use of two main principles, namely feedforward and feedback. Feedforward is model-based and is easier to understand intuitively for humans because it builds directly on our view of a world with causes (u) and effects (y). With feedback control, this relationship is reversed, which makes it more difficult to understand. Interestingly, feedforward and feedback solutions may in some cases yield identical nominal performance. However, if given a choice, feedback solutions are usually preferred because they are much less sensitive to model errors. This is illustrated in the following example.

Example. Feedforward versus feedback control for setpoint tracking

This simple example shows that identical nominal setpoint responses can be obtained with either feedforward or feedback control. The main purpose of the example is to demonstrate the advantage of feedback control, and in particular of integral action, in dealing with model uncertainty. A more general treatment is found in Skogestad & Postlethwaite (2005) (pages 203-205). We consider a linear first-order process with a time constant $\tau = 6$ and steady state gain $k = 3$. The following linear model describes the dynamics:

$$\tau \frac{dy(t)}{dt} = -y(t) + ku(t)$$

However, for our purposes the Laplace domain is more convenient. Using deviation variables, we may write $y(s) = G(s)u(s)$, where the process transfer function is

$$G(s) = \frac{k}{\tau s + 1}, \quad k = 3, \ \tau = 6 \tag{5}$$

We want to design a control system such that the output response $y(t)$ to a step change in the setpoint y_s is first-order with a desired time constant $\tau_c = 4$.

$$\text{Desired response :} \quad y = \frac{1}{\tau_c s + 1} y_s = \frac{1}{4s + 1} y_s$$

Note that we want $\tau_c = 4$, so we want a "speedup" compared to the original dynamics by a factor $\tau_c/\tau = 6/4 = 1.5$.

Feedforward solution. We use feedforward from the setpoint (Figure 5):

$$u = C_{Fy}(s) y_s$$

where we choose

$$C_{Fy}(s) = \frac{1}{\tau_c s + 1} G(s)^{-1} = \frac{1}{k} \frac{\tau s + 1}{\tau_c s + 1} = \frac{1}{3} \frac{6s + 1}{4s + 1} \tag{6}$$

The output response becomes as desired,

$$y = \frac{1}{4s + 1} y_s \tag{7}$$

Feedback solution. We use a one degree-of-freedom feedback controller (Figure 3) acting on the error signal $e = y_s - y$:

$$u = C(s)(y_s - y)$$

We choose a PI-controller with $K_c = 0.5$ and $\tau_I = \tau = 6$ (using the SIMC

PI-rule with $\tau_c = 4$, see Section 4.2):

$$C(s) = K_c(1 + \frac{1}{\tau_I s}) = 0.5\frac{6s+1}{6s} \tag{8}$$

Note that we have selected $\tau_I = \tau = 6$, which implies that the zero dynamics in the PI-controller C, cancel the pole dynamics of the process G. The closed-loop response becomes as desired:

$$y = \frac{1}{\tau_c s + 1} y_s = \frac{1}{4s+1} y_s \tag{9}$$

Proof: $y = T(s)y_s$ *where* $T = L/(1+L)$ *and* $L = GC = kK_c/(\tau_I s) = 0.25/s$. *So* $T = \frac{0.25/s}{1+0.25/s} = \frac{1}{4s+1}$.

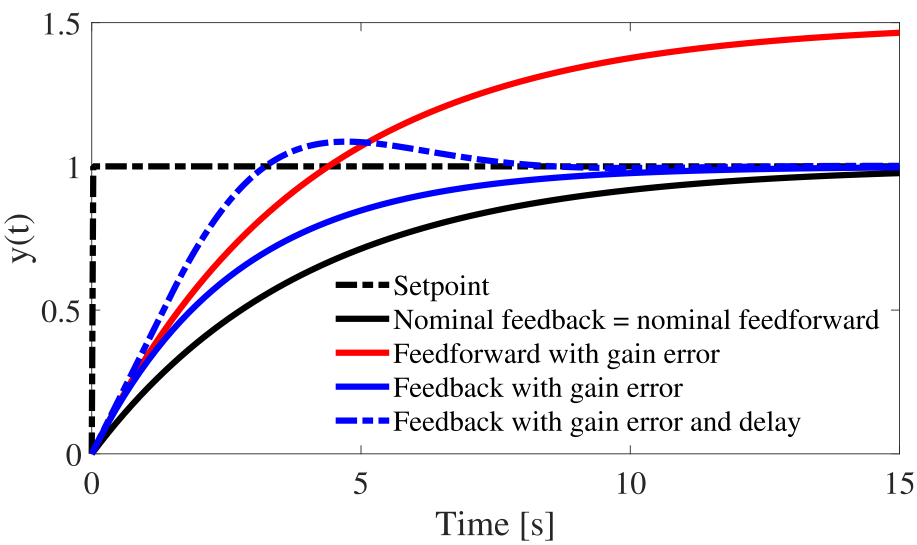


Figure 7: Setpoint response for process (5) demonstrating the advantage of feedback control for handling model error.

Thus, we have two fundamentally different solutions that give the same nominal response, both in terms of the process input $u(t)$ (not shown) and the process output $y(t)$ (black solid curve in Figure 7). However, as illustrated by the simulations in Figure 7, the feedback PI-control solution is a lot more robust towards model error. Consider an increase in the process gain by 50% (from

$k = 3$ to $k' = 4.5$). With the feedforward controller (6), we get the setpoint response $y = \frac{1.5}{4s+1} y_s$ (red curve). Note that the steady-state gain from y_s to y has changed from 1 in (7) to $k'/k = 1.5$. That is, the process gain increase of 50% translates directly into a 50% steady-state control error. On the other hand, with the PI-controller (8), we get the setpoint response $y = \frac{1}{2.67s+1} y_s$ (blue solid curve), so the steady-state gain is unchanged at 1. That is, with PI-control a process gain increase of 50% translates into 0% steady-state control error. The reason for this is the integral action in the controller. However, the process gain increase of 50% does translate into a corresponding reduction in the closed-loop time constant; from 4 to 4/1.5=2.67. Potentially more seriously, the increased gain in the loop may result in instability, for example, if the process or the measurement of y has a time delay. Fortunately, the feedback solution is also fairly robust with respect to time delay changes. This is shown by the blue dashed curve in Figure 7, which shows that even by adding a measurement delay $\theta = 1.5$, the response with PI-control is acceptable. We see that there are some oscillations appearing, but the closed-loop system is stable. Note that instability cannot appear with feedforward control, at least not in the linear case, so this is an advantage of feedforward control.

Thus, the preferred solution is clearly PI feedback control. What about feedforward control? It may be combined with feedback control and it will improve the response for y, if the measurement delay θ for y is larger than about $\tau_c/4$. However, feedback should always be included to achieve zero steady-state offset due to model error and unknown disturbance.

In summary, there are two things to be learned from this example. The first is the power of feedback control in dealing with model uncertainty. The second is that one must be careful not to end up with using feedforward control for cases where feedback control is a much better solution. The latter is a relevant for some controller design methods, for example, model predictive control (MPC).

3. Decomposition of the control system

3.1. Introduction

It is often difficult to explain to someone outside the control community what we mean by "control", because this word has different meanings for different people. Here is a simple definition that I use for my students:

> *"Control" is to make active use of the inputs u to counteract disturbances d such that the outputs y stay close to their desired setpoints y_s.*

The word "active" is to emphasize that this is a dynamic system. Also note here that the word "setpoint" (= command) is included in this definition. However, many control engineers, especially in academia, want to expand the scope of control to also include generating the setpoints, which usually involves economic optimization. This leads to the following definition of the "overall control system":

> *The "overall control system" continuously adjusts the process inputs $u(t)$ so that the controlled system remains stable and close to economically optimal for varying disturbances d.*

For designing and implementing the "overall control system" there are two main approaches:

1. One "big" optimizing controller (one layer). This is centralized optimizing control where the tasks of optimization and control are combined into one a single cost function J. There are no setpoints. In some sense this is the obvious approach, and it has recently become popular in academia as Economic Model Predictive Control (EMPC). One immediate problem is that it may be difficult to put a monetary value on robustness (stability margins). Furthermore, unless the time scales are overlapping, there is little economic benefit of separating the optimization and control tasks.

2. Decomposition into smaller blocks, for example, as illustrated for process control applications in Figure 8. This is the approach used in practice in the process industry, and more generally for essentially all large-scale systems. There are two fundamental ways of decomposing the control system:

   I Vertical (hierarchical; cascade) decomposition

   II Horizontal (decentralized) decomposition

   The vertical decomposition, for example, into separate optimization and control layers, is based on time scale separation. The motivation is that the two tasks of optimization and control are usually at different time scales, which in most cases makes it possible to separate their solutions with only a small loss in performance. Both the optimization and control layers may be further divided into additional layers as shown in Figure 8.

   The horizontal decomposition makes use of simple elements/blocks (Figure 8) with a preference on using single-input single-output feedback controllers (usually PID) whenever possible.

The above definition of "control" applies to the two control layers in Figure 8 (regulatory and supervisory control), whereas the definition of "overall control system" includes also the (local) optimization layer, and in some cases higher layers, including the scheduling layer.

3.2. Structural decisions and decomposition

To be able to decompose the control system into smaller tasks (Figure 8), the engineer needs to make many *structural decisions* which have a large effect on the subsequent controller design. As mentioned in the introduction, this was pointed out clearly by Foss (1973) in his critique article. Morari et al. (1980) followed up this work and write that "a central point often is the unavailability of a method for synthesizing control structures for a complete (chemical) plant. Considering how many papers have been written on control of a single unit

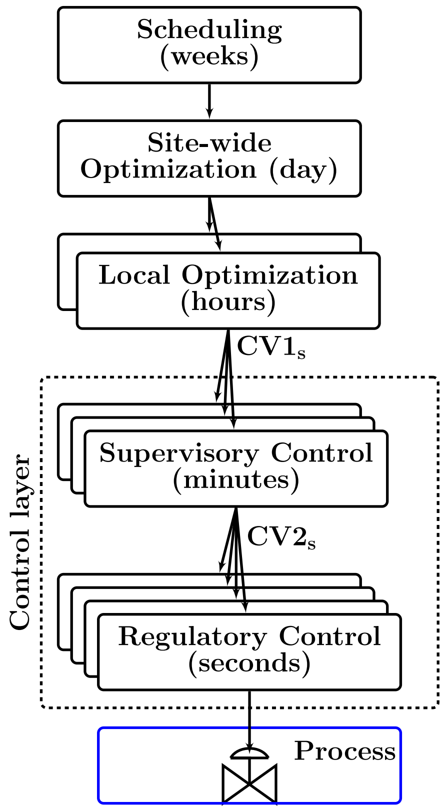


Figure 8: Decomposition of “overall control system” for optimal operation in typical process plant. There is a vertical decomposition into decision layers based on time scale separation, and a horizontal decomposition into decentralized controllers. There is also feedback from the Process to the various layers (not shown in the figure). The decisions in the blocks and layers above the regulatory layer are often manual. This paper considers use of automatic control in the three lowest layers, with focus on the supervisory control layer.
CV1 = Economic controlled variables
CV2 = Stabilizing controlled variables
RTO = Real-time optimization
MPC = Model predictive control
ARC = Advanced regulatory control

operation like distillation, (chemical) plant control has been discussed only a few times because of its inherent complexity". Morari et al. (1980) write that a control structure is composed of the following items:

1. "A set of variables which are to be controlled to achieve a set of specified objectives
2. A set of variables which can be measured for control purposes
3. A set of manipulated variables
4. A structure interconnecting measured and manipulated variables"

These items (structural decisions) are in the process industry referred to as "plant(wide) control" but a more general term is *control structure design.* The first item of controlled variable (CV) (output) selection is discussed in more detail below. The second and third items are often referred to as *input-output selection.* The fourth item is known as *input/output-pairing* or more generally as *control configuration selection* (Skogestad & Postlethwaite, 1996) (Skogestad & Postlethwaite, 2005) or *decomposition of the control system.*

There is a lot of flexibility in these decisions. For example, Shinskey (1981) (page 119) writes in relation to selecting input and output variables for the controller:

> "There is no need to be limited to single measurable or manipulable variables. If a more meaningful variable happens to be a mathematical combination of two or more measurable or manipulable variables, there is no reason why it cannot be used."

3.3. I. Vertical (hierarchical) decomposition.

For process control applications, there are three main layers (Figure 8):

1. *Optimization layer* (real-time optimization, RTO): Determine optimal setpoints for the economic controlled variables (CV1) such the economic cost $J_\$$ is minimized.

In process control, this layer (if present) is usually based on a detailed nonlinear steady-state model where the objective is to minimize and economic cost of the form $J_\$ = p_F F - p_P P + p_Q Q$ [$/s]. Here F denotes feed streams (raw material) [kg/s], P denotes product streams [kg/s], Q utility (energy) usage [W], and p denotes the corresponding prices (e.g., in [$/kg]).

2. *Supervisory ("advanced") control layer.* This layer is the main focus of this paper and it has three main objectives:

    - Follow the setpoints ($\mathrm{CV1}_s$) coming from economic optimization layer
    - Switch between active constraints (change CV1-variables)
    - Look after the regulatory layer (avoid that the physical inputs u saturate, etc.)

    The degrees of freedom (MVs) for the supervisory control layer include the setpoints ($\mathrm{CV2}_s$) to the basic control layer as well as some of the physical inputs (u).

3. *"Basic" regulatory control layer* (PID layer): This is the basic stabilizing control layer, where the main objective is to avoid that the process drifts away from its desired steady state on a fast time scale. This is done by keeping selected controlled variables (CV2) at desired setpoints. These setpoints are either constant or come from the layers above.

In practice, the distinction between the various layers may not be so clear. In some cases, the two control layers are combined. Usually in industry, there is no optimization layer, which means that the economic optimization (if any) must either be performed manually or be moved into the control layer, for example, using selective control or split range control (see below). There may also be further vertical decomposition within each layer using cascade control.

In Figure 8, the setpoints $\mathrm{CV1}_s$ and $\mathrm{CV2}_s$ are the inputs and outputs of the supervisory layer and a key decision is what these variables should be. It is often not obvious. There is also feedback of measurements (y, w) (or possibly estimates) from the process to the various layers and blocks but this is not shown in Figure 8.

In the process industry, the supervisory control tasks are often solved manually. For *automatic* supervisory control, which is the focus of this paper, the process industry uses either advanced regulatory control (ARC) or model predictive control (MPC) or a combination where MPC is a block. This is usually a setpoint-based MPC which sits on top of a basic PID-layer.

Note that there in many implementations is no formal separation between the regulatory and supervisory control layers, and in industry these are often implemented in the same distributed control system (DCS). However, the common use of cascade control within the DCS layer means that there in reality is a decomposition based on time scale separation within the control layer. In this paper, the two control layers are treated separately, because of the fundamental difference between stabilizing (regulatory) control tasks and economic (supervisory, servo, advanced regulatory) control tasks.

It is sometimes claimed that the vertical decomposition in Figure 8 has a potential problem with inconsistency between the models used in the various layers, but this is a misunderstanding. The lower layers follows the commands (setpoints) from the layers above, so except for a dynamic (transient) deviation, there will be no inconsistency, at least not at steady state with integral action in the controllers. Actually, one of the main reasons for using the decomposition in Figure 8 is to make it possible to use different models and different objectives in each layer. Typically, the optimization layer (RTO) uses a physical nonlinear model (usually static), the supervisory layer (with MPC) uses an experimental dynamic linear model, whereas the regulatory PID-controllers are tuned online or based on a simple first-order plus delay model.

The main disadvantage with the decomposition in Figure 8 appears if the assumption of time scale separation does not hold. For example, a batch process is never at steady state, so it may be necessary to include dynamics in the RTO layer. For some simple processes, it may be good to combine the MPC and PID layers. In more rare cases, economic model predictive control (EMPC) may be an attractive solutions, as it may combine all three layers (RTO, MPC and PID).

3.4. Time scale separation

A vertical decomposition into layers, including the use of cascade control, depends on a sufficient time scale separation between neighboring layers. Let

$$\tau_{c1} \text{ (large) } = \text{time constant of upper layer (outer loop)}$$

$$\tau_{c2} \text{ (small) } = \text{time constant of lower layer (inner loop)}$$

The time scale separation is then defined as the ratio τ_{c1}/τ_{c2}. To avoid potential undesired interactions (”fighting”) between the two layers (loops), Shinskey (1981) (page 12) recommends a time scale separation of at least 4, whereas Skogestad & Postlethwaite (2005) (page 425) and Smith (2010) (page 69) recommend at least 5. If the time scale separation gets too small, typically 3 or less, the layers (loops) start interacting and resonance occurs (Young, 1955) (p. 310), such that performance degrades even nominally.

A larger time scale separation gives robustness against process gain variations in both layers (loops). Note in this respect that a process gain *decrease* in the lower layer (inner loop) is “bad” as it translates into a larger (“slower”) value of the actual τ_{c2}. This reduces the time scale separation τ_{c1}/τ_{c2} and in addition τ_{c2} appears as an effective delay as seen from the upper layer (outer loop). On the other hand, for the upper layer (outer loop), a process gain *increase* is “bad” as it translates into a smaller (”faster”) value of τ_{c1} which reduces the time scale separation.

To achieve robustness to both these potential gain variations, it is often recommended to have a time scale separation of 10 (or larger). However, the disadvantage with a too large time scale separation is that it “eats up” more of the available time window, which may be a problem with many layers of cascade control.

In summary, a rule of thumb is to have a time scale separation between layers (cascade loops) in the range 4 to 10.

The understand the basis for the lower value of 4, assume that the closed-loop response of the lower layer (inner loop) is approximated as a first-order system. When the upper layer (outer loop) makes a step change in its MV (which is the setpoint y_{2s} to the lower layer), then it is desirable that the actual value (y_2) immediately goes to y_{2s}. However, the actual time response for a first-order system is

$$y_2(t) = (1 - e^{-t/\tau_{c2}})\ y_{2s}$$

where t is time and τ_{c2} is the closed-loop time constant of the lower layer. Note that $1-e^{-1} = 0.632$, $1-e^{-2} = 0.865$, etc. Thus, as t/τ_{c2} increases from 1 to 2 to 3 to 4, and to 5, the approach to steady state improves from 63.2% to 86.5% to 95% to 98.2 %, and to 99.3%. Thus, at 4 time constants the approach is 98.2%, and convergence (or steady state) has for practical purposes been reached.

Another justification for the lower value of 4, which is especially relevant for cascade control, follows by requiring that the interactions between the loops should not result in oscillations. Consider the series cascade control system in Figure 14. For the linear case, all closed-loop transfer functions contain the term $S = (1+L)^{-1}$ (sensitivity) where $L = G_2C_2 + G_1G_2C_2C_1$. Assuming that both loops (layers) are approximated as first-order systems, we have approximately $G_1C_1 = \frac{1}{\tau_{c1}s}$ and $G_2C_2 = \frac{1}{\tau_{c2}s}$. Setting $1 + L(s) = 0$, we then find that the closed-loop poles are the solutions to $\tau_{c1}\tau_{c2}s^2+\tau_{c1}s+1 = 0$. To avoid oscillations, the poles must not be complex, which requires $\tau_{c1}/\tau_{c2} \geq 4$.

The limiting case of infinite time scale separation corresponds to $\epsilon = (\tau_{c1}/\tau_{c2})^{-1} \rightarrow 0$, which is the singular perturbation condition in the mathematical literature. Note that a time scale separation between 4 and 10, corresponds to ϵ between 0.25 and 0.1.

3.5. II. Horizontal decomposition (decentralized).

The second way of decomposing the control problem, is to divide each layer into separate blocks (see Figure 8), often based on physical separation (distance). The objective of the decomposition is usually to make it possible to use decentralized control with single-loop PID controllers. The most important decision for decentralized control is the input (MV) - output (CV) pairing. For this, the two most important *pairing rules* are:

- **"Pair close" pairing rule**: The MV should have a large, fast, and direct effect on the CV. In particular, we want a small effective delay (small θ), and also a large steady-state gain (large k) and a fast dynamic response (small τ).
- **"Input saturation" pairing rule**: A MV that may saturate should only be paired with a CV that can be given up (when the MV saturates)

If we do not follow the input saturation rule, then we need to switch to using an alternative MV when the primary MV saturates. This adds complexity as we need to add a MV-MV switching logic, for example, split range control.

For some interactive processes, the use of single-loop PID controllers may give poor performance, and multivariable control (e.g., MPC) or the use of decoupling should be considered. The Relative Gain Array (RGA) (Bristol, 1966) may be a useful tool for analyzing interactive systems. In particular, pairing on negative steady-state RGA-elements should be avoided, as it may result in instability if an input (MV) saturates (Grosdidier et al., 1985).

In addition to single-loop PID feedback controllers, further horizontal decomposition (operating at the same time scale) may involve selectors, split range elements, valve position control, ratio and feedforward elements, decouplers, nonlinear elements and estimators (soft sensors).

3.6. What to control (CV1 and CV2)?

As seen from Figure 8, the variables CV1 and CV2 (or rather their setpoints) interconnect the layers, and a key decision is what these variables should be.

However, the choice of these variables is frequently not obvious.

3.6.1. Choice of (stabilizing) controlled variables CV2 in regulatory control layer

The objective of the regulatory layer is to avoid that the system drifts away from its desired steady state on a short time scale. Therefore, we should select controlled variables (CV2) which are sensitive (with a large gain) to inputs (u) and disturbances (d). The sensitivity to the inputs is the most important. Typical choices for the controlled variables (CV2) are levels, selected pressures and selected temperatures.

3.6.2. Choice of (economic) controlled variables CV1 in supervisory control layer

From an economic point of view, the following variables should be controlled (Skogestad, 2003);

- **CV1=Active constraints** (where "active" means that it is (economically) optimal to operate at this constraint). The setpoint is the constraint value.
- **CV1="Self-optimizing" variables** for the remaining unconstrained degrees of freedom. The setpoint needs to be determined by optimization, either using a model (offline or online (e.g., RTO)) or experimentally (e.g., using extremum seeking control).

The ideal self-optimization variable is the gradient $J_u = dJ/du$ (the derivative of the cost J with respect to the unconstrained degrees of freedom u) which has an optimal setpoint of 0. However, the gradient J_u is rarely available as a measurement and its estimation may be difficult, so in practice we would like to use a single measurement (CV1=w) or a measurement combination (usually a linear combination, CV1 $= Hw$). The idea is that the optimal setpoint (CV1$_s$) should be almost constant, that is, depend only weakly on disturbances. In addition, the gain from the MV to the selected CV1 should be large (Skogestad, 2000). The simplest method for selecting optimal measurement combinations as self-optimizing variables (selecting H) is the "nullspace method", but this only takes into account that the setpoint should be independent of disturbances. To take into account measurement error/noise (which effect is reduced if the gain is large) one should use the more general "exact local method". For more details, the reader is referred to Alstad et al. (2009) and Jäschke et al. (2017).

3.7. Active constraint switching

From an economic point of view, the control of the active constraints is usually the most important. The reason is that there may be a large economic penalty imposed by having a "back-off" from the optimal constraint value. The identification and switching between active constraints is usually handled by the supervisory layer. This may seem surprising, because one may imagine that identifying active constraints requires optimization. However, it turns out that in most cases this is not necessary, because the reaching of a constraint can be identified (measured) online, so it is actually only a switching policy that needs to be determined and designed.

Assume we are operating a control system using single-loop controllers (each controller has at any given time one MV and one CV). When a new constraint is reached, then some change usually needs to be made to the control system. In the simplest case, with a short-term saturation on the MV, one may not need to do anything, except for activating anti-windup for the integral action. However, if there is a long-term (steady-state) change in the active constraint set, then one usually needs to change the control structure, that is, one needs to change the pairing of MVs and CVs. There is a fundamental difference between MV and CV constraints because we need an MV to control a CV, whereas an MV can simply be set at its optimal constraint value. Thus, we have three different constraint switching cases: MV-MV, CV-CV and MV-CV switching (Reyes-Lúa & Skogestad, 2020b). Block diagrams for the two first cases are shown in Figures 9 and 10, respectively. Note here that, the "Feedback controller" block may be a combination of simpler control elements and also note that setpoints (CV_s) have been omitted for simplicity.

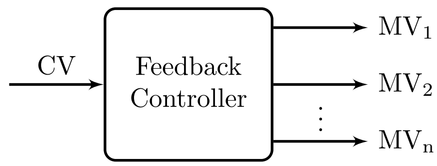


Figure 9: MV-MV switching is used when we have multiple MVs to control one CV, but only one MV should be used at a time. The block “feedback controller” usually consists of several elements, for example, a controller and a split range block.

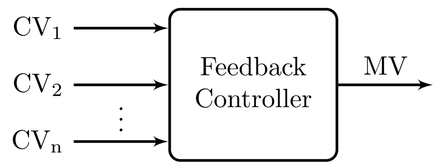


Figure 10: CV-CV switching is used when we have one MV to control multiple CVs, but the MV should control only one CV at a time. The block “feedback controller” usually consists of several elements, typically several PID-controllers and a selector

3.7.1. MV-MV switching.

MV-MV switching is used for cases where multiple MVs (process inputs, degrees of freedom) are used to control one CV (process output), but only one MV should be used at a time. It is also known as input sequencing or multiagent control. When a constraint on the present MV is encountered, one switches to using another MV. For MV-MV switching, we will later consider three alternative approaches (control elements):

1. Split range control
2. One controller for each MV, but each with a different CV-setpoint
3. Valve position control

3.7.2. CV-CV switching.

CV-CV switching is used for cases where one MV (process input) is used to control multiple CVs (process outputs), but only one CV should be controlled at a time. CV-CV switching is frequently used for satisfying inequality constraints. When a CV constraint is encountered, one “gives up” controlling the present CV. CV-CV switching is implemented using selectors as discussed in more detail later.

3.7.3. MV-CV switching

MV-CV switching is used for cases where it is optimal to "give up" (stop controlling) a CV when a constraint on the MV is encountered. We can distinguish between two different cases.

3.7.4. Simple MV-CV constraint switching

If the CV that can be given up is controlled with the MV that saturates, that is, if we followed the "input saturation rule", then it is not necessary to do anything (except for anti-windup).

3.7.5. Complex MV-CV constraint switching (repairing of loops).

Consider next the case where the CV that can be given up is controlled with another MV. That is, we have paired an MV which may saturate with a CV which cannot be given up. This means that the "input saturation pairing rule" was *not* followed, for example, because it did not agree with the "pair-close" rule. This is a more complex case, where one needs to do an input-output "repairing", which may be realized using a series combination of MV-MV and CV-CV switching. First, we use MV-MV switching to keep controlling the CV which cannot be given up, and then we use CV-CV switching (a selector) to give up the other CV.

We discuss later these switches in more detail and how they can be realized using simple control elements.

4. Basic control loops (PID control)

4.1. The PID controller

The most important of the standard control elements is the feedback PID controller and the most important for a PID-controller to work well is to have a good "pairing" between the MV (u) and the CV (y).

Having decided on the pairing, the PID-controller needs to be tuned. There exists many variants and reparameterizations of the PID controller. The most common "ideal-form" PID controller is given by

$$u(t) = K_c e(t) + K_c \tau_D \frac{de(t)}{dt} + \underbrace{\overbrace{\frac{K_c}{\tau_I}\int_{t_0}^{t} e(t')dt'}^{u_I} + u_0}_{\text{bias}=b} \tag{10}$$

where y is the measured CV-value, u is the MV and $e = y_s - y$ is the setpoint deviation (control error). This a one degree-of-freedom controller, since the controller only acts on the error e, see Figure 3.

The "bias" b is defined as the sum of the constant u_0 and the "output" u_I from the integrator,

$$b = u_0 + u_I \tag{11}$$

With integral action, the value of u_0 only matters initially, when the controller is activated (turned on or reactivated), because over time the contribution u_I will "reset" the bias to drive the system to its desired steady state. Without integral action (P- or PD-controller), the value of u_0 is important.

The PID controller has three tuning parameters

$$K_c = \text{controller gain}$$

$$\tau_I = \text{integral time}$$

$$\tau_D = \text{derivative time}$$

In addition, there is often a filter F on the measurement of y (Figure 4), for example, a first-order filter (3) with time constant τ_F .

To avoid the derivative "kick" for setpoint changes, it is common to *not* use derivative action on the setpoint (Figure 11). This then becomes a special case of a two degrees-of-freedom controller, because the setpoint y_s and the measurement y are treated differently.

In most cases, D-action is not used and there are then only two tuning parameters. With only two parameters, it may be tempting to use trial-and-error online tuning, but unless one happens to be lucky, this is time consuming and not recommended. Instead, it is recommended that the tuning is based on a first-order with delay model, obtained with an experiment that excites the process, for example, a step response; see next.

4.2. PID tuning

Design rules for the PID controller were developed by Ziegler & Nichols (1942), and these remained the main tuning rules for at least 50 years. This is surprising, considering that the Ziegler-Nichols rules are aggressive (aiming for a one-quarter decay ratio, whereas one rather should avoid oscillations nominally), have no tuning parameter, and work poorly for "fast" processes (where a small integral time is optimal). In particular, the Ziegler-Nichols-rules work poorly for a pure time delay process, and this is probably reason for the (unjustified) popularity of the Smith Predictor. The only other set of PID tuning rules that were available until about 1985, were the Cohen & Coon (1953) rules, which are also aggressive (aiming at a one-quarter decay ratio) and with no tuning parameter, and in most cases give similar PID-tunings as Ziegler-Nichols.

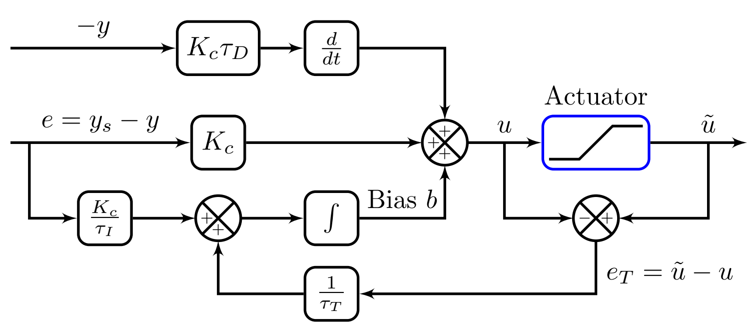


Figure 11: PID-controller with anti-windup using back calculation and without D-action on the setpoint. (Åström & Hägglund, 1988).
u = value computed by the controller.
$\tilde{u}$ = actual value applied to the process.
τ_T = tracking time constant for anti-windup
More generally, the block "Actuator" does not need to be a saturation element, it could represent any element that breaks the link between u and the actual input $\tilde{u}$, for example, a selector.

Finally, in the 1980s academic researchers started showing some interest in PID control. Åström & Hägglund (1988) considered the implementation of PID controllers and recommended the anti-windup scheme shown in Figure 11.

Rivera et al. (1986) proposed the Internal Model Control (IMC) PID-tuning rules and Smith & Corripio (1985) proposed their similar "direct synthesis" rules.

The IMC and "direct synthesis" rules are based on specifying the desired closed-loop response. It is not possible to eliminate a time delay θ in the process G, so a typical specification is a first-order plus delay response, which in the Laplace domain may be written as

$$y(s) = \frac{e^{-\theta s}}{\tau_c s + 1} y_s(s) \tag{12}$$

In the time domain, this corresponds to

$$y(t - \theta) = (1 - e^{-t/\tau_c})\ y_s \tag{13}$$

for a step setpoint change y_s occurring at $t = 0$. Using the Laplace domain, we have $y(s) = T(s)y_s(s)$ where $T = \frac{GC}{1+GC}$. From this one can find algebraically the corresponding controller C (which turns out to be a Smith Predictor controller). Finally, we approximate the time delay in this controller (e.g., using $e^{-\theta s} \approx 1 - \theta s$) to get a fixed-order controller. This becomes a PI or PID controller for a first- or second-order process G (Smith & Corripio, 1985) (Skogestad, 2003). Surprisingly, just by luck, the resulting PI- or PID-controller is generally better, or at least more robust with respect to changes in the time delay θ, than the Smith Predictor controller from which it was derived Grimholt & Skogestad (2018b).

An important advantage with these rules is that they contain a single adjustable tuning parameter:

$$\tau_c = \text{desired closed-loop time constant} \tag{14}$$

Following a step change in the setpoint, τ_c is approximately the time it takes (in addition to the process time delay θ) for the output $y(t)$ to reach 63% of the full change (because $1 - e^{-1} = 0.63$ in (13)). In some papers τ_c is called ϵ or λ. These direct synthesis (IMC) rules became a process industry standard in the 1990's as the "lambda tuning rules". However, lambda-tuning does not apply to integrating processes. To include also integrating processes, Skogestad (2003) proposed the SIMC PID-tuning rule, which is now widely used in industry.

The starting point for the SIMC PI-rule is to represent the process G as a first-order plus delay model from the MV (u) to the measured value of the CV (y):

$$G(s) = \frac{k}{\tau s + 1} e^{-\theta s} \tag{15}$$

This is a simplification for most real processes, but it has proven to be a very useful approximation for controller tuning, at least in the process industries. The model parameters are

$$k = \text{steady-state gain} = \frac{\Delta \text{CV}}{\Delta \text{MV}} \tag{16a}$$

$$\tau = \text{first-order time constant (63\%)} \tag{16b}$$

$$\theta = \text{effective time delay} \tag{16c}$$

We have written "effective" time delay because in most cases it is an approximation of higher-order dynamics. If the sampling time T is large, then it may affect the tunings, and we may add $T/2$ to the effective delay (Skogestad, 2003). It is also useful to introduce

$$k' = \frac{k}{\tau} = \text{initial slope of step response} \tag{17}$$

The SIMC-rule for a first-order with delay process (15) is a PI-controller with (Skogestad, 2003):

$$K_c = \frac{1}{k'} \frac{1}{\tau_c + \theta} \tag{18a}$$

$$\tau_I = \min\left(\tau, 4(\tau_c + \theta)\right) \tag{18b}$$

Here, the integral time $\tau_I = \tau$ follows by specifying a first-order response (12), but this value essentially turns off the integral action for slow or integrating processes with a large τ. To get acceptable rejection of disturbances entering at the process input for such cases, we choose $\tau_I = 4(\tau_c + \theta)$, which is the smallest τ_I that avoids the "slow" oscillations caused by having two integrators in series (one from the process and one from the controller).

For an integrating with delay process, $G(s) = \frac{k'}{s} e^{-\theta s}$, we have $\tau = \infty$, and the integral time is $\tau_I = 4(\tau_c + \theta)$.

For a static process ($\tau = 0$) with delay, $G(s) = ke^{-\theta s}$, the SIMC-rule gives a pure I-controller, $u(t) = K_I \int_0^t e(t)dt$, with integral gain $K_I = \frac{K_c}{\tau_I} = \frac{1}{k(\tau_c+\theta)}$. As mentioned, the Ziegler-Nichols tunings work poorly for such processes.

To achieve good robustness, it is recommended to select the tuning parameter larger than the effective time delay (Skogestad, 2003),

$$\tau_c \geq \theta \tag{19}$$

The lower bound $\tau_c = \theta$ is recommended for cases where one needs "tight control" and gives a gain margin (GM) of about 3. A gain margin of 3 may seem large, but it is actually not large for real implementations. A larger value for τ_c gives a smoother response with less input usage and better robustness margins. It is also possible to select τ_c less than the delay θ, although it is not normally recommended. For example, selecting $\tau_c = 0$ gives "very aggressive" control more similar to the Ziegler-Nichols tunings with GM about 1.5.

Example. *Consider a process with $k = 3, \tau = 6, \theta = 0$. Since there is no time delay, there are no robustness restrictions on the tuning parameter τ_c. To get a "speed-up" of a factor 1.5, we choose $\tau_c = 4$. Using* (18)*) this gives $K_c = (1/3)(6/4) = 0.5$ and $\tau_I = \min(6, 16) = 6$, as used earlier in* (8)*.*

Derivative action is normally only recommended for second-order processes, where the SIMC-rule gives $\hat{\tau}_D = \tau_2$ (this is for the series-form PID and the controller tunings need to be modified by the factor $\left(1 + \frac{\hat{\tau}_D}{\hat{\tau}_I}\right)$ when using the "ideal" form in (10)) (Skogestad, 2003).

With the SIMC PID rules, there is an almost linear relationship between τ_c/θ and the gain margin (GM). In particular, for processes where we use $\tau_I = \tau$ according to (18b), we have an exact linear relationship Grimholt & Skogestad (2012):

$$\mathrm{GM} = \frac{\pi}{2}\left(\frac{\tau_c}{\theta} + 1\right) \tag{20}$$

For example, with $\tau_c = \theta$ (”tight control”) we get GM $= \pi = 3.14$, and with $\tau_c = 3\theta$ we get GM $= 2\pi = 6.28$. For “slow” processes, where we use $\tau_I = 4(\tau_c + \theta)$ according to (18b)), the gain margin is a little smaller but it follows the same linear trend. The largest difference is for an integrating process where GM is about 0.18 lower than the value given in (20) for all values of τ_c/θ (Grimholt & Skogestad, 2012).

If it is important with very tight control for a first-order plus delay process (15), then one may use the “improved” SIMC PID-rule and add derivative action with $\hat{\tau}_D = \theta/3$ (series-form PID). One should then select $\tau_c = \theta/2$ (approximately) to get a performance benefit of the derivative action (Grimholt & Skogestad, 2018a); otherwise one only gets a robustness benefit. This “improved” PID-controller outperforms the Smith Predictor in most cases (Grimholt & Skogestad, 2018b). The word “improved” is put in quotes because the derivative action increases the input usage, so in most cases an engineer would prefer a PI-controller.

For noisy processes, one may add a filter F on the measurement of y, for example, a first-order filter (3) with time constant $\tau_F \leq \tau_c/2$ (preferably even smaller). With this filter, it is not necessary to have an additional filter on the possible derivative part.

4.3. Squeeze and shift rule

For what loops do we need “tight” control with a small value for τ_c, for example $\tau_c = \theta$? The answer is that tight control is usually most important when the output y should not exceed a hard constraint. “Hard” means that the constraint should not be violated, even dynamically. For hard constraints,

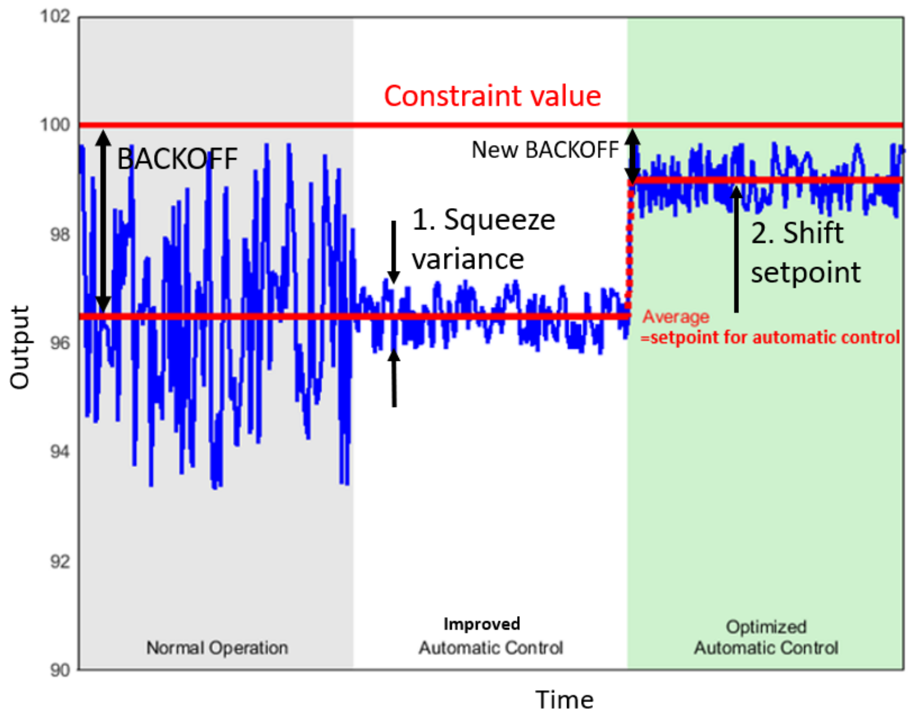


Figure 12: Squeeze and shift rule: Squeeze the variance by improving control and shift the setpoint closer to the constraint (reduce the backoff) to optimize the economics.

we need to introduce a "backoff" between the setpoint y_s and the constraint value, but by improving control we may reduce the backoff and save money. This is illustrated in Figure 12 and is known as the *"Squeeze and shift rule": Use improved control to squeeze (reduce) the variance for an output with a hard constraints in order to shift (move) the setpoint closer to the constraint value* (Richalet et al., 1978). For example, for a max-constraint, the backoff is defined as $B = y_{max} - y_s$. Any backoff from an active constraint will result in an economic loss, which can be quantified by $\lambda \cdot B$ where λ is the Lagrange multiplier (shadow price) for the constraint. The implications for controller tuning is that it is important to have tight control for hard constraints with a large shadow price λ.

If improved PID-tuning is not sufficient to reduce the output variations caused by disturbances, then some other improvement, such as cascade or feedforward control, should be considered.

4.4. Anti-windup (E8)

"Windup" is when the integrator term u_I in (10) grows out of bounds because the error e does not go to zero at steady state as expected. It occurs in a controller with integral action when changes in the controller output (MV or u) have no effect on the controlled variable (CV). The most common reason is that the physical input (e.g. valve) saturates at fully open or closed, but it may also occur when we use selectors or because of given limits on the controller output. There exists many industrial anti-windup schemes.

The simplest is to limit the allowed bias $b = u_0 + u_I$ (by adjusting u_0), or to limit the controller output u in (10) to be within specified limits (by adjusting u_0). These two options have the advantage that one does not need a measurement of the actual applied input value ($\tilde{u}$), and for most loops these suffice (Smith, 2010) (p. 21).

A better and also common anti-windup scheme is "external reset" (e.g., Wade (2004) Smith (2010)). This option is found in most industrial control systems and it uses the "trick" of realizing the integral action using positive feedback around a unit-gain first-order process with time constant τ_I.[3]. With this implementation, anti-windup is easily achieved by replacing the positive feedback from u with the actual applied value ($\tilde{u}$).

4.5. Anti windup with back calculation (E8)

The “external reset” solution is a special case of the further improved “back-calculation” scheme in Figure 11 which is recommended by Åström & Hägglund (1988). The “back-calculation” scheme has a very useful additional design parameter, namely the “tracking” time constant τ_T, which tells how fast the controller output u tracks the actual applied value $\tilde{u}$. This makes it possible to handle more general cases in a good way. e.g., switching of CVs. In the simpler “external reset” scheme, the tracking time is “by design” equal to the integral time ($\tau_T = \tau_I$) (Åström & Hägglund, 1988).

To better understand the recommended “back-calculation” scheme, note that we from Figure 11 get for a one degree-of-freedom PID controller,

$$u(t) = K_c e(t) + K_c \tau_D \frac{de(t)}{dt} + \underbrace{\overbrace{\int_{t'=t_0}^{t} \left(\frac{K_c}{\tau_I} e(t') + \frac{1}{\tau_T} e_T(t') \right) dt'}^{u_I} + u_0}_{\text{bias}=b} \tag{21}$$

The tracking signal $e_T = \tilde{u} - u$ is fed to the input of the integrator through the gain $1/\tau_T$. The signal is zero when the controller is connected to the process so that $\tilde{u} = u$. Thus, it has no effect under normal operation. However, when the actuator saturates (or more generally when the controller is disconnected from the process), a new feedback path is vreated to track $\tilde{u}$ which stops the “windup” of the integrator output b. A smaller tracking time means that the tracking of $\tilde{u}$ is better, which means that the controller activates sooner when the constraint is reached. The disadvantage with a too small tracking time is that it may activate the controller unnecessary.

[3]Note that $\frac{1}{1-\frac{1}{\tau_I s+1}} = \frac{\tau_I s+1}{\tau_I s} = 1 + \frac{1}{\tau_I s}$

To understand this better, assume that we have saturation and that u_{lim} is the saturated (actual) value of u, that is $\tilde{u} = u_{lim}$. At steady state, the integrator input $\frac{K_c}{\tau_I}e + \frac{1}{\tau_T}e_T$ is zero (but note that this does not mean that the integrator output u_I is zero), and we have at steady state that

$$e_T = u - u_{lim} = K_c \frac{\tau_T}{\tau_I} e \tag{22}$$

Note that $e = y_s - y$ is nonzero (and out of our control) when u is disconnected from the process. We see that a small τ_T means that the tracking is better, with u closer to u_{lim}. This may be an advantage because the controller activates sooner. On the other hand, a too small value of τ_T is not desired because it may activate the controller when it is not necessary, because there will always be some "nervous" variations in $u(t)$ due the effect of output variations caused by disturbances and measurement noise on the proportional and derivative terms.

As mentioned, it is common to choose the tracking time equal to the integral time ($\tau_T = \tau_I$). With this value, we get at steady state that the output from the integral part (u_I) is such that the bias b is equal to the constraint value, $b = u_{lim}$. To derive this, note that with $de/dt = 0$ (steady state), (21) gives $u = K_c e + b$ which combined with (22) and $\tau_T = \tau_I$ gives $b = u_{lim}$. For a PI-controller, (21) gives $u(t) = K_c e(t) + b$ (also dynamically), which means that with $\tau_T = \tau_I$, the controller will activate u (i.e, go out of saturation) if the control error e jumps to 0, that is, if y reaches its setpoint y_s. However, this may be too conservative and Åström & Hägglund (2006) say that the value $\tau_T = \tau_I$ is often too large. A reasonable choice in many cases is $\tau_T = \tau_I/2$. Even smaller values were suggested by Markaroglu et al. (2006) but they did not include disturbances and measurement noise which may cause the system to go prematurely out of saturation if τ_T is chosen too small.

Anti-windup and the choice of tracking time is further discussed in Sections 5.9 and 6.5.2.

4.6. Bumpless transfer

Bumpless transfer means that we have a smooth transition between different operating modes of the controller. In most cases this is automatically taken care of by the anti-windup, at least if we use the back-calculation scheme.

However, when switching from manual to automatic control, we may get a "bump". This may happen even with anti-windup using back-calculation, because u does not track the manual input $\tilde{u} = u_{man}$ perfectly. A simple solution is to update u_0, so that u computed from (21) is equal to u_{man} at the time of switching. It may be convenient (but not necessary) to restart the integration (by setting t_0= time of switching) so that $u_I = 0$ at the time of switching.

4.7. On-off control

The most common example of on/off control is a thermostat used for heating or cooling in buildings. On-off controllers are common in industry, both because they are simple and because some units should be operated in an on-off fashion, for example, a vacuum or refrigeration system. Essentially, an on/off-controller works as a P-controller with infinite gain, and the main disadvantage is that it will always cycle around the given CV setpoint (switching value). Because of the infinite gain, there is no steady-state offset (on average), which also means that no anti-windup scheme is needed.

To reduce the frequency of cycling one may instead of a fixed setpoint for the CV (controller input) give a setpoint band (low and high setpoint). The controller will then display hysteresis, with two possible controller outputs (e.g., 0 or 1) when the CV (controller input) is within the specified setpoint band. An example of on/off control with a setpoint band for inventory (level) control is shown in the flowsheet in Figure 36.

5. Advanced regulatory control elements

This section describes in more detail some of the "classical" control elements that are used in industry for "advanced regulatory control".

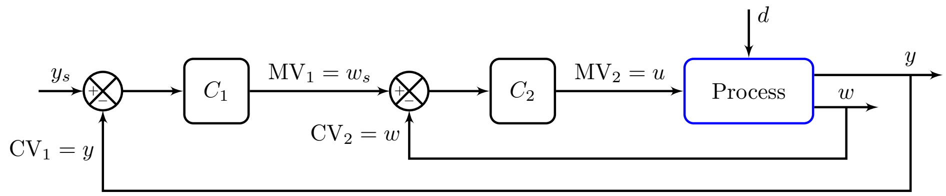


Figure 13: General cascade control scheme with outer (primary) controller C_1 (slow) and inner (secondary) controller C_2 (fast). All blocks are possibly nonlinear.
The objective of the control system is to keep the output y close its setpoint y_s in spite of disturbances d. The extra (secondary) measurement w is controlled on a fast time scale, with the objective of improving the control of y.

A fairly general cascade implementation is shown in Figure 13. The primary controller C_1 has as its manipulated variable (MV_1) the setpoint (w_s) to the secondary (”slave”) controller C_2. Common slave loops in process control involve flow, pressure or temperature (i.e., $w = F$, $w = p$ or $w = T$). Cascade control is a very powerful and simple method. The idea is that fast control of the (extra) measurement w will indirectly benefit the control of y.

To better understand the advantages of cascade control, consider the special series process in Figure 14. Here w is an intermediate (secondary) measurement which directly affects the primary output y through the primary process G_1.

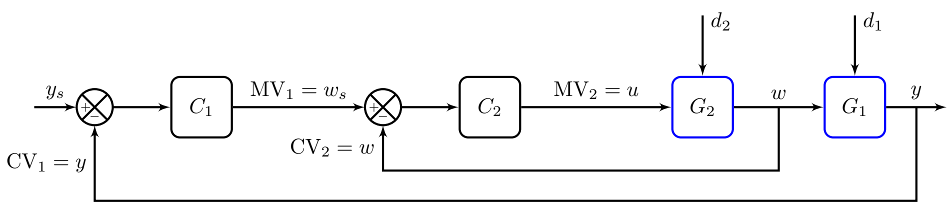


Figure 14: Cascade control for series process where the objective is to control y and w is an intermediate measurement. All blocks are possibly nonlinear.
· C_1=primary/outer controller (slow), G_1 =primary process
· C_2=secondary/inner controller (fast). G_2 =secondary process

An early and very good description of the benefits of cascade control is given by Shinskey (1967). With reference to Figure 14, he writes (p. 154):

The principal advantages of cascade control are these:

1. Disturbances arising within the secondary loop are corrected by the secondary controller (C_2) before they can influence the primary variable (y).
2. Phase lag existing in the secondary part of the process (G_2) is reduced measurably by the secondary loop. This improves the speed of response of the primary loop.
3. Gain variations in the secondary part of the process (G_2) are overcome within its own loop.
4. The secondary loop permits an exact manipulation of the flow of mass or energy (w) by the primary controller.

Tuning of the two controllers should be done sequentially, and it is strongly recommended to use a design method (e.g. SIMC PID-tuning) where the closed-loop time constants τ_{c1} and τ_{c2} are used as design parameters. The inner (secondary) controller C_2 (fast) is tuned first based on the process G_2, and with this loop closed, the outer (primary) controller C_1 (slow) is tuned. For the case with a series process (Figure 14), the tuning of C_1 may be done based on the process G_1 with an added effective delay $\tau_{c2} + \theta_2$ to represent the inner loop. As given by the rule of thumb in Section 3.4, the time scale separation τ_{c1}/τ_{c2} between the loops should typically be between 4 and 10. A larger time separation helps to protect against process gain variations in both the inner and outer loops, but it "eats up" more of the available time window. To avoid eating up the time window, the solution is to tune the inner loops more tightly.

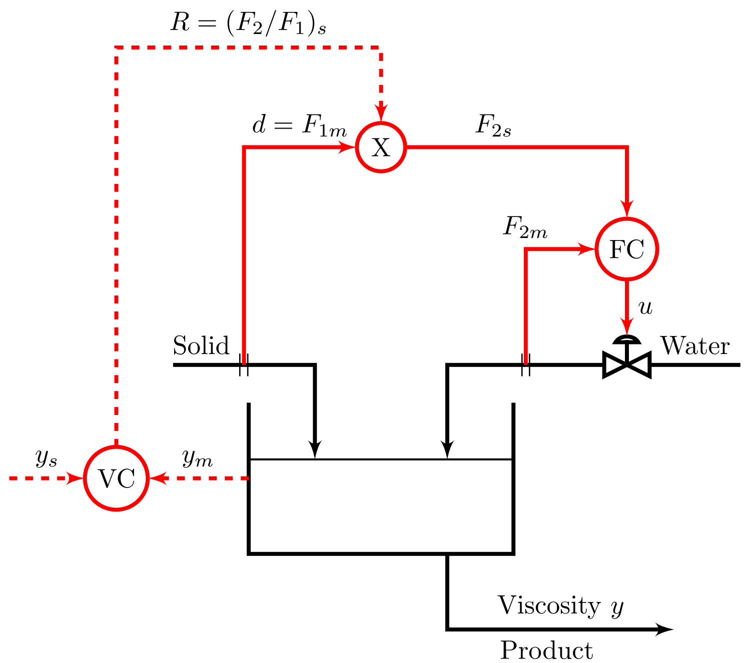


Figure 15: Ratio control with feedback correction (trim).
Flowsheet of continuous mixing process with control of property y (here viscosity). The ratio control is shown with red solid lines. The ratio block (x) multiplies the measured flow disturbance $d = F_1$ with the desired flow ratio R to get the desired input flow F_{2s}. An inner flow controller (FC) with $u = z$ (valve position) is needed to implement more accurately the desired flow F_{2s}. The ratio controller is combined with an outer feedback viscosity controller VC (red dashed lines) which adjusts the ratio setpoint $R = (F_2/F_1)_s$ in order to make $y = y_s$ at steady state.
To maintain the steady-state mass balance, the product outflow is given by a level controller (not shown on the flowsheet).

5.2. Ratio control (E2)

5.2.1. Implementation with multiplication element

A typical ratio control scheme is shown in the flowsheet [4] in Figure 15. This is a mixing process where we mix a solid with water, and ratio control is based on the physical insight that to keep a mixture property (intensive variable) y constant, we need a constant feed flow ratio $R = F_2/F_1$. To implement this, we measure the solid flowrate $d = F_1$ (a disturbance, sometimes called a "wild" flow) and multiply it by the desired ratio R to get the desired water flowrate (process input),

$$F_{2s} = R \cdot F_1$$

In the flowsheet in Figure 15 this corresponds to the multiplication block (x). The setpoint F_{2s} goes to an inner (fast) flow controller which gives $F_2 = F_{2s}$ at steady state. Note that ratio control involves "absolute" flows, and not deviation variables as is often used in block diagrams. Also note that we have implemented ratio control using a multiplication element. One should avoid using a division element because of the danger of dividing by zero.

5.2.2. Feedback trim

In Figure 15 we also have included a feedback adjustment (trim) of the ratio setpoint. We use an outer viscosity controller (VC)[5] which finds by feedback ("trial and error") the correct ratio R which makes the measured viscosity y equal to its setpoint y_s. This kind of feedback correction ("feedback trim") is very common and it avoids the need for having a model for how y depends on

[4] A flowsheet with controllers (or Process & Instrumentation Diagram, P& ID) is an alternative to a block diagram for graphically representing the control system interconnections (signals). However, a flowsheet also shows the process interconnections (usually pipelines). This is different from a block diagram where *all* lines are signals. For the flowsheets in this paper, the solid black lines represent the process interconnections (pipelines). The control signals are shown by dashed or red lines.

[5] In a flowsheet, a controller is written as XC where X tells what kind of variable the CV is, for example, FC is flow control, PC is pressure control, TC is temperature control, LC is level control and IC is inventory control (which usually is level or pressure), These are single-loop controllers with the CV-setpoint and CV-measurement as input signals and the MV as the output signal.

the inputs and disturbances. For example, consider making food, where we first mix the ingredients according to the ratios given in the recipe, and then we fine-tune the ratios based on feedback from an intensive property variable such as taste, color, texture or "thickness" (viscosity).

5.2.3. Theory of ratio control

Ratio control is most likely the oldest control approach (think of recipes for making food or chemical compounds), but despite this, no theoretical basis for ratio control has been available until recently (Skogestad, 2023). Note that with ratio control, the controlled variable y is implicitly assumed to be an *intensive variable*, for example, a property variable like composition, density or viscosity, but it could also be a temperature or pressure.

Ratio control is more powerful than most people think, because its application only depends on a "scaling assumption" and does require an explicit model for y. For a mixing process, the "scaling property" or "scaling assumption" says if all flows are increased proportionally (with a fixed ratio), then at steady state all mixture intensive variables y will remain constant (Skogestad, 1991). The scaling property (and thus the use of ratio control) applies to many process units, including mixers, equilibrium reactors, equilibrium flash and equilibrium distillation. For distillation we must assume that the pressure and efficiency (number of theoretical stages) is constant.

More generally, the scaling property requires that *all extensive variables* (flows, heat rates, sizes of certain equipment) must be scaled by the same factor. Thus, ratio control should not be used if we have saturation in a flow, even if this a unit where the scaling property holds. To have perfect ratio control, we must also require that all independent intensive variables (typical feed composition and temperature) are kept constant, but this is not a critical requirement if we have an outer feedback loop.

There are also many process units where ratio control should not be used, because the scaling property does not hold. This includes, for example, non-equilibrium reactors (where kinetics are important) and heat exchangers. For the scaling property to hold for a heat exchanger, we would need to increase the heat transfer area A proportionally to the flow rates. This is reasonable during design but not during operation (control) when the equipment is fixed.

Ratio control may be viewed as a special case of feedforward control (and decoupling in some cases), but note that we do not need a model for the property y for ratio control, whereas such a model is needed for feedforward control or decoupling and more generally for other model-based schemes, including MPC.

5.2.4. Summary ratio control

Ratio control is very simple to use and it gives nonlinear feedforward action without needing an explicit process model. It is almost always used for chemical processes to set the ratio of the reactant feed streams. This is a mixing process where the scaling assumption clearly holds. However, as mentioned above, ratio control can also be used effectively in many other applications.

Since ratio control is difficult to implement with MPC (also see discussion section), it should also be included when using MPC and MPC then sets the ratio setpoints.

5.3. Selective (limit) control (E3)

Selectors are used for CV-CV switching, which is when one MV (u) is used to control many CVs ($y_1, y_2, \dots$), but only one CV should be controlled at a time. CV-CV switching is frequently used for satisfying inequality constraints. When a new CV constraint is encountered, one stops controlling the present CV (either because the constraint on the present CV becomes over-satisfied or because the present CV can be given up) and switches to the new CV. Some alternative symbols for selectors are shown in Figure 16.

The most general implementation for CV-CV switching is to have one controller for each CV with a selector on the MV as shown in Figure 17 (Reyes-Lúa & Skogestad, 2020b). It may seem surprising that the selector is on the MV, when it is the CV that reaches a constraint, but it turns out to be a very powerful approach.

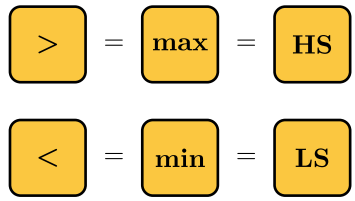


Figure 16: Alternative symbols for selector block. Each selector block has two or more inputs, but only one output. Selector blocks may also be put in series.
HS= high select, LS = low select.

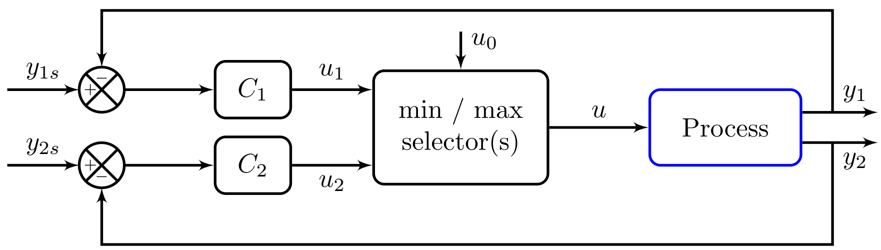


Figure 17: CV-CV switching: Selective control with selector on MV (input u).
Here, y_{1s} and y_{2s} may be constraint values or desired setpoints, whereas u_0 (if used) may be a desired value which may be given up. The block "min/max selector(s)" may be a max- or a min-selector (Rule 1), or a max- and min-selector in series (with order as given by Rule 2),

Note that we have a "single-input-multi-output" (SIMO) process, but this is not "conventional" SIMO control, which usually refers to controlling multiple CVs in some weighted or average manner using a single controller, e.g., Freudenberg & Middleton (1999). In CV-CV switching we have multiple controllers which are working one at a time.

CV-CV switching is sometimes called override control, but this term may be misleading because it gives the impression that we are making some undesired "fix" to the solution. On the contrary, in most cases the CV-CV switching ("override") is the optimal solution at the present operating point (with a given disturbances). This is an important point, because many people tend to dismiss selectors as being some ad-hoc industrial method, but as discussed in Section 10.4, selectors (or something with a similar function) are required for optimal steady-state operation.

For the design of selector structures, the following two rules are helpful (Krishnamoorthy & Skogestad, 2020):

Selector Rule 1. Max or Min selector *(applies to selector on MV, see Figure 17):*

- *Use a max-selector for constraints that are satisfied with a large MV (u).*
- *Use a min-selector for constraints that are satisfied with a small MV (u).*

If all constraints require the same selector (max or min), then only one selector block is needed. For example, in Figure 17, we use $u = \min(u_0, u_1, u_2)$ if both constraints y_{1s} and y_{2s} are satisfied by a small u, and we use $u = \max(u_0, u_1, u_2)$ if both constraints y_{1s} and y_{2s} are satisfied by a large u.

Selector Rule 2. Order of Max and Min selector(if both are needed)*: If the constraints require different selectors, then max- and min-selectors in series are needed with* u_0 *(which may be given up) entering the first selector. In this case, there is a possibility for conflict (infeasibility), and the highest priority constraint should enter the last constraint block.*

For example, in Figure 17 we should use a max-selector followed by a min-selector, $u = \min(u_2, \max(u_0, u_1))$, if constraint y_2 (with highest priority) is satisfied with a small u and constraint y_1 (with lower priority) is satisfied with a large u. This can be implemented as shown in Figure 18.

The main limitation with the selector approach described in this section is that each CV-constraint must be associated with a given MV. If there are more CV-constraints than MVs, then several constraints need to be associated with one MV. This will not cause any problem as long as they are all satisfied either by a small MV (using a min-selector) or a large MV (using a max-selector). However, if both a max- or min-selector is required for the same MV then we have a potential feasibility problem. For example, in Figure 18, we may need to give up on the constraint on y_1, if y_2 reaches its constraint y_{2s}. If giving up y_1 is not acceptable, then we need to find another MV for y_1 and some additional

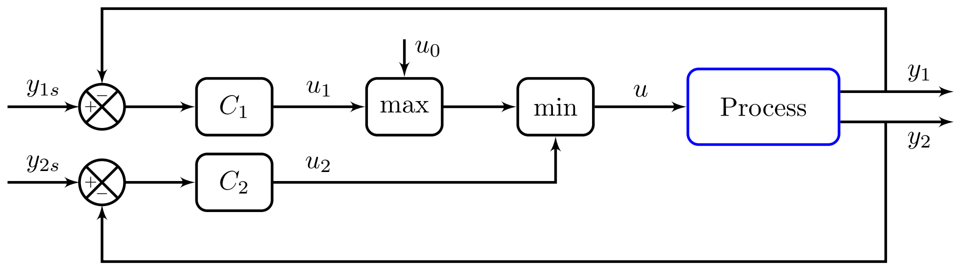


Figure 18: CV-CV switching. Example of case when the constraints on the CVs (y_{1s} and y_{2s}) require different selectors (max and min). In this case, the constraints may be conflicting, so the selector block corresponding to the most important constraint (here y_{2s}) should be at the end (Rule 2).

To understand the logic with selectors in series, start reading from the first selector. In this case, this is the max-selector: The constraint on y_1 is satisfied by a large value for u which requires a max-selector (Rule 1). u_0 is the desired input for cases when no constraints are encountered, but if y_1 reaches its constraint y_{1s}, then one gives up u_0. Next comes the min-selector: The constraint on y_2 is satisfied by a small value for u which requires a min-selector (Rule 1). If y_2 reaches its constraint y_{2s}, then one gives up controlling all previous variables (u_0 and y_1) since this selector is at the end (Rule 2). However, note that there is also a "hidden" max- and min- selector because of the possible saturation of u, so if the MV (input) saturates, then all variables (u_0, y_1, y_2) will be given up.

logic is needed. In some cases, this logic may be quite simple (for example, using split range control for MV-MV switching), but in other cases it may not be possible to find a simple logic scheme, and an model-based solution (MPC) may be simpler.

An alternative (and somewhat less general) cascade implementation of CV-CV switching with the selector on the *setpoint* is shown in Figure 19 (Cao, 2004). As usual with cascade control, this solution is recommended for cases where fast control of y_2 benefits the control of y_1. If the setpoint y_{2s} to the inner loop is a constant (for example, a constraint), then the selector block in Figure 19 may be replaced by a saturation element (Cao, 2004). However, note that the constraint does not need to be in the inner loop as suggested by Cao (2004); it could also be the setpoint to the outer loop if this cascade arrangement is better from a dynamic point of view. The reason why the cascade implementation is said to be "somewhat less general" is because the design of the outer controller depend on the tuning of the inner controller and will have to be "slow" because of the requirement of time scale separation. As an example, consider adaptive cruise control (Section 6.3) where the cascade implementation is not recommended.

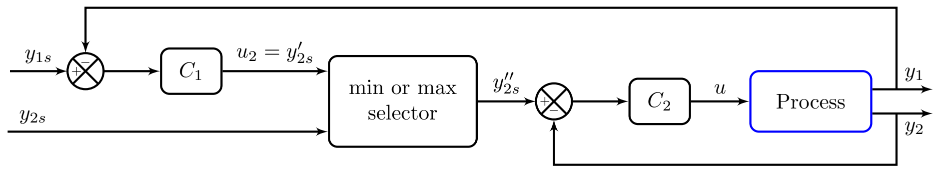


Figure 19: CV-CV switching. Alternative cascade selector implementation with selector on the setpoint.

There is also a third (and much less general) case of CV-CV switching (not shown in any figure), where the selector is on the measurement of y and the controller comes afterwards. This is fairly common and used when all the CVs (y_i) have the same constraint value (y_s). For example, it is used if want to avoid that the maximum temperature (”hotspot”) in a reactor, $y = \max(y_1, y_2, \dots)$, exceeds $y_s = y_{max}$. This solution is sometimes referred to as *auctioneering* Shinskey (1967).

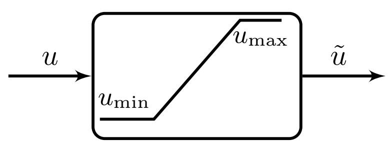


Figure 20: Saturation element (limiter) to represent amplitude limits, for example, for a valve. It is equivalent to a min- and a max-selector in series or to a mid-selector, see (23)

Finally, we have the most common case of ”built-in” selectors for physical inputs (final control elements), for example valves, pumps, etc.. These have a maximum and minimum value which cannot be violated, and may be represented by a saturation element (limiter) with a max- and min-value as shown in Figure 20. As given by the following rule, this implies that all physical inputs have “built-in” (implicit) max- and min-selectors.

Selector Rule 3. Physical inputs have built-in selectors (Figure 20):

- *A low input limit, $u \geq u_{min}$, corresponds to a “built-in” max-*

selector, $\tilde{u} = \max(u, u_{min})$.

- *A high input limit,* $u \leq u_{max}$*, corresponds to a "built-in" min-selector,* $\tilde{u} = \min(u, u_{max})$.

The saturation element in Figure 20 is equivalent to a max- and min-selector in series (in any order) or to a mid-selector:

$$\tilde{u} = \text{mid}(u_{min}, u, u_{max}) = \max(u_{min}, \min(u_{max}, u)) = \min(u_{max}, \max(u_{min}, u)) \tag{23}$$

The order of the "built-in" max- and min -selector in (23) does not matter because there is no possibility for conflict, as the two constraints (limits), u_{min} and u_{max}, cannot be active at the same time. However, in general, the order of the selectors does matter, and in cases of conflict, Rule 2 says that we should put the most important constraint at the end. Note that the "built-in" max- and min-selector of the physical input (valve) always come at the end, so there is always a danger that a CV constraint cannot be satisfied because of input saturation. In such cases, if the CV constraint cannot be given up, one of the schemes for MV-MV switching has to be implemented.

In some cases, the functioning of a control solution depends on having these "built-in" input selectors, and to show this more clearly we will include saturation element in the block diagram for such cases, e.g. see Figure 25.

Some physical inputs may also have a "built-in" rate (derivative) limiter. For example, a valve may have an electric motor that moves the valve with a maximum speed. More generally, limiters on the amplitude or the rate may be added by the designer, for example, to avoid that an outer controller generates a setpoint outside the range that the system can cope with (Åström & Hägglund, 2006).

5.4. Input (valve) position control (VPC) to improve the dynamic response (E4)

5.4.1. Common case: VPC with two MVs

Consider a "multi-input-single-output" (MISO) process with two MVs (inputs, u_1, u_2) and one CV (y), but only one MV (u_2) is used for steady-state

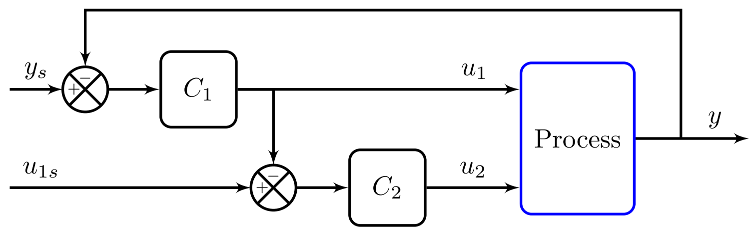


Figure 21: Valve (input) position control (VPC) for the case when an "extra" MV (u_1) is used to improve the dynamic response.
C_1 = fast controller for y using u_1.
C_2 = slow valve position controller for u_1 using u_2 (always operating).
u_{1s} = steady-state resting value for u_1 (typically in mid range. e.g. 50%).

control. The other MV (u_1) is an "extra" input (for example, a bypass stream) which is used to improve dynamically the control of the CV (y), but on a longer time scale u_1 should be reset to a desired setpoint u_{1s}. A common solution is to use valve (input) position control as shown in Figure 21. This solution is also known as mid-ranging control (Allison & Isaksson, 1998) (Åström & Hägglund, 2006) or input resetting. The fast controller (C_1) is tuned first and next the slower valve position controller (C_2). This is a cascaded scheme, so as discussed earlier the time scale separation between the two loops should typically be in the order 4 to 10. Allison & Ogawa (2003) discuss tuning of the PI-controllers, and they say that C_2 is frequently an I-only controller. Both controllers usually have integral action, but Åström & Hägglund (2006) notes that anti windup is not needed for C_1 since its input u_1 is controlled by the slower valve position controller C_2. For cases where the controller C_2 "disturbs" the controlled variable y (which is likely if the time scale separation is small), they suggest introducing one-way decoupling from u_2 to u_1.

5.5. Alternative to VPC: Parallel control

An alternative solution is to use "parallel control" (Figure 22) where both C_1 and C_2 control the same y. However, only one of the controllers should have integral action (Balchen & Mumme, 1988). More precisely, to make sure that the input u_1 returns to u_{1s} at steady state, the loop involving C_2 must have one more integrator than the loop involving C_1. Usually, this means that C_2 is a PID-controller and C_1 is a P- or PD-controller with the bias set at u_{1s}, see Figure 22.

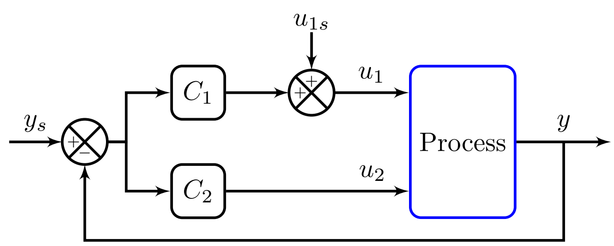


Figure 22: Parallel control as an alternative to the VPC solution in Figure 21. The “extra” MV (u_1) is used to improve the dynamic response, but at steady-state it is reset to u_{1s}. The loop with C_2 has the main integral action and wins a steady state.

The advantage with valve position control compared to parallel control, is that the two controllers in Figure 21 can be tuned independently (but C_1 must be tuned first) and that both controllers can have integral action. On the other hand, with some tuning effort, it may be easier to get good control performance for y with parallel control.

5.5.1. VPC with one MV (stabilizing cascade control)

A different application of VPC is when we use the input u dynamically to stabilize the system, but on a longer time scale u is reset to a desired setpoint u_s. This can be realized with a cascade control system (Figure 23) (Storkaas & Skogestad, 2004). The inner fast controller (C_2) manipulates u to control (”stabilize”) the measurement w_1, and the outer slow valve position controller (C_1) manipulates $w_{1,s}$ to reset u to its desired setpoint u_s. This means that we have $y = u$ for the outer loop. In Figure 23 we have also added an inner flow controller C_3 (very fast), but this is not generally needed.

A common application is to ”stabilize” (stop drift of) pressure by controlling $w_1 = p$ on a fast time scale, but on a longer time scale pressure is “floating” because the VPC manipulates p_s. Applications of “floating pressure” operation are found in steam systems and distillation columns (Shinskey, 1979) (Wade, 2004).

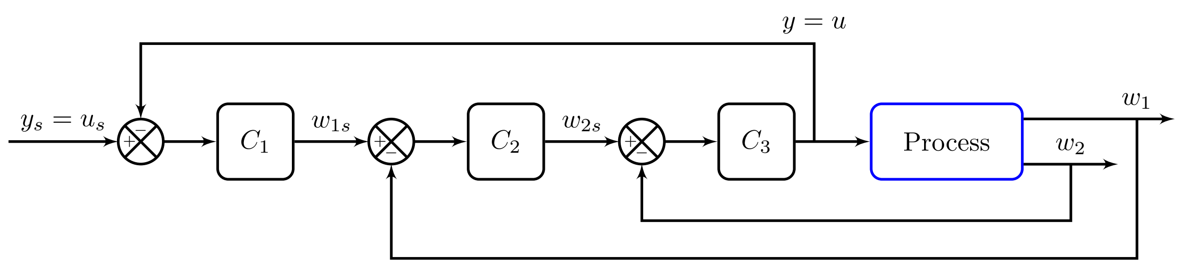


Figure 23: Stabilizing cascade control with input resetting (VPC).
Note that the process variables (w_1, w_2) have no fixed setpoint, so they are "floating".
It corresponds to the flowsheet in Figure 24 with u = valve position, $w_1 = p$ (pressure), $w_2 = F$ (flow), C_1 = outer VPC (slow) , C_2 = stabilizing controller (fast), C_3 = inner flow controller (very fast) (not needed).

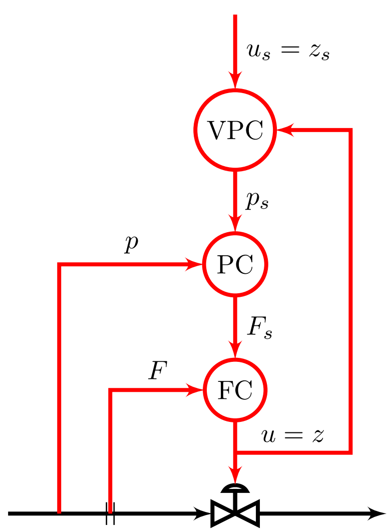


Figure 24: Flowsheet for anti-slug control where pressure controller (PC) is used for stabilization. The inner flow controller (FC) (fast) provides linearization and disturbance rejection. The outer VPC (slow) resets the valve position to its desired steady-state setpoint ($u_s = z_s$).

5.5.2. Example VPC with one MV: Stabilizing anti-slug control

An application for stabilizing multiphase flow (Storkaas & Skogestad, 2004) is shown in the flowsheet in Figure 24. It corresponds to the block diagram in Figure 23. As the oil field ages and more gas is produced, we may enter an undesirable flow regime with "severe slugging". The objective is to stabilize the non-slug flow regime[6] by using a pressure controller (C_2 = PC). An inner flow controller (C_3 =FC) is added to linearize the valve and reduce fast disturbances. The outer valve position controller (C_1 = VPC) manipulates the pressure setpoint (p_s) to bring the valve position back to its desired steady-state position (z_s). For this application, an almost fully open valve ($z_s = 80\%$) may be preferred to maximize the production rate (F).

Note that this is a cascade control system, where we need at least a factor 4 (and preferably 10) between each layer. This implies that the outer VPC (C_1) must be at least 16 (and preferably 100) times slower than the inner flow controller (C_3). This may not be a problem for this application, because flow controllers can be tuned to be fast, with τ_c less than 10 seconds (Smuts, 2011). Another more fundamental problem is that any unstable mode (RHP pole) in the process will appear as an unstable (RHP) zero as seen from the VPC (C_1) (Storkaas & Skogestad, 2004), and this will limit the achievable speed (bandwidth) of the outer loop.

5.6. Split range control for MV-MV switching (E5)

Consider a "multi-input-single-output" (MISO) process with many MVs ($u_1, u_2, \dots$) and one CV (y), where all the MVs are needed for steady-state control, but we want to use them one at a time in a specific order (first u_1, then u_2, etc.). This is the case of MV-MV switching, for which the oldest approach is split-range control (Eckman, 1945) as shown in Figure 25. An example is when

[6] Ant-slug control is a bit similar to attempting to stabilize laminar flow at high Re-numbers where one normally expects turbulence, However, stabilizing laminar flow is a much more difficult control problem as the transition between flow regimes happens much faster; although it may be possible, for example, with distributed actuators that locally manipulate the diameter of a flexible pipeline.

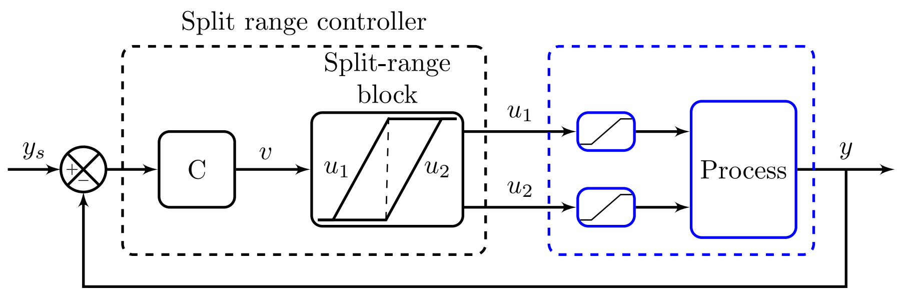


Figure 25: Split range control for MV-MV switching.

we want to control the temperature ($y = T$) using two sources of heating, for example, hot water (u_1) and electric heat (u_2). Since u_1 is cheaper, it should be used first as illustrated in the split-range block in Figure 25. In Figure 25, there is only one controller C which computes the internal variable v that enters the split range block. This means that we with split-range control need to use the same integral and derivative times for all MVs ($u_1, u_2, \dots$). Fortunately, the (effective) controller gain can be made different for each MV by moving the transition point for v (dashed vertical line in the split-range block), such that the slopes (gains) from v to each u_i become different (Reyes-Lúa et al., 2019).

The limitation in terms of tuning (same integral and derivative time for all MVs) can be avoided by using generalized split range control (Reyes-Lúa & Skogestad, 2020a) but this requires additional logic and is more complicated to implement.

5.7. Separate controllers (with different setpoints) for MV-MV switching (E6)

.

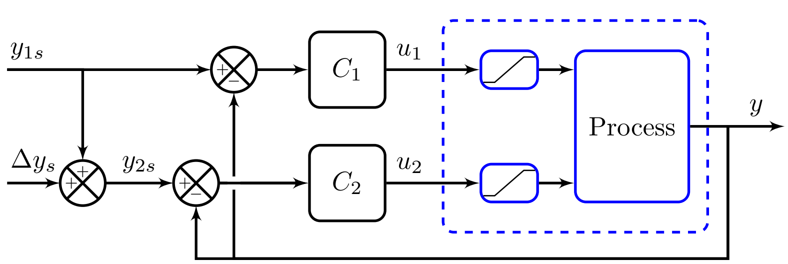


Figure 26: Separate controllers with different setpoints for MV-MV switching.

Consider again MV-MV switching where we want to use one MV at a time in a specific order (first u_1, then u_2, etc.). An alternative to split range control is to use separate controllers for each MV with different setpoints (Figure 26) (Smith, 2010) (Reyes-Lúa & Skogestad, 2019).

The setpoints should be ordered in the same order as we want to use the MVs. The setpoint difference Δy_s (Figure 26) should be large enough such that only one controller (with associated MV) is active at a given time, while the other MVs are at their limits. This solution has two important advantages compared to split range control in Figure 25. First, the controllers $(C_1, C_2, \dots)$ can be designed independently for each MV, whereas in split range control there is a single controller C. Second, and probably more importantly, one avoids in Figure 26 the need to include the MV limits $(u_{1,min}, u_{1,max}, u_{2,min}, \dots)$ which are needed in the split range block in Figure 25. Instead, any saturation limit (or similar) is detected indirectly by feedback through the loss of control of the CV (y), and the next MV will take over (after some transition time) when the CV reaches the next setpoint. This indirect detection is a big advantage if the switching does not occur at a fixed MV-value, for example, when a selector (for CV-CV switching) takes over the MV. The solution in Figure 26 is therefore very flexible and is preferred for the case of complex MV-CV switching.

The main disadvantage with separate controllers is the difference in setpoints. First, this means that control of y is temporary lost during MV-MV switching. Thus, this solution is not recommended for cases where MV-MV switching occurs frequently or where tight control of y is needed. Second, the setpoint is not constant, because $y = y_{1s}$ when we use u_1, whereas $y = y_{2s} = y_{1s} + \Delta y_s$ when we use u_2. The last disadvantage can be avoided (at least at steady state) by using the implementation in Figure 27. Here, a slower outer loop (C_0) controls y to a fixed setpoint y_s by manipulating (resetting) the setpoint y_{1s} in a cascade manner. The setpoint difference(s) Δy_s is kept unchanged.

However. the setpoint difference can also be an (economic) advantage in some cases. For example, if the two inputs for temperature control are heating

(u_1) and cooling (u_2), then we may be willing to accept a lower setpoint (say, $y_{1s} = 21$C) in the winter than in the summer (say, $y_{2s} = 23$C) to save energy (and money) for heating and cooling (Reyes-Lúa & Skogestad, 2019).

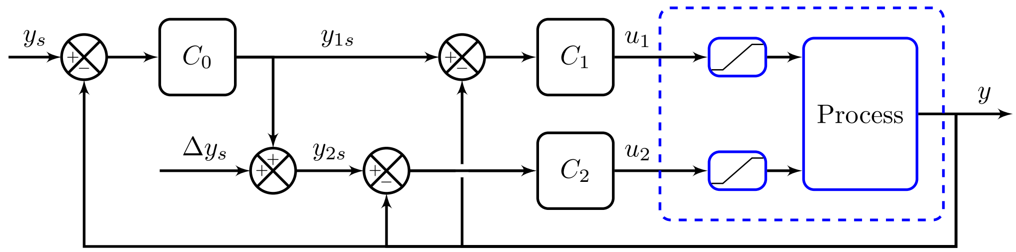


Figure 27: Separate controllers for MV-MV switching with resetting of setpoint.
This is an extension of the scheme in Figure 26, where a slower outer controller C_0 resets y_{1s} to keep a fixed setpoint $y = y_s$ at steady state.

5.8. VPC for MV-MV switching (E7)

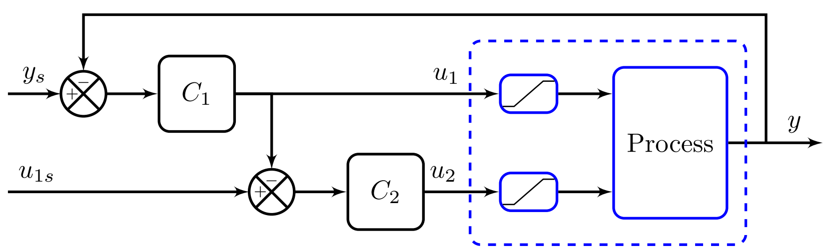


Figure 28: Valve (input) position control for MV-MV switching.
u_{1s} = value of u_1 where we to switch to using u_2 (typically at 90%).
C_2 = valve position controller (only operating when u_1 reaches u_{1s}; otherwise u_2 is at its constraint, typically $u_2 = u_{2,min} = 0$.).

Consider yet again MV-MV switching, and assume that we for dynamic reasons would like to always use u_1 to control y. We cannot let u_1 become fully saturated because then control of y is lost, but we can use the other inputs ($u_2, \dots$) to avoid that u_1 saturates. This can be realized using valve position control as shown in Figure 28.

The main advantage with the VPC scheme (Figure 28) compared to the two alternative schemes for MV-MV switching (split range control in Figure 25 and multiple controllers in Figure 27) is that the same input (u_1) is always used to control y. The disadvantage is that when u_2 is used, we also need to use a "little" of u_1. This is a disadvantage both economically and in terms of utilizing the whole range for u_1. For example, if the two MVs (inputs) for temperature control are heating (u_1) and cooling (u_2), then VPC (Figure 28) requires that we also use a little heating when we actually need cooling.

The VPC solution for MV-MV switching (Figure 28) is expected to be the preferred solution in the following cases

- When the input u_2 is only rarely needed for control of y.
- When u_2 is not suited for control of y, for example if u_2 is an on-off input (e.g., a pump with constant speed).

Comment 1 on VPC. The two valve position schemes in Figures 21 and 28 seem to be the same, but actually their behavior is very different. In Figure 21 (VPC for improved dynamic control) we expect no saturation of the inputs u_1 and u_2. On the other hand, in Figure 28 (VPC for MV-MV switching) we have that either u_2 is saturated (typically $u_2 = u_{2min} = 0$) or that u_1 is almost saturated (e.g., $u_1 = u_{1s} = 0.9$).

Comment 2 on VPC. A valve position controller (VPC) should not be confused with a *valve positioner* (Smith, 2010) (p. 178). The latter is an inner (fast) cascade controller which is delivered by the valve manufacturer together with the valve. A valve positioner is usually a high-gain P-controller which ensures that the valve position z desired from another controller is equal to the actual measured valve position.

5.9. Anti-windup for selective and cascade control (E8)

In this paper, we recommend anti-windup with back-tracking as given in Eq. (10) and Figure 11. In general, anti-windup needs to be implemented for controllers with integral action for cases where the MV (= controller output = u in Figure 11) is disconnected for some time from the remaining system. Three common cases are

1. Input saturation for the MV
2. Selective control where another controller overrides the MV
3. Cascade control with saturation in the inner loop.

In all three cases, one may use the anti-windup scheme in Figure 11 with $e_T = \tilde{u} - u$ where u is the desired MV (output of the present controller) and $\tilde{u}$ is the actual MV.

For cascade control (Figures 13 and 14), the question is how we should apply anti windup in the outer loop (C_1) when there is a saturation for MV_2 in the inner (secondary) loop. Saturation in the loop will cause loss of control as seen from the outer loop, and with integral action, $\text{MV}_1 = w_s$ and $\text{CV}_2 = w$ will drift apart. To avoid this we can use anti-windup with back-calculation. In terms of the notation in Eq. (10) and Figure 11, this is achieved if we for C_1 use $u = w_s$, $\tilde{u} = w$ and $e_T = w - w_s$. In addition, one must assume that the inner controller (C_2) has integral action (otherwise, one needs to introduce some other logic which identifies the saturation in the inner loop and stops the windup for MV= w_s in the outer controller). Of course, the inner controller C_2 must also have anti windup, and this is achieved in the "normal" way with $e_T = \tilde{u}–u$.

With cascade control, one may want to avoid that the anti-windup for the outer controller (C_1) corrects for the expected "normal" dynamic control error in the inner loop. This may in particular be a problem if the time scale separation between C_1 and C_2 is small. One solution for C_1, is to replace $e_T = w–w_s$ by $e_T = w–T_2(s)w_s$, where $T_2(s)$ is the expected transfer function for the inner loop. For a linear series cascade system (Figure 14), we have $T_2(s) = G_2C_2(1 + G_2C_2)^{-1}$, where we with a SIMC PID-controller get approximately $T_2(s) = e^{-\theta_2 s}/(\tau_{c2}s + 1)$.

6. Comparison of alternatives for switching

In this section, we further discuss and compare some of the elements for switching and provide some examples.

6.1. MV-MV switching

We have given three alternatives for the MV-MV switching. Which is the best? The answer is of course that this depends on the situation.

6.1.1. Split range control (E5)

This solution has the advantage of being simple to understand, because of the nice visualization with the split range block. One disadvantage is that one must use the same integral and derivative time for all MVs. The controller gains can be adjusted for each input by changing the slopes in the split range block. If one is willing to use more logic elements (programming) then one may use a generalized split range control strategy which allows for independent controller tunings for all inputs. One such example is the baton strategy of Reyes-Lúa & Skogestad (2020a).

Another (and usually more serious) disadvantage is that it may be difficult to combine with CV-CV switching. The reason is that in this case the switching value may be different from the physical max/min-value because it is set by another controller. This may result in delay in switching or it may require fairly complex programming and/or logic.

6.1.2. Multiple controllers with different setpoints (E6)

This is usually the simplest solution to implement as it requires no logic. The switching occurs indirectly by feedback from the output, so there is no need to know the constraint values for the inputs, which is an important advantage. When an input saturates, then one temporarily lose control of the output, and when the output has drifted to reach the next setpoint, the corresponding feedback controller will activate. In addition to being simple to implement, this solution has advantage of allowing for independent tuning of the controllers.

Also, as mentioned earlier, in some cases the setpoint separation may actually be an economic advantage. Smith (2010) (p. 102) mentions the example of pressure control in a storage tank where the two MVs are addition of inert gas (to increase pressure) and vent to air (to reduce pressure). With two controllers with a pressure setpoint difference the consumption with inert gas will be less than with split range control.

The main disadvantage is that the setpoints need to be different, and also that we lose control for some time during switching. We cannot make the setpoint difference too small, because this will result in undesired switching for smaller output variations Therefore, the solution will multiple controllers should not be used for applications where it is necessary to control at the same setpoint all the time, for example, for a critical reactor temperature control (Smith, 2010) (p. 102).

6.1.3. Input (valve) position control (VPC) (E7)

The advantage is that we always control the CV (y) with the same "main" MV (u_1). Thus, this is the preferred solution if tight control of the output y is desired and can only be achieved with u_1, for example, because of a large effective delay for u_2 or because u_2 can only be on/off. The disadvantage is an economic loss because we cannot use the full range for u_1 and also that we need to use both u_1 and u_2 at the same time (e.g., both heating and cooling) in some operating regimes.

6.2. CV-CV switching

For CV-CV switching we have only considered the use of a selector (E4) or some logic element with an equivalent function. We have considered two alternative implementations

1. Selector on the MV (input u) (most general) (Figure 17)
2. Selector on a CV setpoint if we use a cascade implementation (Figure 19)

For both alternatives, the main limitation is that we must assume that each CV (constraint) is paired with a single MV. This is always possible if we have at least as many MVs as we have constraints (CVs), and it may also be possible with more constraints if the constraints are not potentially conflicting, that is, if they require the same kind or selector (max or min).

As an example of where we encounter this limitation, consider a process with two inputs (u_1, u_2) and three inequality constraints (on y_1, y_2, y_3). In addition, each of the two inputs has a desired value $(u_{1,o}, u_{2,o})$ which may be given up if we reach a constraint. We assume that the constraints on y_1 and y_2 are both satisfied by a large u_1 or a large u_2, whereas the constraint on y_3 is satisfied by a small u_1 or a small u_2. Here, we may pair constraint y_1 with u_1 (using a max-selector with $u_{1,0}$ as the other selector input), and pair constraint y_2 with u_2 (using a max-selector with $u_{2,0}$ as the other selector input). However, the constraint on y_3 requires a min-selector (Constraint Rule 1), which is potentially conflicting with the constraint on y_1 and y_2. Note that since we have only two inputs, we can have at most have two active constraints at any given time, so there always exists a feasible solution. The problem is that we cannot guarantee that a feasible solution is realized with the simple selector structure discussed in this paper. To solve the problem one may use a more complex "adaptive" selector structure with additional logic (Bernardino et al., 2022) or one may use MPC.

6.3. Example with combined CV-CV and MV-MV switching: Adaptive cruise control

Adaptive cruise control aims at keeping your car at the desired speed setpoint whenever the surrounding traffic makes it feasible. A simple solution with a CV-CV switch (two controllers with a min-selector) followed by a MV-MV switch (split range control) is shown in Figure 29. Note that this is not a case of "complex MV-CV switching" because the CV-CV switching (selector) comes first.

The following CVs (y_1, y_2) and MVs (u_1, u_2) are involved:

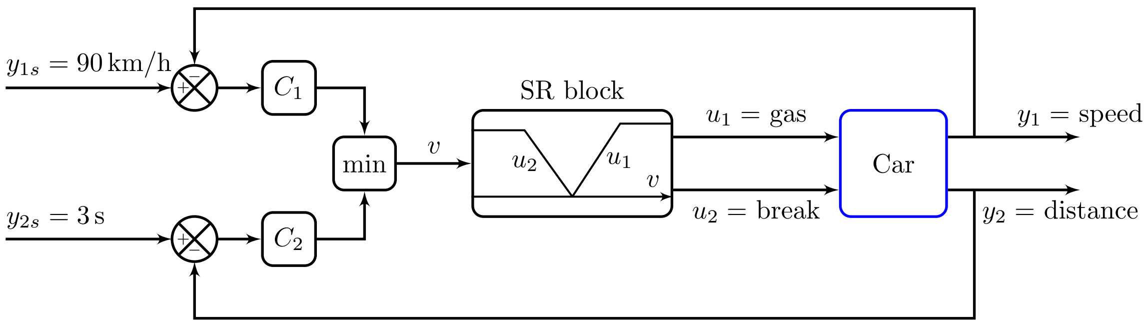


Figure 29: Adaptive cruise control with selector and split range control.

- $y_1 = $ speed (with a typical setpoint $y_{1s} = y_{1,max} =$ speed limit $=$ 90 km/h)
- y_2 = distance to car in front (with a typical setpoint $y_{2s} = y_{2,min} = 3$ seconds)
- u_1 = position of gas pedal (from 0 to 1, where 1 is full gas)
- u_2 = position of brake pedal (from 0 to 1, where 1 is full breaking)

The CV-CV switching uses a selector to switch between controlling the speed y_1 (using C_1) and the distance y_2 (using C_2) and the MV-MV switching uses split range control to switch between using the gas pedal (u_1) and the brakes (u_2). The CV-CV switching uses a min-selector because both the max-speed constraints and the min-distance constraint and satisfied are by a small input v (using little gas) (Selector Rule 1).

For the CV-CV switching, a cascade implementation (Figure 19) is not recommended for this application. First, we cannot have the distance control in the inner loop because it will be inactive when there is no car in front. Second, we should not have the speed control in the inner loop because this will slow down the distance control, which is not acceptable for safety reasons.

For the MV-MV switching there are generally three alternatives, but split-range control is the best in this case. First, it is not clear how to implement the alternative with two controllers. It would require one controller for gas (u_1) and one for breaking (u_2), which would come in addition to the two controllers (for y_1 and y_2) that we already have. Anyway, even if we could find a way to implement two controllers (with two setpoints) for MV-MV switching, it would result in a temporarily loss of distance control during transition between gas and breaking, which is not acceptable for safety reasons. Finally, the VPC alternative, is also not acceptable. For example, if u_1=gas selected to control speed or distance at all times, it requires using both gas (u_1) and breaking (u_2) at the same time for cases where only breaking is needed.

Thus, we should use split range control, but note that this means that we must use the same integral time for both gas and breaking. If this is not acceptable, we need to use a more complex split-range scheme with logic and with four controllers in total.

6.4. MV-CV switching

MV-CV switching is used for cases where it is optimal to "give up" (stop controlling) a CV when a constraint on the MV is encountered.

6.5. Simple MV-CV switching

We first consider the case where we have followed the input saturation pairing rule, which means the CV (y) that should be given is paired with the MV (u) that saturates. Here, the switch is already "built-in" (Rule 3 for selectors), that is, it is not necessary to do anything, except that we must implement anti windup for the controller to ensure that we get good performance when control of y is reactivated, that is, when u is no longer saturated (Reyes-Lúa & Skogestad, 2020b).

There may be two reasons why the CV can be given up when the MV saturates:

- If we are originally at an unconstrained optimal operation point and the CV is a "self-optimizing" variable (with an economically optimal setpoint) then it may be optimal to give up controlling this CV when the MV saturates.

- If we are originally operating at a constraint for the CV, then it may happen that the CV-constraint becomes over-satisfied as the MV saturates, and thus the CV no longer needs to be controlled.

The last situation is common. A simple example is when we want to minimize the driving time between two cities, and thus we want to drive at the speed limit (MV=gas pedal, CV=speed, CV_s = speed limit). If we are going up a steep hill and are driving an old car (or an electric car with a low battery) then the MV may saturate at its maximum ("full gas"), and it will be "optimal" with our bad car (although not desirable) to give up controlling the CV at the speed limit.

It may seem like simple MV-CV switching by "doing nothing" is a trivial and obvious solution, but this is not necessarily true. It requires pairing a MV with a CV that can optimally be given up when saturation occurs, as discussed in the next example.

6.5.1. Example: Anti-surge control

As a less obvious example of simple MV-CV switching (at least to the author), consider anti-surge control of a compressor or pump (Figure 30). For simplicity assume that we have a constant speed compressor, so the compressor itself does not have any control degrees of freedom. However, to avoid too low flow through the compressor, we have implemented a recycle around the compressor with a recycle valve (MV=z).

The objective is to avoid that the flow through the compressor (CV=$y = F$) drops below a minimum value ($F_s = F_{min}$). The recycle valve (MV=z) goes to closed position ($z = 0$) when the throughput (feed flow, F_0) is higher than the minimum flow ($F_0 \geq F_s = F_{min}$), and at this point it is optimal (and also desired) to give up control of CV=F. Let us try to explain in more detail why it works. The minimum flow constraint is satisfied by a large valve opening (MV) so it requires a max-selector (Rule 1 for selectors). This is consistent with the low input limit ($z = 0$) of the input saturation which we know corresponds to a "built-in" max-selector (Rule 3 for selectors). Since both give a max-selector, there is no conflict, and we can "give up" controlling F_{min} just as the valve reaches a closed position ($z = 0$).

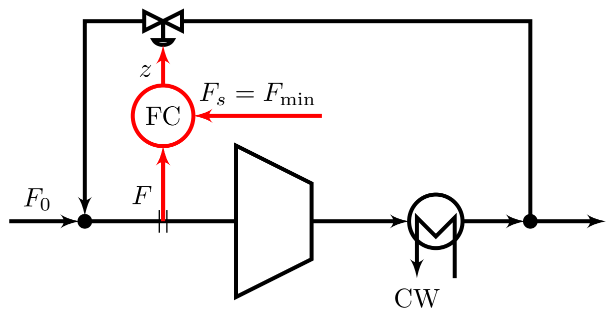


Figure 30: Flowsheet of anti-surge control of compressor or pump (CW = cooling water). This is an example of simple MV-CV switching. When MV=z (valve position) reaches its constraint ($z = 0$) we no longer need to control CV=F at F_{min}.

To further understand how this works, consider a somewhat more complicated case where we also have a maximum constraint on the throughput F_0. For example, it could be that the outflow from the compressor goes to a reactor which cannot handle too high flow because it destroys the catalyst. We then have three constraints

$$\text{MV} = z \geq 0; \quad \text{CV}_1 = F \geq F_{min}; \quad \text{CV}_2 = F_0 \leq F_{0,max}$$

However, we only have one MV, which is the recycle valve position z, so it may seem that there are cases where we cannot satisfy all constraints. However, also the "new" constraint ($F_0 \leq F_{0,max}$) is satisfied by a large value of z, so it also requires a max-selector. Thus, the constraints are never conflicting and the system can be optimally operated using a max-selector as shown in Figure 31.

The MV constraint ($z_{min} = 0$) is included as an input to the max-selector in Figure 31 to show clearly that it is consistent with the other two constraints. However, there is also a "built-in" max-selector in the valve, so it is not really needed and this is why it shown with a parenthesis and dotted line. On the other hand, a potentially fully open valve ($z_{\max} = 1$) is not consistent as it corresponds to a "built-in" min-selector, so if $z = 1$ is reached one needs to give

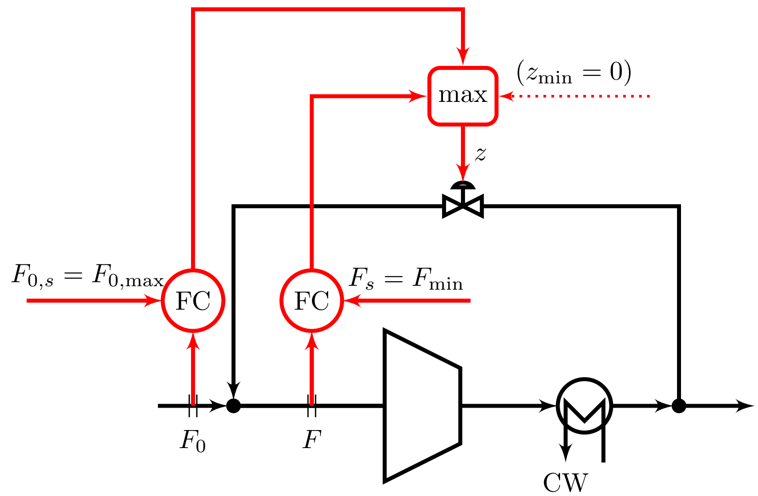


Figure 31: Anti-surge control with two CV constraints. This is an example of simple MV-CV-CV switching.
MV $= z$, CV$_1 = F$, CV$_2 = F_0$ (all potentially active constraints).

up the constraint on F or F_0 (whichever is active at the moment).

6.5.2. Anti-windup and choice of tracking time for simple MV-CV switching (E8)

We need anti-windup in both controllers in Figure 31. If one uses back-calculation as in (21) then $\tilde{u}$ is output from the max-selector and the tracking time τ_T can be used as a degree of freedom to decide when the controller activates. A smaller tracking time means that the tracking of $\tilde{u}$ is better, which means that the controller activates sooner and even before the CV-constraint (F_{min} or $F_{0,max}$) is reached. The disadvantage with a too small tracking time is that it may activate unnecessary.

For example, consider a case when the system is initially operating with a closed recycle valve ($z = 0$), that is, F_0 is between the limits of F_{min} and $F_{0,max}$. We then get a drop in feed flow F_0 (for example, because the inlet pressure p_0 drops) so that F_0 becomes less than F_{min}. Then, with a small tracking time (e.g., $\tau_T = \tau_I/2$ or lees), the P-action in the controller for F will activate (open) the recycle flow sooner, that is, before the flow F through the compressor reaches its constraint (minimum) value F_{min}. This will reduce the undershoot for F and thus reduce the need for back-off from F_{min}, which is a hard constraint because compressor surge can be very damaging. For the other controller (for F_0), we may choose $\tau_T/\tau_I = 1$ or larger if the constraint $F_{0,max}$ is not hard (and thus can be violated dynamically for a shorter time).

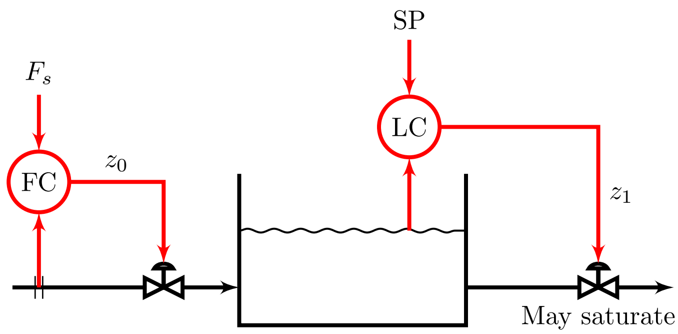
6.6. Complex MV-CV switching = Repairing of loops

Consider next the case where the CV that should be given up is not controlled with the MV that saturates. That is, the MV that saturates (and is causing the need to give up controlling the CV) is used for controlling another CV which cannot be given up. In short, we have *not* followed the input saturation pairing rule, for example, because it did not agree with the "pair-close" rule.

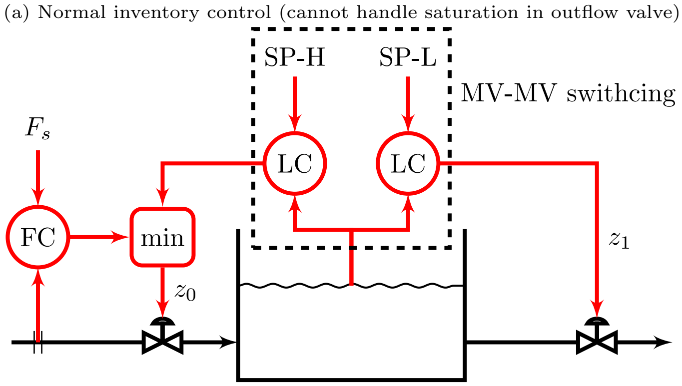
In this case one needs to do an input-output "repairing", which may be realized using MV-MV switching followed by CV-CV switching. First, we use MV-MV switching to keep controlling the CV that cannot be given up, and then we use CV-CV switching (a selector) to give up the other CV. Which of the three MV-MV switching schemes should be used? The answer is that the alternative with multiple controllers is usually the best, because it switches based on feedback from the output (CV) and does not need additional logic for the limits as for split range control (Zotică et al., 2022).

What about the other two alternatives for MV-MV switching? Split range control is not favorable for complex MV-CV switching because the "new" input is already used by another controller so the max/min bounds in the split range block will not be equal to the actual value for when the switch occurs. This will cause a delay in the switching unless some more complicated logic is added. Valve position control (VPC) is not feasible for complex MV-CV switching because it is based on a fixed MV-CV pairing, whereas we need a repairing of loops.

Note that Shinskey (1978) has proposed a separate scheme for complex MV-CV switching, see Figure 9 in Reyes-Lúa et al. (2019), but it is not discussed in this paper.



(a) Normal inventory control (cannot handle saturation in outflow valve).



(b) Bidirectional inventory control (handles saturation in outflow valve by complex MV-CV switching).

Figure 32: Inventory control of single unit for case with desired feed flow F_s (can be given up).

The (total) inventory of liquid or gas in a unit is sometimes self-regulated, but especially for liquids it usually requires feedback control. Liquid inventory is measured by level (sometimes pressure) and gas inventory is measured by pressure. Consider inventory (level) control of a single unit (tank) for the case where the inflow is given. The level then needs to be controlled using the outflow as shown in Figure 32a. However, if the inflow is too large then the outflow valve (MV for level control) may saturate at fully open ($z_1 = 1$). We then lose control of the level, which is not acceptable, so we must instead use the inflow (another MV) for level control. The required repairing of the inventory loop is a case of complex MV-CV switching which can be realized by a combination of MV-

MV switching (using two level controllers with different setpoints) and CV-CV switching (using a min-selector), see Figure 32b. This solution is also known as bidirectional inventory control (Shinskey, 1981).

7. Inventory control of units in series

7.1. Throughput manipulator and radiation rule

As an extension to the previous example, consider inventory control of units in series (Figure 33). Before getting into the details of the control structure, we need to introduce the very important process control concept of *throughput manipulator (TPM)*:

> *TPM = MV used for setting the throughput (production rate) for the process.*

Usually the TPM is a flowrate, but it can in some cases even be a intensive variable, for example, the reactor temperature. Even complex processes usually have only one TPM, because for optimal operation, all feed and utility streams should be in approximate constant ratio to each other. The location of the TPM is a very important decision that determines the structure of the inventory control system. In terms of maximizing production, a good idea, in order to minimize the back-off, is to locate the TPM close to the production bottleneck (Downs & Skogestad, 2011). This is generally inside the process. However, the most common TPM location is at the feed (process inflow) or at the product (process outflow).

Consider first the common case when the feed flow is given. For example, for the simple series process in Figure 33a this means that TPM= F_0. In this case, the inventories need to be controlled using their outflows, that is, inventory control is in the direction of flow. However, if the inflow becomes too large then we may encounter a bottleneck, for example, the outflow of the last unit may saturate at fully open ($z_3 = 1$). This now sets the (maximum) throughput, so in effect we have that the product flow is given, TPM= F_3.

With z_3 saturated at fully open, we lose control of inventory in the last unit, which is not acceptable. To avoid rearranging (repairing) all the inventory loops, the simplest is to start using the inflow F_0 (which can no longer be set freely because of the bottleneck) to control the the last inventory. This results in the control structure in Figure 33b with a “long loop”. This long loop clearly does not follow the “pair close” pairing rule, so control performance for the last inventory is expected to be poor. Thus, this is not a good solution. The best solution, at least in terms of inventory (level) control performance, is to rearrange all the inventory loops to get inventory control opposite direction of flow as shown in Figure 33c.

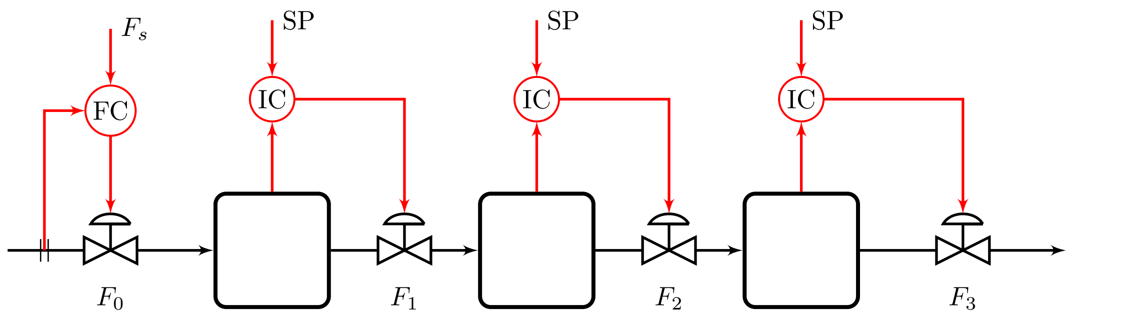
More generally, any internal flow between the units may be specified or be a bottleneck (and thus become the TPM), and to satisfy the ”pair-close” pairing rule for inventory control, we must follow the radiation rule:(Buckley, 1964) (Price et al., 1994) (Aske & Skogestad, 2009):

> *Radiating rule (Figure 32): Inventory control should be “radiating” around a given flow (TPM), that is, it should be in the direction of flow downstream the TPM and it should opposite the direction of flow upstream the TPM.*

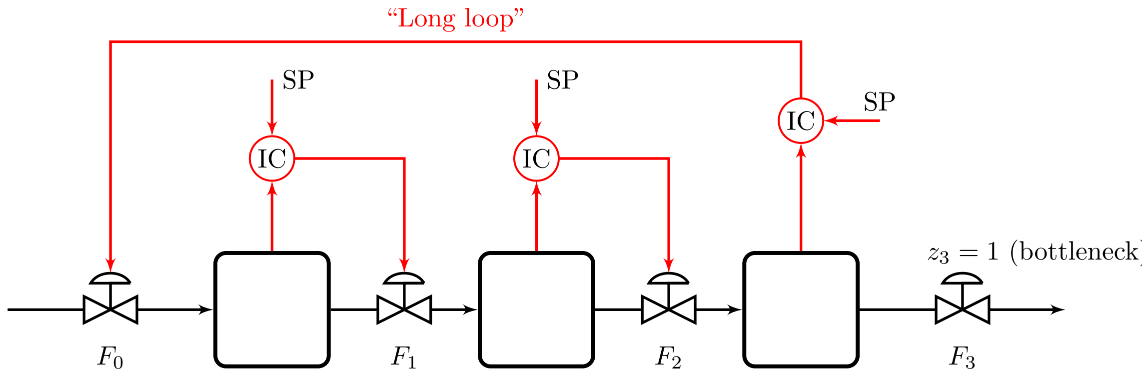
To follow this rule, we need to rearrange the inventory loops if the TPM moves, which seems complicated in terms of logic and coordination. For example, switching from Figure 33a (TPM at feed) to Figure 33c (TPM at product), requires rearranging three loops. Fortunately, it turns out the reuse of the bidirectional inventory control structure discussed in Figure 32b solves the problem in an elegant way. This is the topic of the next section.

7.2. Bidirectional inventory control for units in series

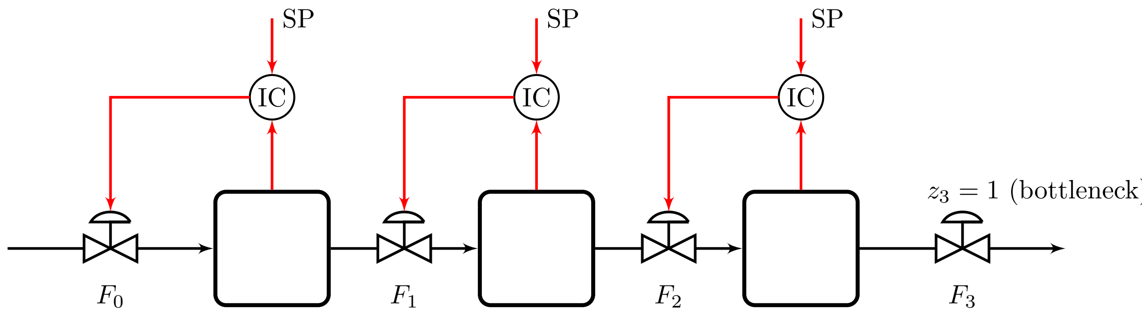
The proposed structure for bidirectional inventory control of units in series is shown in Figure 34 (Shinskey, 1981) (Zotică et al., 2022). Each inventory has two controllers, one with a high setpoint (SP-H) for the inflow and one with a low setpoint (SP-L) for the outflow. This accomplishes the MV-MV switching.



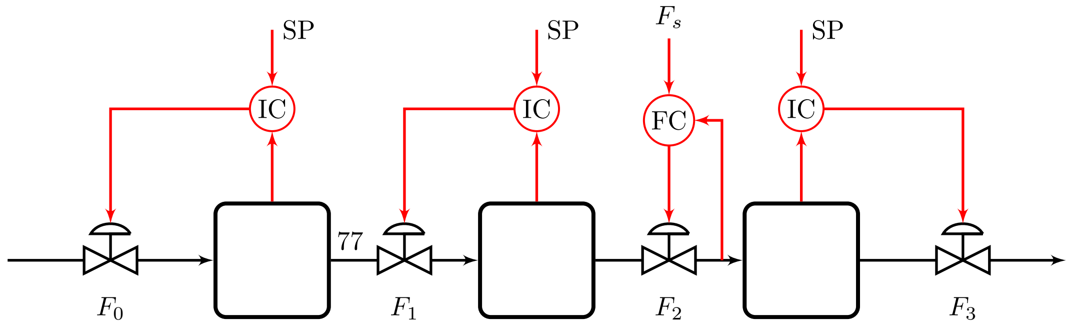
(a) Inventory control in the direction of flow (for given feed flow, TPM = F_0)



(b) Inventory control with undesired "long loop" (for given product flow, TPM= F_3)



(c) Inventory control in the opposite direction of flow (for given product flow, TPM= F_3)



(d) Radiating inventory control around TPM (shown for TPM = F_2)

Figure 33: Inventory control for units in series

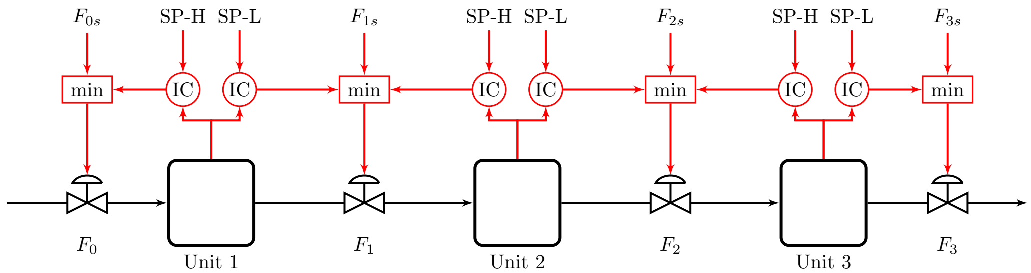


Figure 34: Bidirectional inventory control structure for automatic reconfiguration of loops and production maximization. (Shinskey, 1981) (Zotică et al., 2022).
SP-H and SP-L are high and low inventory setpoints, with typical values 90% and 10%.
Strictly speaking, with setpoints on flows ($F_{i,s}$), the four valves should have slave flow controllers (not shown). However, one may instead have setpoints on valve positions ($F_{i,s}$ replaced by $z_{i,s}$), and then flow controllers are not needed.

For each flow (valve) the decision on what to control (CV-CV switching) is made by a min-selector.

With two controllers for MV-MV switching, we can take advantage of the difference between the high (SP-H) and low (SP-L) inventory setpoints to keep production going in case of temporary bottlenecks, and thus maximize production. Typically, we may set SP-H=90% and SP-L=10%. The inventory controllers should then be fairly tightly tuned. This is to avoid overflowing (100%) or emptying (0%) the units (tanks). We have also introduced a flow setpoint to be able to set the flow (or valve position) at each location, but since it enters a min-selector, the setpoint it is in reality the maximum flowrate. If the flow setpoint is set at a sufficiently low value it becomes the throughput manipulator (TPM) and sets the flow through the whole system. If all flow setpoints are set to infinity, the control system in Figure 34 will automatically make use of the inventories to maximize the throughput, identify the bottleneck, and give a radiating control system around this bottleneck. Yes, it is almost like magic! Zotică et al. (2022) demonstrates this by simulations and find that the solution makes optimal use of available storage for isolating temporary bottlenecks. Shinskey (1981)(p. 46) provides the following enlightening explanation: "Production rate can be set at either end of the process or constrained at any intermediate point without loss of inventory control" (by changing the setpoints F_s). "Should the operator determine that feed rate is too high, he may reduce the setpoint F_{0s} below its measurement .... The subsequent reduction of inflow to tank (unit) 1 ... will cause its level (inventory) to fall. Ultimately, its low-level (SP-L) controller will react by taking control of outflow. This action will cause tank (unit) 2 level to fall, repeating the same scenario. Eventually a new steady state will be reached at the lower production rate and with lower levels in all tanks (units). ... The tank capacities are used for buffering between operations, delaying the transmission of upsets in either direction. Momentary upsets in one operation might not interfere with adjacent operations at all."

7.3. Example: Several layers of selectors for bidirectional inventory control

Figure 35 shows a rather complex control system with a series of min-max-min selectors to avoid a minimum flow constraint on the intermediate flow F_2 (Bernardino & Skogestad, 2023). This may be desirable if unit 3 cannot operate at a low load. To protect against this, we increase the low inventory setpoint in the upstream unit (from L to ML) and decrease the high inventory setpoint in the downstream unit (from H to MH). The setpoint values for ML and MH depend on the nature of future disturbances and whether it is most important to keep production at its maximum or to protect unit 3 against a too low federate. As a starting point one may set, for example, L=10%, ML=40%, MH=60% and H=90%. The logic is further explained at the end of this example, but note that the reason for having two min-selectors is that it is more important to avoid an empty or full tank (unit) (L=10% or H=90%) than to maintain the intermediate inventory (ML=40%, MH=60%).

The control structure in Figure 35 may easily be dismissed as being too complicated so that MPC should be used instead. At first this seems reasonable, but a closer analysis shows that it is not at all clear. First, it seems to be very difficult to make a MPC solution that achieves the objective, which is to maximize throughput for cases with temporary bottlenecks, while at the same time protecting against a minimum flow constraint(Bernardino & Skogestad, 2023). The response of the simpler control structure in Figure ¨34, which is to maximize production, may be realized with MPC by requiring that all inventories must be constrained (between L and H) and using the "trick" of having the four flowrates as CVs with unachievable high setpoints. However, this trick does not seem possible for the more complex case in Figure ¨35. Of course, one may add the minimum flow constraint, but how does MPC know that to protect against reaching this constraint, it is smart to keep some distance to the level constraints (L and H), for example, by using ML rather than L in the upstream unit. For MPC to do the right thing, it seems it would need to know the future disturbances (which is impossible), or a least it must make use of a scenario of expected disturbances, which would make the solution very complicated.

Second, is the control structure on Figure ¨35 really that complicated? It depends on how much time one is willing to put into understanding and explaining it. Traditionally, people in academia have dismissed almost any industrial structure with selectors to be ad hoc and difficult to understand, but this idea needs to be challenged.

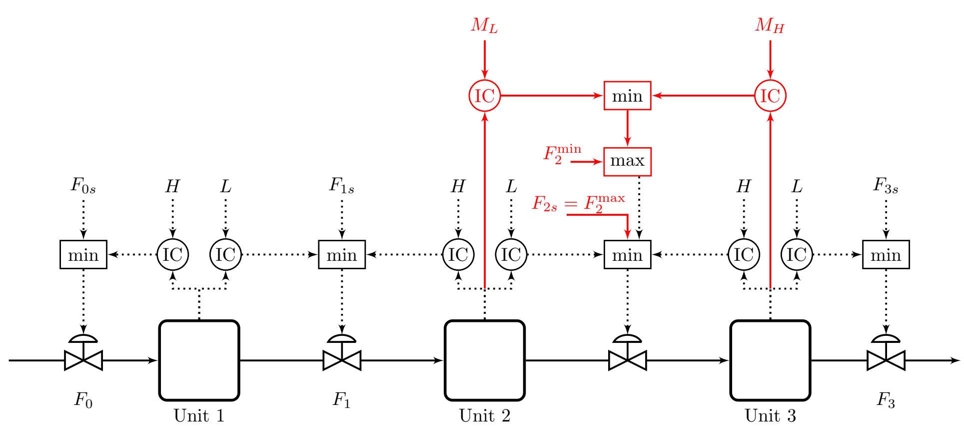


Figure 35: Bidirectional inventory control with minimum flow constraint on F_2.
H, L, M_L and M_H are inventory setpoints.

To this end, we provide an explanation for the red selector logic in Figure 35. As an example (without loss of generality), assume that the throughput initially is set at the feed (F_0) and that none of the constraints on F_2 (F_2^{min} and F_2^{max}) are active. Then we have inventory control in the direction of flow (Figure 33), and for the "red" logic related to F_2, the first (upper) min-selector gives that the inventory (level) in Unit 2 is controlled at the intermediate setpoint ML using F_2. Now, if the feed flow F_0 is reduced so that F_2 drops below F_2^{min}, the "red" max-selector will activate and we lose control of the inventory (level) in Unit 2, and it will keep dropping below M_L until it reaches the low setpoint L. At this point the last "black" min-selector will activate and we start manipulating (decreasing) F_2. This means that at this point we have to give up keeping the feed (F_2) to Unit 3 higher than F_2^{min}. If this is not allowed, then we either need to stop Unit 3 and set $F_2 = 0$ or maybe we can introduce some recycle around Unit 3. However, note that stopping Unit 3, does not necessarily mean that we immediately need to stop the other units (and set all flows to zero), because the inventories in units 1 and 2 will be at L and the the inventory in unit 3 will be at H. So if we can increase F_0 within a reasonably short time (before the inventories in units 1-3 reach their opposite limits), we may be able to recover the lost production in Unit 2.

7.4. Example: On/off control for bidirectional inventory control

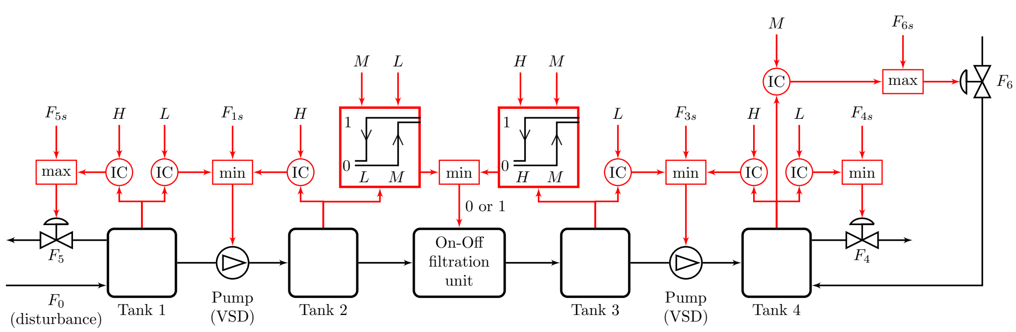


Figure 36: Bidirectional inventory control for industrial case with on/off control of filtration unit.
H, L and M are inventory setpoints with typical values 90%, 10% and 50%.
If it is desirable to set a flowrate (F_s) somewhere in the system, then flow controllers must be added at this location.

Figure 36 shows an industrial process for feed water treatment with a proposed bidirectional inventory control structure. There are six (physical) manipulated variables (three valves, two variable speed pumps and one of/off filtration unit), four inventories that need to be controlled, a desired throughput (F_{4s}) and finally there are maximum and minimum limits on all manipulated variables. Feed F_0 (a disturbance) is a source of cheap "dirty" water and feed F_6 (which can be manipulated) is a source of expensive pure water. If F_0 is too large (larger than the desired production rate F_{4s}), then the excess goes in waste stream F_5, which normally is zero (closed valve).

The cheap feed water F_0 needs to be cleaned in an ultrafiltration unit which must operate in an on/off fashion. This is the reason why the two corresponding inventory controllers are on/off hysteresis controllers which, depending on which of the two is active, let the level in tank 2 vary between M and L, and in tank 3 between H and M.

The desired production rate (throughput) is set by giving the product flow F_{4s}, and a min-selector for F_4 is needed for cases where the feed streams (F_0+F_6) are not large enough, such that the level in tank 4 reaches its low setpoint (L). There are also min-selectors on the three flows between the four tanks in order to get the desired bidirectional inventory control.

It is assumed that the setpoints on F_5 and F_6 are minimum constraints and this gives max-selectors because a large flow satisfies the constraint (Selector Rule 1). In the industrial case, it is desirable that these two flows should be as small as possible ($F_{5s} = 0, F_{6s} = 0$), and then the max-selectors are not needed because the valve has a built-in max-selector. Actually, in the industrial case, F_5 is set by overflow so then the corresponding IC-H-controller (left in the figure) can be omitted.

On the other hand, F_{1s} and F_{3s} are maximum values and are normally set at a large value (infinity) to maximize the flow at these locations, but it is possible to set them at lower values, for example, if temporary reductions in flow are needed. The three intermediate inventory setpoints (M) should be set based on expected disturbances (F_0, F_4, stops etc.), and they may also be adjusted online by the operators based on knowledge about expected future disturbances. It also possible to use a predictive controller (MPC) to adjust these setpoints (M) in a more optimal way. The inventory (level) controllers (IC) are typically

PI-controllers. Also P-controllers may be used, which have the advantage that anti-windup schemes are not needed, but the disadvantage is steady-state offset.

8. Linear feedforward and decoupling

Feedforward, decoupling and linearization may in some cases be indirectly achieved by making use of feedback through cascade control. In particular, it is frequently achieved by adding a fast flow controller. However, more generally model-based approaches are needed, which essentially are based on model inversion.

8.1. Linear feedforward control (E11)

Consider first feedforward control based on a linear process model:

$$y = G_d d + G_u u$$

Assuming a perfect measurement of the disturbance d, we achieve perfect feedforward control ($y = 0$) using $u = -G_u^{-1} G_d d$, so the feedforward controller $u = C_{Fd} d$ in Figure 5 becomes

$$C_{Fd,ideal} = -G_u^{-1} G_d$$

There are two fundamental problems here. The first is that C_{Fd} may not be realizable, for example, if the delay in G_u is larger than in G_d. Second, the model may be wrong, and feedforward control is generally sensitive to model errors. Specifically, if the gain in G_u increases by more than a factor 2, then the resulting input u will be too large, and in fact so large that the output y overshoots more in the opposite direction (in magnitude) than without control ($u = 0$), so feedforward control is worse than no control.

Proof of sensitivity of feedforward to model error. Let the actual process model be $y = G'_d d + G'_u u$. Then the response with ideal feedforward control is $y = G'_d d + G'_u C_{Fd.ideal} d = (G'_d - G'_u G_u^{-1} G_d) d$. With $G'_u = \alpha_u G_u$ and $G'_d = \alpha_d G_d$ (where α_u and α_d are the gain change factors, with nominal values 1), we get $y = (\alpha_d G_d - \alpha_u G_u G_u^{-1} G_d) d = (\alpha_d - \alpha_u) G_d d =$

$(1 - \alpha_u/\alpha_d)G'_d d$, which with $|1 - \alpha_u/\alpha_d| > 1$ is worse in magnitude than with no control ($y = G'_d d$ with $u = 0$) . For example, with $G'_u = 2.5G_u(\alpha_u = 2.5)$ and $G'_d = G_d$ ($\alpha_d = 1$) we get $y = -1.5G_d d$ (with feedforward) which is 50% worse in magnitude than with no control ($y = G_d d$). In another example, with $G'_u = 1.5G_u(\alpha_u = 1.5)$ and $G'_d = 0.5G_d$ ($\alpha_d = 0.5$), we get $1 - \alpha_u/\alpha_d = -2$ or $y = -2G'_d d$ (with feedforward) which is 100% worse in magnitude than with no control ($y = G'_d d$).

To avoid this potential "overshooting" with feedforward control, one may introduce a "chicken factor" f and choose $C_{Fd} = f \cdot C_{Fd,ideal}$, where typically $f = 0.8$. Nevertheless, feedforward control may be very helpful in many cases, but it may be even better to use nonlinear feedforward control (see Section 9) to avoid changes in the linear model caused by nonlinearity.

8.2. Linear decoupling (E12)

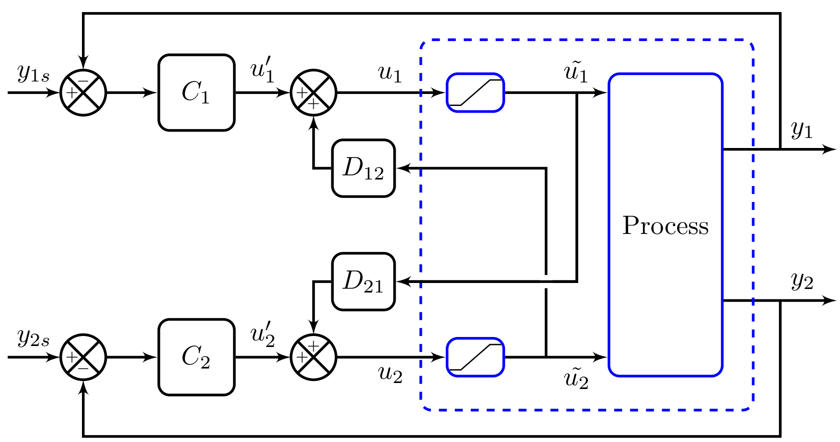


Figure 37: Linear decoupling with feedback (reverse) implementation of Shinskey (1979)

The feedforward idea can also be applied to decoupling as illustrated in Figure 37. For the 2x2 case, let the process model be $y = Gu$, where

$$G = \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix}$$

We then have $y_1 = G_{11}u_1 + G_{12}u_2$ and considering u_2 as a measured disturbance and setting $y_1 = 0$ we get $u_1 = -G_{11}^{-1}G_{12}u_2$. We can do the same for y_2. Thus, for ideal decoupling, the two decoupling elements in Figure 37 become

$$D_{12} = -\frac{G_{12}}{G_{11}}, \quad D_{21} = -\frac{G_{21}}{G_{22}}$$

To make the decoupling elements realizable, we need a larger (effective) delay in the off-diagonal elements than in the diagonal elements. This means that the "pair close" rule should be followed also when using decoupling. An alternative is to use static decoupling or partial (one-way) decoupling.

Note that Figure 37 uses the feedback decoupling scheme of Shinskey (1979). It is also referred to as inverted decoupling (Wade, 1997). Compared to the to the more common "feedforward" scheme (where the input to the decoupling elements is u' rather than $\tilde{u}$), the feedback decoupling scheme in Figure 37 has the following nice features (Shinskey, 1979):

1. With inverted decoupling, the model from the controller outputs (u') to the process outputs (y) becomes (assuming no model error) $y_1 = G_{11}u'_1$ and $y_2 = G_{22}u'_2$. Thus, the system, as seen from the controllers C_1 and C_2, is in addition to being decoupled (as expected), also identical to the original process (without decoupling). This simplifies both controller design and switching between manual and auto mode.

2. The inverted decoupling works also for cases with input saturation, because the actual inputs ($\tilde{u}$) are used as inputs to the decoupling elements.

Note that there is potential problem with internal instability with the inverted implementation because of the positive feedback loop $D_{12}D_{21}$ around the two decoupling elements. However, this will not be a problem if we can follow the "pair close" pairing rule. In terms of the relative gain array (RGA), we should avoid pairing on negative RGA-elements.

To avoid this potential problem (and also for other reasons, for example, to avoid sensitivity to model uncertainty for strongly coupled processes) one may use one-way decoupling where one of the decoupling elements is zero. For example, if tight control of y_2 is not so important, then one may select $D_{21} = 0$.

The scheme in Figure 37 can easily be extended to 3x3 systems and higher. Also here one may simplify by using using static decoupling or partial decoupling, that is, using decoupling only for the important outputs. However, for many multivariable control problems, model predictive control is the preferred technique.

Finally, it should be noted that in many cases, feedforward and decoupling can be achieved in a simpler way using ratio control. This is a then special case of nonlinear feedforward and decoupling as discussed next.

9. Nonlinear feedforward, decoupling and linearization (E13)

9.1. Example: Mixing process

As an introductory example, consider the mixing of component A (with flow $u_1 = F_1$ [kg/s]) and component B ($u_2 = F_2$ [kg/s]) to make a product with composition $y_1 = x$ (fraction of A) and total flow $y_2 = F$ [m3/s]; see Figure 38. For example, A could be methanol and B could be water, that is, we have $x_1 = 1$ and $x_2 = 0$. An equivalent process from a control point of view, would be a shower process where we mix hot and cold water.

This is a coupled process and it my be difficult to decide on good pairings between the manipulated variables u and controlled variables y for single-loop control. However, based on physical insight (or a steady state model), the system becomes decoupled if we use as "transformed" manipulated variables (McAvoy, 1983) (p. 136),

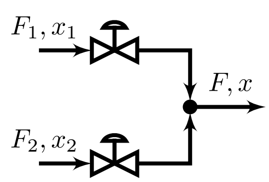


Figure 38: Flowsheet of in-line blending system (mixer) where F is the flowrate [kg/s] and x is the mass fraction of component A [kg A/kg]

$$v_1 = \frac{F_1}{F_1 + F_2} \tag{24a}$$

$$v_2 = F_1 + F_2 \tag{24b}$$

The resulting model becomes

$$y_1 = v_1$$

$$y_2 = v_2$$

Seborg et al. (2016) (p. 343) write about the choice of transformed manipulated variables in (24): "This means that the controlled variables are identical to the manipulated variables! Thus, the gain matrix is the identity matrix, and the two control loops do not interact at all. This situation is fortuitous, and also unusual, because it is seldom possible to choose manipulated variables that are, in fact, the controlled variables".

As shown next, the statement that this is "fortuitous, and also unusual" is incorrect. If we assume that the disturbances are measured, then it is always possible to introduce ideal transformed manipulated variables v_0 which are equal to the controlled variables y, simply by choosing v_0 as the right-hand side of the steady-state model equations (Skogestad et al., 2023).

9.2. Ideal transformed inputs

Consider a steady-state model

$$y = f_0(u, d) \tag{25}$$

and select the ideal transformed input v_0 (controller output) as the right-hand-side,

$$v_0 = f_0(u, d) \tag{26}$$

For implementation, one needs to invert the model by solving (26) with respect to u for given values of v_0 and d. We can formally write the solution as

$$u = f_0^{-1}(v_0, d) \tag{27}$$

At steady state, the resulting transformed system simply becomes

$$y = v_0 \tag{28}$$

That is, we have $y = Iv_0$, so we have perfect feedforward control, decoupling and linearization at steady state. It looks like magic, but it works in practice. To have perfect control, we must assume that all disturbances d are measured, but if this is not the case then one may use a simpler variant of f_0 as the transformed input v, to get partial feedforward or decoupling. To correct for unmeasured disturbances and model error, the setpoint for v_0 is adjusted by an outer controller C (usually a decentralized PID controller). The final control structure is then as shown in Figure 39. Here we have allowed for treating some measured states w as disturbances because this my allow for simpler models (Skogestad et al., 2023).

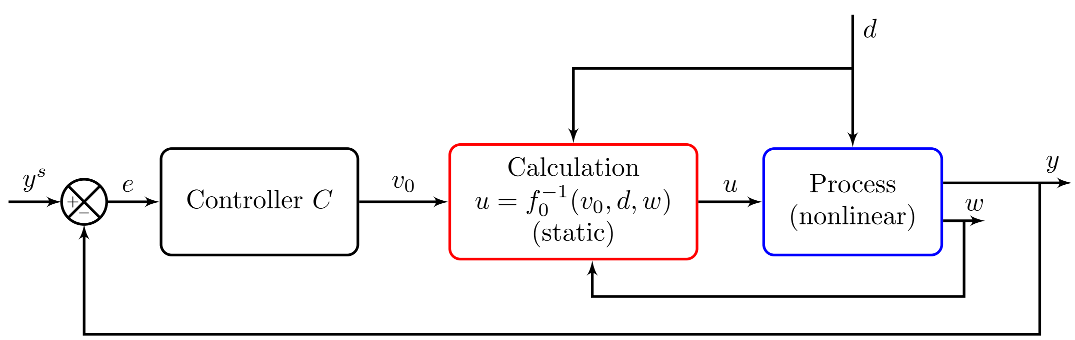


Figure 39: Feedforward, decoupling and linearization (red calculation block) using transformed inputs $v_0 = f_0(u, d, w)$ based on static model $y = f_0(u, d, w)$. In the ideal case with no model error, the transformed system from v_0 to y (as seen from the controller C) becomes $y = Iv_0$ at steady state.
d = measured disturbance
w = measured state variable

The method in (27) and Figure 39 was published only recently (Skogestad et al., 2023), but it is not new. Industry frequently makes use of nonlinear static model-based "calculation blocks", "function blocks", or "ratio elements" to provide feedforward action, decoupling or linearization (adaptive gain), and Shinskey (1981) and Wade (2004) provide examples. In particular, Wade (2004) (pages 217, 225 and 288) presents similar ideas. However, the generality of the method is new.

The method is based on a static model, so it may be necessary to "fine tune" the implementation by adding dynamic compensation (typically lead-lag with delay) on the measured variables (d or w) to improve the dynamic response. Alternatively, there is also a dynamic variant of the method based on using a first-order model, which turns out to be a special case of the nonlinear control method called "feedback linearization" Skogestad et al. (2023).

9.3. Example: Ideal transformed inputs for mixing process

This example is a generalization of the previous mixing example, where we do not assume that the two feeds are pure components. Let x_1 and x_2 represents the mass fraction of A in the two feed streams (in the previous example we had $x_1 = 1$ and $x_2 = 0$). We want to mix feed 1 (with flowrate $u_1 = F_1$ and fraction $d_1 = x_1$) and feed 2 ($u_2 = F_2, d_2 = x_2$) to make a product with fraction $y_1 = x$ [kg/kg] and total flow $y_2 = F$ [kg/s]. The steady-state model (component mass balance for A and total mass balance) gives

$$x = \underbrace{\frac{F_1 x_1 + F_2 x_2}{F_1 + F_2}}_{v_{0,1}} \tag{29a}$$

$$F = \underbrace{F_1 + F_2}_{v_{0,2}} \tag{29b}$$

Note that the two ideal transformed inputs, $v_{0,1}$ and $v_{0,2}$, are simply the right-hand side f_0 of the model equations. Also note that with $x_1 = 1$ and $x_2 = 0$, they are identical to v_1 and v_2 in the previous example. To implement the transformed inputs, we may invert the model equations (29) to get

$$F_1 = \frac{v_{0,2}(v_{0,1} - x_2)}{x_1 - x_2} \tag{30a}$$

$$F_2 = \frac{v_{0,2}(x_1 - v_{0,1})}{x_1 - x_2} \tag{30b}$$

(30) can be implemented as a nonlinear calculation block using Figure 39. However, inspired by the linear feedback decoupling scheme in Figure 37, an alternative implementation is shown in Figure 40. To derive this scheme, we solve (29a) with respect to F_1 and we solve (29b) with respect to F_2, to get

$$F_1 = F_2 \; \frac{v_{0,1} - x_1}{x_2 - v_{0,1}} \tag{31a}$$

$$F_2 = v_{0,2} - F_1 \tag{31b}$$

These equations are coupled, but may be solved by feedback as shown in Figure 40. The resulting transformed system from v_0 to y is $y = Iv_0$, so we have perfect feedforward control, decoupling and linearization. The role of the two outer PID-controllers C_1 and C_2 in Figure 40 is to correct for model uncertainty and unmeasured disturbances.

Besides being simple to understand and implement, the advantage with the implementation in (31) and Figure 40, compared to an inversion using (30), is that it provides partial decoupling and disturbance rejection also when F_1 or F_2 saturate. That is, when F_2 saturates, we will maintain control of $y_1 = x$ but lose control of $y_2 = F$. Similarly, when F_1 saturates, we will maintain control of y_2 but lose control of y_1. However, if $y_1 = x$ (composition, or temperature for a shower) is the most important to control then we may want to give up $y_2 = F$ (flow) also in the latter case. This may be achieved by making the anti-windup from both inputs ($u_1 = F_1$ and $u_2 = F_2$) go to controller C_2 which controls $y_2 = F$.

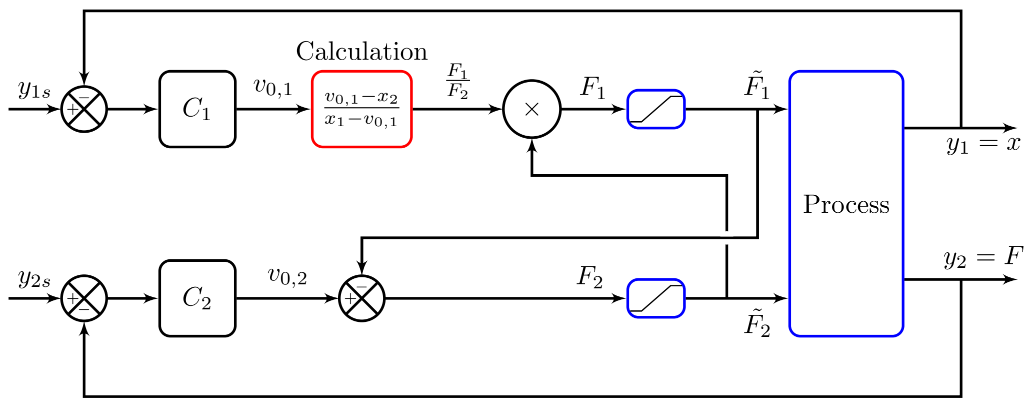


Figure 40: Feedback implementation (31) of ideal feedforward, linearization and decoupling for the mixing process (blending system) in Figure 38.
The output from feedback controller C_1 is the ideal transformed input $v_{0,1}$. From this and measured disturbances (inlet compositions x_1 and x_2), the feedforward calculation element (red) computes F_1/F_2. The decoupling uses the actual measured flowrates $(\tilde{F}_1, \tilde{F}_2)$ and is given by one multiplication element and one subtraction element. The resulting transformed system as seen from the feedback controllers (C_1, C_2) is simply $y_1 = v_{0,1}$ and $y_2 = v_{0,2}$ (with no model error).
Note that there are two inner flow controllers (for F_1 and F_2) which are not shown in the figure.

10. Discussion

10.1. Understanding and improving advanced industrial control solutions

Academics tend to dismiss industrial advanced control solutions as ad-hoc and difficult to understand. However, with the knowledge of control elements presented in this paper, it should be possible to understand most industrial solutions and also to propose alternatives and improvements.

If the industrial solution has a selector (sometimes realized using a limiter, especially for the cascade implementation) then generally there is a CV constraint involved. Most likely, the selector is performing a steady-state CV-CV switch, but there may be exception; see the cross-limiting example below.

A CV-CV switch can be realized in two ways, either with two (or more) independent controllers with a selector on the MV, or as a cascade implementation with a selector on the CV setpoint. If there are several CVs (max and min) is series then we know that the constraints are potentially conflicting and that the highest priority constraint is at the end.

If the industrial solution has a valve position controller (VPC) then there may be two quite different problems that it is trying to address, and it may not be immediately clear which. If we have an extra MV for dynamic reasons (Figure 21) then the two controllers (and MVs) are used all the time. The MV used by the VPC is then used on the long time scale, whereas the MV controlling the CV is used for dynamic reasons (fast control). Here, an alternative is to use parallel control (Figure22).

On the other hand, if we have an extra MV for steady-state reasons (Figure 28) then we have a case of MV-MV switching where the VPC is only active part of the time (when the "primary" MV (u_1) saturates).

For MV-MV switching there are two alternatives to VPC, namely split range control (Figure 25) or multiple controllers with different setpoints (Figure 27). Split range control (Figure 25) is usually easy to identify. Multiple controllers for the same output (with different setpoints) (Figure 27) may be a bit more difficult to identify.

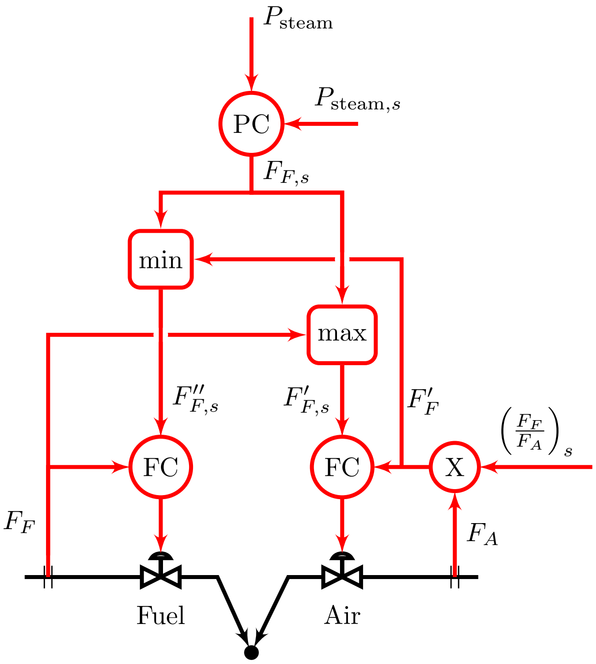


Figure 41: Cross-limiting control for combustion where air (A) should always be in excess to fuel (F).

The objective of cross-limiting control is to mix air and fuel in a given ratio for combustion, but during dynamic transients when there may be deviations from the given ratio, one should make sure that there is always excess of air. The scheme in Figure 41 with a crossing min- and max- selector is widely used in industry and is mentioned in many industrial books (e.g., Liptak (1973), Nagy (1992) and Wade (2004)). The setpoint for the ratio, $(F_F/F_A)_s$, could be set by a feedback controller (not shown) which controls, for example, the remaining oxygen after the combustion.

The selectors in Figure 41 are used to handle the dynamic (transient) case, so this is a somewhat rare case where the selectors are not performing a steady state CV-CV switch.

How does it work? When the main fuel controller (which in the figure controls steam pressure (PC), but it could be temperature, power etc.) wants to change the load (firing), it does this by increasing both fuel and air in a desired ratio, $(F_F/F_A)_s$. This could be accomplished with the control structure in Figure 41 without the two selectors. The only thing which would then be a bit strange is that the air flow controller seems to be controlling the fuel flow (F_F'), but note that this is an inner controller for the ratio control, so it gives the right result.

Now let us look at how it works with the two selectors included. When the fuel controller (PC) demands higher flows, the air flow will increase first, while the min-selector holds back the fuel increase. On the other hand, when the controller (PC) demands lower flows, the fuel flow increases first while the max-selector holds back the air flow (so it remains high for a longer time). In summary, we are guaranteed to always have excess of air during dynamic transients.

Is it possible to derive or understand this scheme based on what is presented in this paper? No, this seems to be a unique "invention". This invention can be applied more generally to chemical reactors where one should always have excess of one of the reactants.

There exists probably many more such inventions which are not discussed in this paper, for example, they may be found in the books by Shinskey. Also Liptak (1999) shows many control structures for various applications, which may contain other inventions. It would be nice to get an overview of special control structures (inventions) that solve specific control problems. However, efforts must be made to minimize the number of special structures and clearly explain what problem they are solving.

When one sees a complex structure like in Figure 41, then it is reasonable to think that MPC may provide a simpler solution. This may be possible in some cases, but it is not clear that MPC can solve the cross-limiting problem in a good way. This is left as a challenge to the MPC community.

10.3. Smith Predictor

Note that the Smith Predictor (Smith, 1957) is not included in the list of control elements given in the Introduction, although it is a standard element in most industrial control systems to improve the performance for processes with time delay. The reason why it is not included, is that PID control is usually a better solution, even for processes with a large time delay (Ingimundarson & Hägglund, 2002) (Grimholt & Skogestad, 2018b). There has been a myth that PID control works poorly for processes with delay, but this is not true (Grimholt & Skogestad, 2018b). The origin for the myth is probably that the Ziegler-Nichols PID tuning rules happen to work poorly for static processes with delay.

The Smith Predictor is based on using the process model in a predictive fashion, similar to how the model is used in internal model control (IMC) and model predictive control (MPC). With no model uncertainty this works well. However, if tuned a bit aggressively to get good nominal performance, the Smith Predictor (and thus also IMC and MPC) can be extremely sensitive to changes in the time delay, and even a *smaller* time delay can cause instability. When this sensitivity is taken into account, a PID controller is a better choice for first-order plus delay processes Grimholt & Skogestad (2018b).

Also note that the potential extreme sensitivity to time delay error of the Smith Predictor may not appear when considering other common robustness measures, like the gain margin (GM), phase margin (PM) or sensitivity peak (M_s-value). However, it affects the delay margin (DM [s]) which is the smallest change in the time delay that will cause the closed-loop to become unstable. In general, we have

$$\mathrm{DM} = \frac{\mathrm{PM}}{\omega_c} \tag{32}$$

where ω_c [rad/s] is the crossover frequency (where the loop gain $|L(j\omega)|$ crosses 1 from above) and PM [rad] is the phase margin at this frequency. As opposed to a PID controller, the Smith Predictor may have multiple crossover frequencies, resulting in very large values for ω_c and thus in a very small delay margin.

10.4. Theoretical basis for selectors

Consider a static constrained optimization problem,

$$\min_{u} J(u.d), \quad \text{subject to } g(u,d) \leq 0 \tag{33}$$

By introducing the dual variables λ (also know as Lagrange multipliers or shadow prices) it can be reformulated as an equivalent unconstrained optimization problem

$$\min_{u,\lambda} \underbrace{(J(u,d) + \lambda g(u,d))}_{\mathcal{L}(u,d,\lambda)} \tag{34}$$

with the following necessary optimality (KKT) conditions

$$\nabla_u \mathcal{L} = 0, \quad \lambda \geq 0, \quad g \cdot \lambda = 0 \tag{35}$$

Here, $\nabla_u \mathcal{L}$ is the gradient of the Lagrange function $\mathcal{L}$ with respect to the degrees of freedom (primal variables; inputs) u. The requirements $\lambda \geq 0$ and $g \cdot \lambda = 0$ are needed because the constraint g is an inequality rather than equality constraint. Note here that the lower limit $\lambda = 0$ corresponds to unconstrained operation. Using dual decomposition, the KKT optimality conditions may be solved by feedback control as shown in Figure 42 (Dirza et al., 2021) (Krishnamoorthy & Skogestad, 2022). The outer slow "constraint controller" is typically a decentralized PI-controller which controls the constraint (CV=g with CV$_s = 0$) by manipulating the dual variable (MV=λ). This value is send to a max-selector, $\max(\lambda, 0)$, which is then used for solving the following unconstrained optimization problem with respect to the primal variables u:

$$\nabla_u \mathcal{L} = \nabla_u J + \lambda \nabla_u g = 0$$

In Figure 42 this problem is solved by feedback using a "gradient controller" but it could alternatively be solved numerically and using a calculation block. Importantly, the max-selector in Figure 42 provides the optimal transition be-

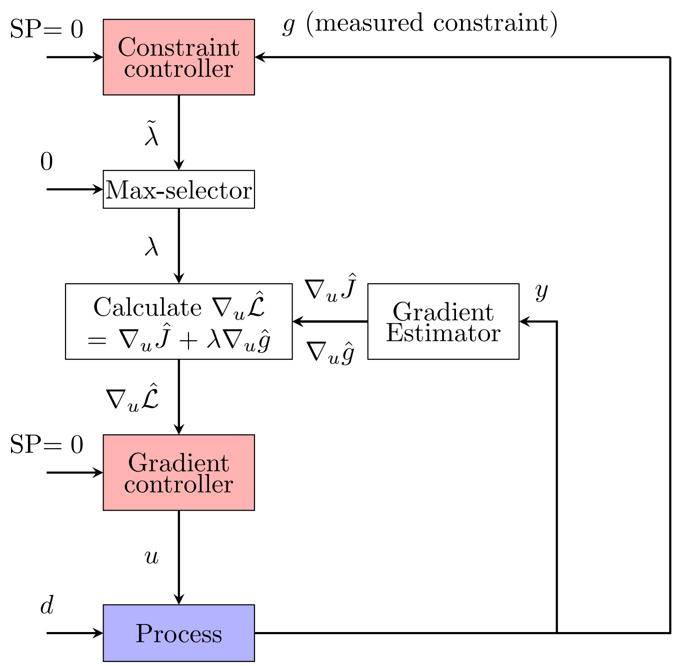


Figure 42: Dual decomposition of constrained optimization with upper (slow) constraint controller and max-selector on the dual variable λ (Lagrange multiplier).

tween optimal constrained and unconstrained steady-state operation (and the reverse), in the same way as the selectors elements used in this paper.

In summary, a selector is needed somewhere it the control structure in order handle steady-state constraint switching in an optimal manner. This justifies the use of selectors for optimal steady-state CV-CV switching.

10.5. Model-based optimizing control

10.5.1. Economic model predictive control (EMPC)

Economic model predictive control combines the two objectives of optimization and control into one mathematical optimization problem. There is no separation into layers and thus no controlled variables or setpoints. At any given sample time, the optimal input u is found as the solution to an open-loop dynamic optimization problem with given initial values of the states, x_0, and given expected future disturbances d_k. Here k denotes the sample time. In discrete form, the objective is to minimize the aggregated cost J from the present time ($k = 0$) and into the future ($k = N$):

$$\min_{u_k} J, \quad \text{where } J = \sum_{k=0}^{k=N} J_k$$

(often $N = \infty$). The cost J is minimized subject to given model equations, e.g. $dx/dt = f(x, u, d)$ (appropriately discretized), and operational constraints, $g_k \leq 0$. This is an open-loop online optimization problem which gives a sequence of optimal inputs u_k into the future, but importantly only the first value u_0 is actually implemented. Feedback is introduced by resolving the optimization problem at every sample with an updated value for the initial state x_0. In EMPC, the cost J includes a purely economic term $J_\$$ [\$ or \$/s] as well as a "regularization" term J_c related dynamic control performance, $J = J_\$ + J_c$. However, EMPC is rarely used in practice, both because it is often complex and difficult to tune, and because there is often a time scale separation between the tasks of optimization and control which makes it possible to separate the tasks of minimizing $J_\$$ and J_c with little economic loss.

10.5.2. *Conventional MPC (with setpoints)*

Conventional MPC is setpoint-based, so it may be combined with an upper real-time optimization layer (RTO, usually static) which computes the optimal setpoints y_s. Typically, MPC implements these setpoints in an "optimal" way by minimizing at each sample time the following quadratic cost function

$$J_c = \sum_{k=0}^{k=N} (y_k - y_{s,k})^T Q (y_k - y_{s,k}) + \Delta u_k^T R \Delta u_k \tag{36}$$

Here, Δu_k represents the input change between samples, and Q and R are weight matrices. By increasing Q relative to R the control engineer can put more emphasis on setpoint tracking, which generally results in more aggressive control (larger changes in u and less robustness). Note that MPC is formulated as an open-loop optimization problem. However, for linear unconstrained systems with a quadratic cost J_c, it happens that the solution to this open-loop linear quadratic (LQ) problem can be realized as a simple closed-loop control law, $u(t) = Kx(t)$ (in continuous time) (e.g., Skogestad & Postlethwaite (1996)). That is, it is optimal to use proportional control from the present value of the states. The matrix K may be precomputed for a given problem (with given weights).

This can be generalized to linear systems with constraints by using a different precomputed K-matrix in each region of the expected future dynamic constraints. This solution is known as explicit MPC (Bemporad et al., 2002). However, in practice the number of regions tends to get very large, and the original repeated open-loop solution based on (36) is usually preferred. Nevertheless, the fact the open-loop solution is equivalent to a feedback solution, $u = Kx$, at least locally (in a linear region), indicates that it inherits some of the robustness benefits of feedback control, provided that the MPC problem is solved as a repeated online optimization problem.

10.5.3. Shortcomings of MPC

Model predictive control has been commercially available since about 1980 and it became very popular in the refining and petrochemical industry at the end of the 1980s. At this time, a bright future was expected for MPC in all process industries and many expected that it would replace most of the "outdated" industrial advanced control solutions, which were viewed as ad-hoc and difficult to understand and design. It was even proposed that MPC would replace the PID controller as the standard controller for basic control tasks (e.g., Pannocchia et al. (2005)). However, the relatively slow penetration of MPC into other process industries over the last 30 years, shows that MPC also has serious shortcomings in terms of its practical use.

First, even with a detailed model, MPC may not be the best solution for a given control problem. In particular, as shown next, optimal control (LQG) and MPC can handle only indirectly and with much effort the three main inventions of process control; namely integral action, ratio control and cascade control. This in itself explains why MPC will never take over as the only tool in the control engineers toolbox.

10.5.4. Integral action and MPC

Consider again the simple setpoint tracking problem in section 2. Figure 7 compares the responses with feedforward and feedback control. The responses are identical nominally, but the feedback solution is a lot more robust to gain uncertainty. Which solution would we get with MPC? With some measurement error (which must be included in the estimator problem), MPC will give the feedforward solution, because with no model error this is optimal. To make MPC include feedback and in particular integral action (which is needed to handle model uncertainty), the solution in the original industrial MPC implementations (e.g., DMC Cutler & Ramaker (1980)) was to let difference between the measured and predicted output be added as a bias. This is the same as assuming that the deviation is caused by a step disturbance acting on the output. However, this approach does not work well for processes with slow dynamics, because of disturbances acting on the input which appear as ramp-like disturbances at the output (e.g., Lundström et al. (1995). An observer-based implementation may avoid this limitation, and to get integral action the standard "trick" is to add in the estimator (observer) one integrating disturbance ("process noise") for each output y (e.g., Rawlings (2000)). The larger this integrating disturbance is made (by changing a corresponding weight), the more feedback MPC will use. This illustrates both the weakness and the strength of MPC. The weakness is that the engineer cannot specify directly the desired solution, in this case to use feedback (PI control) only. The strength of MPC is that, in a more complex case, for example with a long measurement delay for y, it is possible to coordinate the use of feedback and feedforward control in a good way, by changing a single tuning parameter, namely the weight (magnitude) of the integrating disturbance.

10.5.5. Cascade control and MPC

MPC is not the right tool when cascade control (Figure 13) is the preferred solution. The problem with MPC is that it cannot make use of an extra process measurements (w) unless it has a model of how the output y and w are related. In addition, even with a model, it is not clear that MPC can be tuned to put proper emphasis on using the measurement w rather than using the uncertain model. On the other hand, with conventional cascade control (Figure 13) an engineer can easily make use of an extra measurement w, just using the physical insight that fast control of w, will indirectly benefit the control of y, and the tuning of the two controllers is easily done online in a sequential manner, where first the fast inner controller is tuned.

10.5.6. Ratio control and MPC

Ratio control is difficult to implement with MPC. We need a nonlinear model for how y depends on u and d, which may be a quite complex model, for example, if y is viscosity. On the other hand, a simple ratio control implementation (e.g., Figure 15) does not require a model for how y depends on u and d, we just need the physical insight that y will be constant if we keep the ratio u/d constant (see Section 5.2.3).

10.5.7. Summary of MPC shortcomings

Some shortcomings of MPC are listed below, in the expected order of importance as seen from the user's point of view:

1. MPC requires a "full" dynamic model involving all variables to be used by the controller. Obtaining and maintaining such a model is costly.

2. MPC can handle only indirectly and with significant effort the three main inventions of process control; namely integral control, ratio control and cascade control.

3. Since a dynamic model is usually not available at the startup of a new process plant, we need a simpler control system, based on advanced regulatory control elements, for the initial time period. MPC will then only be considered if the performance of this initial control system is not satisfactory.

4. It is often difficult to tune MPC (e.g., by choosing weights or sometimes adjusting the model) to give the engineer the desired response. In particular, since the control of all variables is optimized simultaneously, it may be difficult to obtain a solution that combines fast and slow control in the desired way.

5. The solution of the online optimization problem is complex and time-consuming for large problems.

6. Robustness to model uncertainty is handled in an ad hoc manner, for example, through the use of the input weight R. On the other hand, with the SIMC PID rules, there is a direct relationship between the tuning parameter τ_c and robustness margins, such as the gain, phase and delay margin (Grimholt & Skogestad, 2012).

7. With MPC, the approach of using a separate estimator for the states is not optimal because the separation principle only holds for linear systems without uncertainty (see Section 10.5.9).

Shortcomings 2 - 5 are related and become more serious for larger problems. Thus, even with MPC, the problem is often decomposed by using separate MPCs for each process unit. There have been many academic efforts over the last 30 years to deal with the last two shortcoming, and significant progress has been made. However, this makes the problem even more difficult to formulate and solve.

10.5.8. Summary of MPC advantages

The above limitations of MPC, for example, with respect to integral action, cascade control and ratio control, do not imply that MPC will not be an effective solution in many cases. On the contrary, MPC should definitely be in the toolbox of the control engineer. First, standard ratio and cascade control elements can be put into the fast regulatory layer and the setpoints to these elements become the MVs for MPC. More importantly, MPC is usually better (both in terms of performance and simplicity) than advanced regulatory control (ARC) for:

1. Multivariable processes with (strong) dynamic interactions.
2. Cases where we want to coordinate feedforward and feedback control in a good way.
3. Cases where we want to dynamically coordinate the use of many inputs (MVs) to control one CV.
4. Cases where future information is available, for example, about future disturbances, setpoint changes, constraints or prices.
5. Nonlinear dynamic processes (nonlinear MPC).

The handling of constraints is often claimed to be a special advantage of MPC, but it can it most cases also be handled well by ARC (using selectors, split-range control solutions, anti-windup, etc.).

It is often argued that MPC is more complex than ARC, but this may not be true. On the contrary, ARC solutions can get complex in some cases, for example, with may layers of cascades and selectors. Thus, even if ARC may give acceptable control performance, MPC may be simpler and therefore the preferred solution for some problems.

10.5.9. The fundamental problem with MPC: The separation principle does not hold

With MPC, we find at each sample time the optimal input solving an open-loop (feedforward) control problem (see Section 10.5.2). Feedback is only introduced indirectly by updating the initial states x_0. In particular, for nonlinear MPC, it is frequently assumed that all the states x are perfectly measured, but this is not realistic, especially not in process control applications.

If all states are not measured, the standard approach is to obtain the "optimal" estimate of the initial states $\hat{x}_0$ from the available measurements y by solving a separate estimation problem (usually another quadratic optimization problem). In the linear case, this optimal estimate is the Kalman filter, and the combined solution resulting from using at every sample $u_0 = K\hat{x}_0$ is known as the Linear Quadratic Gaussian (LQG) control. However, this assumes that the "separation principle" applies, which means that the control and estimation problems can be separated. Unfortunately, the separation principle only holds for a limited class of problems, specifically for the linear case without model uncertainty. This was demonstrated by a famous counterexample (Doyle, 1978) which showed that in extreme cases the robustness of LQG (and MPC) to model uncertainty can be arbitrary poor. (Fun fact: The title of the paper is "Guaranteed margins for LQG regulators" and the extremely short abstract simply states: "There are none"). This is why the word "optimal" estimate was put in quotes above. The reason why the separation principle generally fails, is that it does not take into account the feedback created by the combined control and estimation. That is, the process input u resulting from the control problem affects the measurement y which affects the next state estimate, $\hat{x}$, which again affects the next u, and so on.

Having said this, it should be noted that practical experience has shown that LQG control (and MPC) usually has good robustness to model uncertainty, at least when tuned properly. For example, with LQG one may use the approach of "loop transfer recovery" (Stein & Athans, 1987) to recover most of the good robustness margins of LQ control (which assumes perfect measurements of all states) by using the weights in the estimation problem as tuning parameters (usually, to make the estimation fast). These weight then lose their original interpretation as representing the magnitude of the process and measurement noise.

The conclusion is that model predictive control is not as "optimal" as most academics would like to believe.

10.5.10. MPC research challenges

There has been a large academic effort over the last 30 years to extend the MPC theory (and in particular the numerical solutions) to include nonlinear systems, hybrid systems (mixed continuous and discrete) and systems with uncertainty. However, very little of this effort has had any impact on the industrial use of MPC, at least in the process industry where MPC originally was developed. New MPC applications in the process industry are still based mainly on linear experimental models, often derived from step responses, and using the MPC algorithms developed by the MPC vendors in the 1980s and 1990s. Strangely, the use of nonlinear physical models (and nonlinear MPC) has yet to find much use in the process industry. This is strange because it it time consuming and costly to obtain experimental linear models. The academic MPC research, especially for nonlinear systems, has probably had more impact on the control of mechanical systems. One reason is that it is usually much easier to derive physical models for mechanical systems, and also that the control solution can be duplicated on many identical plants (e.g., cars). On the other hand, most processing plants are one of a kind. However, even for mechanical systems, like automotive and flight control systems, the simpler approaches based on advanced regulatory control are still dominating in practical applications (although this does not seem to be the case when reading academic papers), and they are not likely to disappear in the future because of their effectiveness and simplicity.

One reason why academic researchers are attracted to MPC solutions is that they are viewed as being optimal and general. However, this is a myth. As explained above (Section 10.5.9), they are not as general and "optimal" as many academic researchers would like to believe, because the separation principle does not hold. I remember something Professor John Doyle said in 1985 at Caltech when I was a student: "There is two ways a theorem can be wrong. Either it's simply wrong or the assumptions make no sense". In this case, the "wrong" assumption is that all the states are measured or that they can be estimated optimally by solving a separate estimation problem (which does not consider how the estimates are used by the controller). This is the reason the word "optimal" is put in quotes.

In general, to be optimal (without quotes), the tasks of control and estimation need to be combined into one controller block, that is, to find a "control law" that directly connects measurements y and inputs u. However, both for nonlinear systems and for linear systems with uncertainty (and especially for nonlinear uncertain systems) this is an unsolved problem. To understand this better, note that the best tool for linear uncertain systems with unstructured and parametric uncertainty is the (real) structured singular value μ, but the use of μ is only reliable for analysis, and even this problem is NP-hard (Braatz et al., 1994). For design, there is no method with guaranteed convergence to find the μ-optimal controller $C(s)$ for an uncertain linear system. The available DK-iteration method frequently diverges (e.g., Skogestad & Postlethwaite (2005)) and when it converges it results in an optimal controller approaching infinite order. Since this is the best we can do for linear uncertain systems, it means that none of the available MPC stability and design results hold rigorously (in terms of being tight and optimal) for realistic uncertain systems, not even in the linear case.

With MPC, to restrict the controller order, one may represent the uncertainty using a multi-model or scenario approach, but this is generally optimistic (and may even give instability), because the worst case, for example, the worst-case time delay, may be an intermediate value which is not in the assumed model set.

A completely different approach is to restrict the set of allowed control laws (including fixing the order of the controller) and search for the best controller parameters, e.g., multivariable PID parameters. However, this gives a very hard mathematical problem. The simplest is to use proportional control, $u = Ky$, and search for the optimal matrix K. However, even in the linear case with no uncertainty and a quadratic objective, the optimal static output feedback problem is unsolved and believed to be non-convex and NP-hard. (e.g., Sadabadi & Peaucell (2016)). This illustrates that the controller design problem does not become simpler by imposing limitations on the controller, like limiting the order (static output feedback) or requiring decentralized control (corresponding to specifying zero elements in the controller C). On the contrary, decentralized controllers are actually more complex to synthesize and implement than their centralized counterparts (e.g., Anderson et al. (2019)).

The mathematical problem is therefore usually simplified by *removing* decomposition restrictions, for example, by combining the control layers in Figure 8 into a single Economic MPC (EMPC). This makes is tempting for academic researchers to propose the use of EMPC, but for practical implementation and tuning this combination of layers is rarely a good solution. Thus, EMPC should only be used for small problems or if it is really necessary, for example, if we cannot achieve acceptable time scale separation between the optimization and control layers.

The reason for including this discussion section on MPC research, is not say that people should stop research on MPC or EMPC. On the contrary, impressing progress has been made over the last 30 years, for example, on the numerical efficiency and robustness of nonlinear MPC. Rather, the discussion is included to point out that for real systems with model uncertainty, (conventional) MPC is a sub-optimal (and ad hoc) solution. Unfortunately, since the underlying problem of finding the optimal control law for an uncertain system is NP-hard, even for linear systems, there is little hope that this will ever change.

The conclusion is that model predictive control will never be as ”optimal” as most academics would like to believe.

Therefore, it is worthwhile for the academic control community to focus some research on the simpler (also ad hoc) “advanced regulatory control” elements described in this paper. The potential of these simpler solutions has been repeatedly demonstrated by engineers over the last 100 years who have designed workable (although certainly not optimal) control systems for very complex and difficult real processes. The aim of this research should be to improve the understanding and develop design methods for these simpler solutions.

10.6. Simplicity, the KISS principle and fragility

The KISS principle (Keep it simple stupid) states that most systems work best if they are kept simple rather than made complicated; therefore, simplicity should be a key goal in design, and unnecessary complexity should be avoided. Leonardo da Vinci stated that “Simplicity is the ultimate sophistication”. Albert Einstein is claimed to have said: “Make everything as simple as possible, but not simpler”. Steve Jobs said “~~Simplify, Simplify,~~ Simplify”, which simplified Henry David Thoreau’s quote “Simplify, simplify, simplify.” for emphasis. A related idea is Occam’s razor which says that the simplest explanation is usually the best one. All of this is according to Wikipedia (20 March 2023).

The KISS principle is widely accepted in most engineering disciplines, including industrial process control, but it does seem to be accepted as a goal within the academic control community. There are a few exceptions. Rosenbrock (1974) writes: “A good design usually has strong aesthetic appeal to those who are competent in the subject” and “The act of specifying the requirements in detail implies the final solution, yet has to be done in ignorance of this solution, which can then turn out to be unsuitable in ways that were not foreseen." John Doyle calls this sensitivity of an optimized solution to unforeseen events for "fragility", and he has coined the phrase "robust yet fragile" (Doyle et al., 2005). Carlson & Doyle (1999) state that a system designed for "highly optimized tolerance" with "high efficiency, performance, and robustness to designed-for uncertainties" (i.e., it appears very robust) tends to have "hypersensitivity to design flaws and unanticipated perturbations" (i.e., it is extremely fragile).

The justification for both the KISS principle and the "robust yet fragile" nature of highly optimized designs of complex systems is more on a philosophical than mathematical level, but it is based on experience from widely different systems, including control systems, biological systems and the internet.

In terms of control, simple control systems tend to be less fragile, mainly because they rely more on feedback from the real process and thus are less sensitive to errors in the model, and because they have fewer parameters that can be optimized to give unforeseen behavior. In addition, simple systems are easier to correct if an unforeseen event happens.

Only when these simple solutions become too "complex" or cannot solve the problem, should one consider more centralized model-based solution, like MPC. Of course, there is no clear definition of what "complex" is, and the tendency of the academic community has been to dismiss many workable industrial solutions as being complex, although this may not really be the case.

MPC solutions (and especially centralized EMPC solutions) tend to be "highly optimized" for a given problem definition, and have the danger of being "robust yet fragile". In addition, MPC solutions may be costly to implement and maintain. However, MPC solutions may serve as a benchmark for simpler solutions, like advanced regulatory control (ARC) elements. This can be used as a basis for improving the simple ARC solution or, if the performance loss is large, for concluding that MPC is a better solution.

11. Conclusion and challenges to the academic control community

The topic of the is paper is on classical or industrial advanced control, here denoted advanced regulatory control (ARC). These industrial solutions are based on decomposing the overall controller into simple control elements. By doing this, the engineer directly specifies the control solution (structure), and the tuning parameters in each control element usually have a direct and clear effect on the system responses. In addition, the modeling requirements are much less than with model-based methods like MPC. Instead, the engineer uses structural information (e.g., the process flowsheet), process insight and information about constraints and control objectives.

This means that it is possible to propose a control strategy (flowsheet with controllers) at an early stage, long before the process is build. Actually, a workable control strategy together with a startup procedure, is required before a decision is made to start detailed design of a new process plant. Later in the project, the control strategy is further developed into the process & instrumentation diagram (detailed flowsheet with controllers). Furthermore, by scaling the variables and using simple dynamic models or just insight about the dominant dynamics, initial "default tunings" may be proposed for most control loops (e.g., Smuts (2011),. p. 303). The fine-tuning of the controllers may be done sequentially during startup using experimental data.

These ARC solutions have proven their success in industrial applications over the last 100 years, in spite of receiving little academic attention. The lack of academic attention, implies that students have not received proper training in these methods, and that proper design methods have not been developed. At the moment, the control engineer is pretty much left in the dark, with the main source of knowledge into advanced regulatory control solutions being "pattern recognition" based on previous designs.

The academic control community can help rectify this and there is a large potential for improvements. In addition to mathematical generality and rigor, the research goal should include the industrial use and benefit of the technology, where decomposition and simplicity is important. Simple control solutions are easier to implement, understand, tune (and retune) and change.

The list of standard elements of advanced regulatory control (E1-E18) given in the introduction provide a good starting point for the research. Additional more specialized solutions have been proposed over the years, in particular, by Greg Shinskey, but these solutions have often been dismissed as being complex and ad hoc. Rather, Greg Shinskey should be recognized as a an important innovator and source of ideas, and efforts should be spent on understanding and expanding his solutions and developing theory to make them less ad hoc.

Here is an incomplete list of possible research topics, which are important but have received limited (or no) academic attention:

1. Time scale separation in hierarchically decomposed systems (considering performance and robustness)

2. Selection of variables that link the different layers in the control hierarchy, for example, self-optimizing variables (CV1 in Figure 8) and stabilizing variables (CV2). Selection of intermediate controlled variables (w) in a cascade control system[7].

3. Tuning of cascade control systems (Figures 13 and 14)

4. Structure of selector logic

5. Tuning of anti-windup schemes (optimal choice of tracking time constant, τ_T) for input saturation, selectors, cascade control and decoupling.

6. New basic control elements

7. How can we make decomposed control systems based simple elements easily understandable to operators and engineers?

[7]Note that it may be possible (and desirable) to have the same variable being controlled twice in the same cascade hierarchy. For example, one may have two pressure controllers ($y = p$) on top of each other (one fast for stabilization and one slow for optimization with sets the setpoint to the fast controller), or there may be a VPC in between (with $w = u$) so that pressure is "floating" (uncontrolled) on an intermediate time scale.

8. Simple schemes for decoupling and feedforward control

9. Default tuning of PID controllers (including scaling of variables) based on limited information

10. Comparison of selector on input or setpoint (cascade)

11. A concise list of special (smart) control structures (inventions) that solve specific control problems, for example, cross-limiting control.

12. Case studies that compare alternative solutions, for example, the three solutions for MV-MV switching.

What about research on PID tuning? Except for the problem of "default tunings", PID tuning has probably received enough academic attention. One exception may be oscillating systems, but these are rare in process control, provided robust tunings are used in the lower-layer control loops. In addition, both for unstable and oscillating processes, a better approach may be to use a cascade (see footnote 7) of a fast inner P- or PD- controller which stabilizes or removes oscillations and a slower PID-controller which changes the setpoint to the inner loop. In summary, "PID control" researchers are recommended to switch their attention to "advanced PID control", that is, the interconnection of the PID controller with the other advanced control elements.

The above list of research topics deals mainly with the individual elements. A much tougher research issue is the design of an overall decomposed control structure, that is, the interconnection of the simple control elements for a particular application. This area definitely needs some academic efforts.

One worthwhile approach is case studies. That is, to propose "good" (= effective and simple) control strategies for specific applications, for example, for a cooling cycle, a distillation column, or an integrated plant with recycle. It is here suggested to design also a centralized controller (e.g., MPC) and use this as a benchmark to quantify the performance loss (or maybe the benefit in some cases) of the simpler decomposed ARC solution. A related issue, is to suggest new smart approaches to solve specific problems, as mentioned in item 11 in the list above. Maybe a new case-study based journal, with a title like "Journal of smart control structures", could be a good idea.

A second approach is mathematical optimization: Given a process model, how to optimally combine the control elements E1-E18 to meet the design specifications. However, even for small systems, this is a very difficult combinatorial problem, which easily becomes prohibitive in terms of computing power. It requires both deciding on the control structure as well as tuning the individual PID controllers.

As a third approach, machine learning may be useful. Machine learning has one of its main strength in pattern recognition, in a similar way to how the human brain works. I have observed over the years that many students, with only two weeks of example-based teaching, are able to suggest good process control solutions with feedback, cascade, and feedforward/ratio control for realistic problems, based on only a flowsheet and some fairly general statements about the control objectives. This is the basis for believing that machine learning (e.g., a tool similar to ChatGPT) may provide a good initial control structure, which may later be improved, either manually or by optimization.

The paper has gone into some detail about the shortcomings of MPC. This criticism should not really have been necessary in a paper about advanced regulatory control (ARC), because because both MPC and ARC should be in the toolbox of all control engineers. However, a discussion about MPC shortcomings is included because many academic researchers think that the industrial approaches (ARC) are outdated and ad hoc and will be replaced by MPC. As argued in this paper, this will not happen, partly because MPC is itself an ad hoc solution for many simple control tasks (like simple feedback with integral action (PID control), cascade control and ratio control) and partly because the effort to obtain the model and define the MPC problem may be too costly even for problems where MPC is the better solution in terms of performance.

In summary, it is proposed that a lot more academic research is focused on developing theory for the advanced regulatory control solutions described in this paper. The problems are very challenging. For example, the mathematical problems related to the optimal decomposed and decentralized control solutions are in general non-convex, and the stability analysis of switched systems (for example, with selectors, anti-windup and split range control) is very difficult as it may result in limit cycles and chaotic behavior. This, in addition to an unclear problem definition, may scare academic researchers away, but hopefully the importance of the problem and the prospect of seeing the solutions being used in practice and thus benefiting humanity, may provide motivation to consider these important and challenging problems.

Acknowledgements

The author gratefully acknowledges fruitful discussions and inputs from Cristina Zotică, Krister Forsman, Adriana Reyes-Lúa, Lucas Ferreira Bernardino, Dinesh Krishnamoorthy and Risvan Dirza.

References

Allison, B. J., & Isaksson, A. J. (1998). Design and performance of mid-ranging controllers. *Journal of Process Control*, *8*, 469–474.

Allison, B. J., & Ogawa, S. (2003). Design and tuning of valve position controllers with industrial applications. *Transactions of the Institute of Measurement and Control*, *25*, 3–16.

Alstad, V., Skogestad, S., & Hori, E. (2009). Optimal measurement combinations as controlled variables. *Journal of Process Control*, *19*, 138–148.

Anderson, J., Doyle, J. C., Low, S. H., & Matni, N. (2019). System level synthesis. *Annual Reviews in Control*, *47*, 364–393.

Aske, E., & Skogestad, S. (2009). Consistent inventory control. *Ind. Eng. Chem. Res.*, *48*, 10892–10902.

Balchen, J. G., & Mumme, K. (1988). *Process Control. Structures and Applications*. New York: Van Nostrand Reinhold.

Bemporad, A., Morari, M., Dua, V., & Pistikopoulos, E. N. (2002). The explicit linear quadratic regulator for constrained systems. *Automatica*, *38*, 3–20.

Bennett, S. (1988). Nicolas Minorsky and the automatic steering of ships. *Control Systems magazine*, (pp. 10–15).

Bernardino, L. F., Krishnamoorthy, D., & Skogestad, S. (2022). Optimal operation of heat exchanger networks with changing active constraint regions. *PSE Symposium in: Computer Aided Chemical Engineering (Elsevier)*, *49*, 421–426.

Bernardino, L. F., & Skogestad, S. (2023). Bidirectional inventory control with optimal use of intermediate storage and minimum flow constraints. *IFAC World Congress, Japan)*, *1000*, 1–1.

Blickley, G. (1990). Modern control started with Ziegler-Nichols tuning. *Control Engineering*, (pp. 10–15).

Bode, H. W. (1945). *Network Analysis and Feedback Amplifier Design*. New York: Van Nostrand.

Braatz, R. D., Young, P. M., Doyle, J. C., & Morari, M. (1994). Computational complexity of μ calculation. *IEEE Transactions of Automatic Control*, *AC-39*, 1000–1002.

Bristol, E. H. (1966). On a new measure of interactions for multivariable process control. *IEEE Transactions on Automatic Control*, *AC-11*, 133–134.

Buckley, P. C. (1964). *Techniques of Process Control*. Wiley.

Cao, Y. (2004). Constrained self-optimizing control via differentiation. *Adchem synposium, Hong Kong, Jan. 2004, IFAC Proceedings Volumes*, *37*, 63–70.

Carlson, J. M., & Doyle, J. C. (1999). Highly optimized tolerance: A mechanism for power laws in designed systems. *Physical Review E*, *60*, 1412–1427.

Cohen, G., & Coon, G. (1953). Theoretical consideration of retarded control. *Trans. ASME*, *75*, 827–834.

Cutler, C., & Ramaker, B. (1980). Dynamic matrix control - a computer control algorithm. *Joint automatic conference (ACC), San Francisco*, .

Dirza, R., Skogestad, S., & Krishnamoorthy, D. (2021). Optimal resource allocation using distributed feedback-based real-time optimization. *Adchem synposium, IFAC PapersOnLine*, *54*, 706–711.

Downs, J. J., & Skogestad, S. (2011). An industrial and academic perspective on plantwide control. *Annual Reviews in Control*, *35*, 99–110.

Doyle, J. C. (1978). Guaranteed margins for LQG regulators. *IEEE Transactions on Automatic Control*, *AC-23*, 756–757.

Doyle, J. C., Alderson, D. L., Li, L., Low, S., Roughan, M., Shalunov, S., Tanaka, R., & Willinge, W. (2005). The "robust yet fragile" nature of the Internet. *Proceedings of the National Academy of Sciences of the United States of America (PNAS)*, *102*, 14497–14502.

Eckman, D. P. (1945). *Principles of industrial process control*. New York: Wiley.

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

Foss, A. S. (1973). Critique of chemical process control theory. *AIChE Journal*, *19*, 209–214.

Freudenberg, J., & Middleton, R. (1999). Properties of single input, two output feedback systems. *Int. J. Control*, *72*, 1446–1465.

Grimholt, C., & Skogestad, S. (2012). Optimal PI-control and verification of the SIMC tuning rule. *IFAC Conference on Advances in PID Control (PID-2012), IFAC Proceedings Volumes*, *45*, 11–22.

Grimholt, C., & Skogestad, S. (2018a). Optimal PI and PID control of first-order plus delay processes and evaluation of the original and improved SIMC rules. *J. Process Control*, *70*, 36–46.

Grimholt, C., & Skogestad, S. (2018b). Should we forget the Smith Predictor? *3rd IFAC Conference on Advances in PID Control, Ghent, Belgium, IFAC Papers Online*, *51*, 769–774.

Grosdidier, P., Morari, M., & Holt, B. R. (1985). Closed-loop properties from steady-state gain information. *Industrial and Engineering Chemistry Fundamentals*, *24*, 221–235.

Hägglund, T., & Guzman, J. (2018). Development of basic process control structures. *IFAC Papers Online*, *51*, 775–780.

Ingimundarson, A., & Hägglund, T. (2002). Performance comparison between PID and dead-time compensating controllers. *Journal of Process Control*, *12*, 887–895.

Jäschke, J., Cao, Y., & Kariwala, V. (2017). Self-optimizing control–a survey. *Annual Reviews in Control*, *43*, 199–223.

Krishnamoorthy, D., & Skogestad, S. (2020). Systematic design of active constraint switching using selectors. *Computers and Chemical Engineering*, *143*, 107106.

Krishnamoorthy, D., & Skogestad, S. (2022). Real-time optimization as a feedback control problem – a review. *Computers and Chemical Engineering*, *161*, 107723.

Liptak, B. (Ed.) (1973). *Instrumentation in the process industries*. Chilton Book Co.

Liptak, B. G. (1999). *Optimization of industrial unit processes, 2nd Edition*. (2nd ed.). CRC press.

Lundström, P., Lee, J., Morari, M., & Skogestad, S. (1995). Limitations of dynamic matrix control. *Comp. & Chem. Engng*, *19*, 409–421.

Luyben, W. L., Tyerus, B. D., & Luyben, M. L. (1998). *Plantwide process control*. McGraw-Hill.

Markaroglu, H., Guzelkaya, M., Eksin, I., & Yesil, E. (2006). Tracking time adjustment in back calculation anti-windup scheme. *Proceedings 20th European Conference on Modelling and Simulation*, .

McAvoy, T. J. (1983). *Interaction Analysis - Principles and Applications*. Instrument Society of America (ISA).

Morari, M., Arkun, Y., & Stephanopoulos, G. (1980). Studies in the synthesis of control structures for chemical processes, Part I: Formulation of the problem. Process decomposition and the classification of the control tasks. Analysis of the optimizing control structures. *AIChE Journal*, *26*, 220–232.

Morari, M., & Stephanopoulos, G. (1980a). Studies in the synthesis of control structures for chemical process, Part II: Structural aspects and the synthesis of alternative feasible control schemes. *AIChE Journal*, *26*, 232–246.

Morari, M., & Stephanopoulos, G. (1980b). Studies in the synthesis of control structures for chemical process, Part III: Optimal selection of secondary measurements within the framework of state estimation in the presence of persistent unknown disturbances. *AIChE Journal*, *26*, 247–260.

Nagy, I. (1992). *Introduction to chemical process instrumentation*. Elsevier.

Pannocchia, G., Laachi, N., & Rawlings, J. B. (2005). A candidate to replace pid control: SISO-constrained LQ control. *AIChE Journal*, *51*, 1178–1189.

Price, R. M., Lyman, P., & Georgakis, C. (1994). Throughput manipulation in plantwide control structures. *Ind. Eng. Chem. Res.*, *33*, 1197–1207.

Rawlings, J. B. (2000). Tutorial overview of model predictive control. *IEEE Control Systems Magazine*, *20*, 38–52.

Reyes-Lúa, A., & Skogestad, S. (2019). Systematic design of split range controllers. *Processes*, *7*, 941.

Reyes-Lúa, A., & Skogestad, S. (2020a). Multi-input single-output control for extending the operating range: Generalized split range control using the baton strategy. *Journal of Process Control*, *91*, 1–11.

Reyes-Lúa, A., & Skogestad, S. (2020b). Systematic Design of Active Constraint Switching Using Classical Advanced Control Structures. *Ind. Eng. Chem. Res.*, *59*, 2229–2241. doi:10.1021/acs.iecr.9b04511.

Reyes-Lúa, A., Zotică, C., Forsman, K., & Skogestad, S. (2019). Systematic design of split range controllers. *DYCOPS conference, IFAC-PapersOnLine*, *52*, 893–903.

Richalet, A., Raulet, Testud, J., & Papon, J. (1978). Model predictive heuristic control: Apllications to industrial processes. *Automatica*, *14*, 413–428.

Rivera, D., Morari, M., & Skogestad, S. (1986). Internal model control. 4. pid controller design. *Ind. Eng. Chem. Res.*, *25*, 252–265.

Rosenbrock, H. H. (1974). *Computer-Aided Control System Design*. New York: Academic Press.

Sadabadi, M. S., & Peaucell, D. (2016). From static output feedback to structured robust static output feedback: A survey. *Annual Reviews in Control*, *42*, 11–126.

Seborg, D. E., Edgar, T. F., Mellichamp, D. A., & III, F. J. D. (2016). *Process Dynamics and Control*. (4th ed.). Wiley.

Shinskey, F. G. (1967). *Process control systems*. McGraw-Hill.

Shinskey, F. G. (1978). *Energy conservation through control*. Academic Press.

Shinskey, F. G. (1979). *Process control systems*. (2nd ed.). McGraw-Hill.

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

Skogestad, S. (1991). Consistency of steady-state models using insight about extensive variables. *Ind. Eng. Chem. Res.*, *30*, 654–661.

Skogestad, S. (2000). Plantwide control: The search for the self-optimizing control structure. *Journal of Process Control*, *10*, 487–507.

Skogestad, S. (2003). Simple analytic rules for model reduction and PID controller tuning. *Journal of Process Control*, *13*, 291–309.

Skogestad, S. (2023). The theoretical basis of ratio control. *Ind. Eng. Chem. Res. or JPC*, *In progress*.

Skogestad, S., & Postlethwaite, I. (1996). *Multivariable Feedback Control: Analysis and Design*. (1st ed.). Wiley.

Skogestad, S., & Postlethwaite, I. (2005). *Multivariable Feedback Control: Analysis and Design*. (2nd ed.). Wiley.

Skogestad, S., Zotică, C., & Alsop, N. (2023). Transformed inputs for linearization, decoupling and feedforward controller. *Journal of process Control*, *122*, 113–133.

Smith, C. A., & Corripio, A. (1985). *Principles and Practice of Automatic Process Control*. New York: Wiley.

Smith, C. L. (2010). *Advanced Process Control - Beyond single-loop control*. New York: Wiley.

Smith, O. (1957). Closed control of loops with dead time. *Chemical Engineering Progress*, *53*, 217–219.

Smuts, J. F. (2011). *Process control for practitioners*. OptiControls.

Stein, G., & Athans, M. (1987). The LQG/LTR procedure for multivariable feedback control design. *IEEE Transactions on Automatic Control*, *AC-32*, 105–114.

Storkaas, E., & Skogestad, S. (2004). Cascade control of unstable systems with application to stabilization of slug flow. *Adchem synposium, Hong Kong, Jan. 2004, IFAC Proceedings Volumes*, *37*, 335–340.

Wade, H. L. (1997). Inverted decoupling: A neglected technique. *ISA Transactions*, *36*, 3–10.

Wade, H. L. (2004). *Basic and Advanced Regulatory Control: System Design and Application*. (2nd ed.). ISA.

Young, A. J. (1955). *An introduction to process control system design*. Longmans.

Ziegler, J. G., & Nichols, N. B. (1942). Optimum settings for automatic controllers. *Transactions of the A.S.M.E.*, *64*, 759–768.

Zotică, C., Forsman, K., & Skogestad, S. (2022). Bidirectional inventory control with optimal use of intermediate storage. *Computers and Chemical Engineering*, *159*, 107677.

Åström, K. J., & Hägglund, T. (1988). *Automatic tuning of PID controllers*. Research Triangle Park: ISA.

Åström, K. J., & Hägglund, T. (2006). *Advanced PID control*. Research Triangle Park: ISA.