Contents lists available at ScienceDirect

Annual Reviews in Control

journal homepage: www.elsevier.com/locate/arcontrol

Review article

Advanced control using decomposition and simple elements

Sigurd Skogestad

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

ARTICLE INFO

Keywords:
Control structure design
Feedforward control
Cascade control
PID control
Selective control
Override control
Time scale separation
Decentralized control
Distributed control
Horizontal decomposition
Hierarchical decomposition
Layered decomposition
Vertical decomposition
Network architectures

ABSTRACT

The paper explores the standard advanced control elements commonly used in industry for designing advanced control systems. These elements include cascade, ratio, feedforward, decoupling, selectors, split range, and more, collectively referred to as "advanced regulatory control" (ARC). Numerous examples are provided, with a particular focus on process control. The paper emphasizes the shortcomings of model-based optimization methods, such as model predictive control (MPC), and challenges the view that MPC can solve all control problems, while ARC solutions are outdated, ad-hoc and difficult to understand. On the contrary, decomposing the control systems into simple ARC elements is very powerful and allows for designing control systems for complex processes with only limited information. With the knowledge of the control elements presented in the paper, readers should be able to understand most industrial ARC solutions and propose alternatives and improvements. Furthermore, the paper calls for the academic community to enhance the teaching of ARC methods and prioritize research efforts in developing theory and improving design method.

Contents

E-mail address: Sigurd.Skogestad@ntnu.no.

https://doi.org/10.1016/j.arcontrol.2023.100903
Received 25 March 2023; Received in revised form 3 July 2023; Accepted 25 July 2023

1. Introduction

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

- Advanced regulatory control (ARC): Decomposed control system using standard control elements, including PID controllers.
- Model predictive control (MPC): On-line optimizing control using a dynamic process model.

This paper focuses on the first approach and how one may put together standard control elements to control complex multivariable nonlinear constrained industrial processes. In addition, the objective of the paper is to point out the need to significantly increase teaching and research in this important area.

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.

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, electronics, marine, aerospace, power, medical, and agriculture.

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 a need for some "pre-act" (derivative) action. The first paper on chemical process control is Grebe et al. (1933) from the Dow Chemical Company who give an excellent status on the measuring instruments and control techniques available in the US process industry at the time. At the end of the paper, Grebe adds a comment on the need for setting standards for control. He writes: "You will notice all the while that I have been stumbling over words. The conceptions back of it all are the same, ... so if the Institute [of Chemical Engineers] can do anything about getting instrument manufacturers together to define and set standards for control, let us do it".

Minorsky introduced a three-term PID controller for steering of ships as early as 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 followed up 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 only three years after its introduction, John Ziegler and Nathaniel Nichols (both from Taylor Instrument Co.) published their groundbreaking 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 on 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, by far the 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 control elements became widely used in the process industry[1]:

1. PID control, and in particular the use of integral action to reset the bias
2. Cascade control
3. Ratio control

[1] In the opinion of the author, these are the three main inventions of process 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 18 standard "advanced" control elements listed below were used in the process industry (in addition to simple PID and on/off feedback controllers).

1.1. List of advanced control elements

First, there are some elements that are used to improve control for cases where simple feedback control is not sufficient:

E1*. Cascade control[2]
E2*. Ratio control
E3*. Valve (input)[3] position control (VPC) on extra MV to improve dynamic response,

Next, there are some control elements used for cases when we reach constraints:

E4*. Selective (limit, override) control (for output switching)
E5*. Split range control (for input switching)
E6*. Separate controllers (with different setpoints) as an alternative to split range control (E5)
E7*. VPC as an alternative to split range control (E5)

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, 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*. Calculation 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, absolute value, 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 (also see summary in Table 1) then we note that each element links a specific subset of inputs to a specific subset of outputs. Thus, this results in a decomposed control system and the control engineer needs to make structural pairing decisions to use the standard elements. This makes it difficult to handle very interactive processes where the pairing is not obvious, so here model-based methods, like MPC, may be preferred. On the other hand, an important advantage with fixed pairings is that the engineer can specify more directly how the system responds in a given situation.

1.2. The industrial and academic control worlds

The above list of control elements makes up 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* (ARC).

Almost in a different universe, 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) which was already mentioned. The other was the academic book by Hendrik Wade Bode on "Network Analysis and Feedback Amplifier Design" (Bode, 1945). Although both books deal 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 makes use of Laplace transforms and transfer functions which 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 1980s, most academic researchers have switched from Laplace transforms to the time domain for research, both for numerical reasons and to handle nonlinear systems. This is closer to the approach used in the industrial world, but otherwise the two worlds have remained largely separated.

The only academic control approach which is presently widely used in the process industries is model predictive control (e.g., see Tables 2 and 3 in Samad et al. (2020)). The present state-space version of MPC is a result of a fusion between two heuristic (at least originally) industrial approaches for repeated on-line feedforward optimizing control from the 1970s, namely Model Predictive Heuristic Control of Richalet et al. (1978) and Dynamic Matrix Control (DMC) of Cutler and Ramaker (1980), and the academic optimal control theory (LQG control) of Kalman and others of the 1960s. MPC has been in industrial use since the mid 1970s and it became common in the petrochemical and refining industry at the end of the 1980s. However, in spite of a large academic focus on MPC since about 1990, its adaptation into other process industries has been significantly slower than was anticipated in the 1990s.

In summary, based on the author's experience, the advanced regulatory control elements listed above, remain the main tool for advanced control in most process industries (except refining and petrochemicals). 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, Åström and their coworkers) develop improved methods to replace the Ziegler–Nichols tuning rules from 1942.

[2] The control elements with an asterisk * are discussed in more detail in this paper.

[3] In this paper, Valve Position Control (VPC) refers to cases where the input (independent variable) is controlled to a given setpoint ("ideal resting value") on a slow time scale. Thus, the term VPC is used for other inputs (actuator signals) than valve position, including pump power, compressor speed and flowrate, so a better term might have been Input Position Control.

Table 1
Standard advanced control elements studied in this paper.

Control element	Main use	Inputs	Outputs
E1. Cascade control Figs. 9 and 10	Linearization and local disturbance rejection	Outer master controller: • CV_{1s}-CV Inner controller: • CV_{2s}-CV_2	Outer master controller: • CV_{2s} Inner controller: • MV
E2. Ratio control Fig. 11	Feedforward or decoupling without model (assumes that scaling property holds)	• R (desired ratio) • DV or MV_1	• MV = R · DV, or • MV_2 = R · MV_1
E3. VPC on extra dynamic input Fig. 12	Use extra dynamic input MV_1 to improve dynamic response (because MV_2 alone is not acceptable). MV_1 setpoint is unconstrained (mid-range) and controlled all the time	• MV_{1s} - MV_1	• MV_2
E4. Selector Figs. 17, 18 and 19	CV-CV switching: Many CVs (CV_1, CV_2, ...) controlled by one MV	• MV_1 • MV_2, ... (generated by separate controllers for CV_1, CV_2, ...)	• MV = max/min (MV_1, MV_2, ...)
E5. Split-range control Figs. 21 and 23	MV-MV switching: One CV controlled by sequence of MVs (using only one controller)	• CV_s-CV	• MV_1 • MV_2, ...
E6. Separate controllers with different setpoints Fig. 22	MV-MV switching: One CV controlled by sequence of MVs (using individual controllers with different setpoints)	• CV_{s1} – CV • CV_{s2} - CV ...	• MV_1 • MV_2 ...
E7. VPC on main steady-state input Fig. 24	MV-MV switching: One CV controlled by main MV_1 with use of extra MV_2 to avoid saturation of MV_1. MV_1 setpoint is close to constraint and only controlled when needed	• MV_{1s} - MV_1	• MV_2
E9. Two degrees-of-freedom feedback controller Fig. A.41	Treat setpoint (CV_s) and measurement (CV) differently in controller C	• CV_s • CV	• MV
E11. Feedforward control Fig. A.42	Reduce effect of disturbance (using model from DV and MV to CV)	• DV	• MV
E12. Decoupling element Fig. 26	Reduce interactions (using model from MV_1 and MV_2 to CV)	• MV_1 • MV_2	• MV_2 • MV_1
E14. Calculation block based on transformed input Fig. 27	Static nonlinear feedforward, decoupling and linearization based on nonlinear model from MV, DV and w to CVv	• Transformed input = feedback trim (v) • DV (d) • Extra meas. (w)	• MV (u)

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, many of the ARC problems are challenging theoretically, for example, decomposition and decentralized control (including the pairing problem) and the behavior of switched systems (which may display limit cycles and even chaotic behavior). The third reason, as pointed out by Foss (1973) in his famous paper with the title "Critique of Chemical Process Control Theory", is that the academic community has largely neglected the structural issues, that is, the decisions on what to control (outputs, CVs) and how to decompose the system into single-variable 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.

1.3. Previous work on Advanced regulatory control

Following the paper of Foss (1973), some research was initiated on control structure design and "chemical plant(wide) control", for example, the three-part series (Morari et al., 1980), Morari and Stephanopoulos (1980a) and Morari and Stephanopoulos (1980b)). They introduced the concept of feedback-optimizing control. The main idea is to move the optimization into the control layer by selecting good controlled variables (CVs). One should always control 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 time (in the 1980s), MPC became popular and many academic researchers expected that MPC would soon replace the seemingly ad-hoc and complex industrial "advanced PID" structures. Therefore, with a few exceptions, the academic research efforts on

structural issues and more generally on advanced regulatory control died away during the 1990s. Reviews of some academic research on control structure design and advanced regulatory control are found in Chapter 10 in Skogestad and Postlethwaite (2005) and in Skogestad (2015). Good overviews of the current industrial status on advanced regulatory control are found in the books "Basic and advanced regulatory control" by Wade (2004), "Advanced process control - beyond single-loop control" by Smith (2010), and "Process control - a practical approach" by King (2011). A good source of process control case studies are the many papers and books by Bill Luyben, e.g., Luyben et al. (1998).

1.4. Motivation for studying advanced regulatory control

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 and Guzmán (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".

Do we really need a theory for advanced regulatory control (ARC) when it seems to be working well already? Yes, we do. First, the fact that it is working, does not mean that it is "good" (where "good" here means "close to optimal"). Second, without theory, it is difficult to improve the methods and suggest alternatives. Third, without some theory, teaching becomes difficult. Fourth, the expertise to apply ARC may be disappearing from the process industry. Myke King writes based on his experience from the oil and petrochemical industries (King, 2011) (page x): "MPC has rightly replaced many of the more complex ARC techniques, but it has been used by too many as the panacea to any control problem. There remain many applications where ARC outperforms MPC; but appreciation of its advantage is now hard to find in industry. The expertise to apply it is even rarer".

The aim of this paper is to present the various standard ARC elements and illustrate their use, with particular emphasis on how to handle changes in active constraints.

1.5. Notation

The most important notation is summarized in Figs. 1 and 2. The physical process to be controlled (Fig. 1) has as independent variables the input u and the disturbance d and as measured dependent variables the output y (with reference value or setpoint y_s) and the state w. The feedback controller in Fig. 2 has as inputs (independent variables) the controlled variable (CV) and its setpoint (CV_s), and as output the manipulated variable (MV). This is called a two degrees-of-freedom controller because the controllers acts independently on CV and CV_s. A common one degree-of-freedom negative feedback control system is shown in Fig. 3. Here the controllers acts on the error signal CV_s-CV= $y_s - y$. In this paper, the controller C is usually a PID controller.

Note the following for the rest of the paper:

- We often write y and u for the controller input and output signals (Fig. 3), although strictly speaking, with reference to Fig. 2, it would be more correct to write CV and MV.

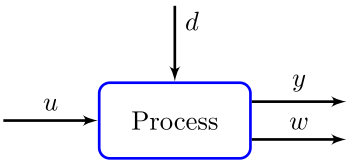


Fig. 1. Block diagram of general process (usually dynamic and nonlinear). The process block (physical equipment) usually includes also the actuator and measurement devices.
u = process input (actuator signal) (independent variable that can be manipulated)
d = disturbance (DV) (independent variable outside our control) (sometimes measured)
y = process output = (primary) process variable (PV) (sometimes called y_1) (dependent variable with setpoint y_s) (usually measured)
w = secondary process variable (y_2)= state variable (x) (usually measured).

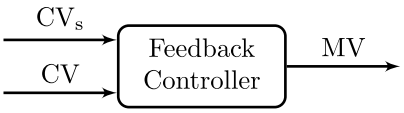


Fig. 2. Block diagram of general "two degrees-of-freedom" feedback controller (usually dynamic and possibly nonlinear). In the multivariable case, the feedback controller may consist of several simpler control elements.
CV = controlled variable (with setpoint CV_s) = controller input
MV = manipulated variable = controller output.

- In the block diagrams, all connecting black lines are signals (information). The controller blocks are also black, whereas the process blocks (physical equipment) are blue.
- We will also make use of flowsheets (simple Process & Instrumentation Diagrams). One example is shown in Fig. 11. Here, black color is used for process flows and process equipment, and red is used for signals and control elements.

Additional notation for feedback and feedforward control is given in Appendix A.

2. Decomposition of the control system

2.1. What is control?

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. With reference to the process block diagram in Fig. 1, 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.*

Importantly, disturbances (independent variables outside our control) are included in the definition, because in most cases these are the reason for why we need control. The word "active" is to emphasize that this is a dynamic system.

In addition, and this is not covered by the above definition, there is the fundamental difference between feedforward control and feedback control. Only feedback control can change the dynamics of the system, for example, to stabilize an unstable process. Also, feedback control is generally much more robust to model and measurement errors than feedforward control, as demonstrated by a simple example in Appendix B. The reader is recommended to look at this example, and especially to note that if one is not careful, then one may end up with feedforward control for cases where feedback control is much better; for example, this may happen when using model predictive control.

The word "setpoint" (= command) is included in the above definition of control. 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" where setpoints are replaced by economic optimality:

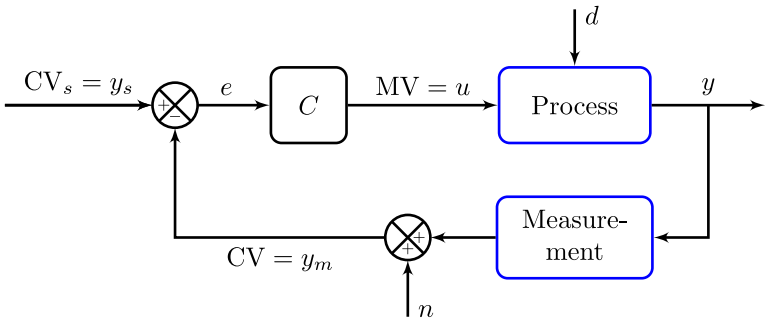


Fig. 3. Block diagram of common "one degree-of-freedom" negative feedback control system.
All three blocks are generally dynamic and nonlinear. The objective of the control system is to keep the process output y close its setpoint y_s in spite of disturbances d.
y_m = measured value of y
n = measurement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

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.

The above definition of "control" applies to the two control layers in Fig. 4 (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. Note that the decisions inside the blocks and layers above the regulatory layer are often manual, but this papers considers only automated decision making.

2.2. Decomposition approaches

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

1. Academics often propose to use one "big" optimizing controller (one layer). This is centralized optimizing control where the optimization and control objectives 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 with 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 may be little economic benefit of combining the optimization and control tasks.
2. Real control systems are decomposed into smaller blocks, as illustrated for process control in Fig. 4. Here, the separate layers for optimization and control are connected using setpoints. 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 (Fig. 4):

I Vertical (hierarchical; cascade) decomposition
II Horizontal (distributed/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 at different time scales, which 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 Fig. 4.

The horizontal decomposition makes use of distributed or decentralized controllers (Fig. 4) and is usually based on physical separation.

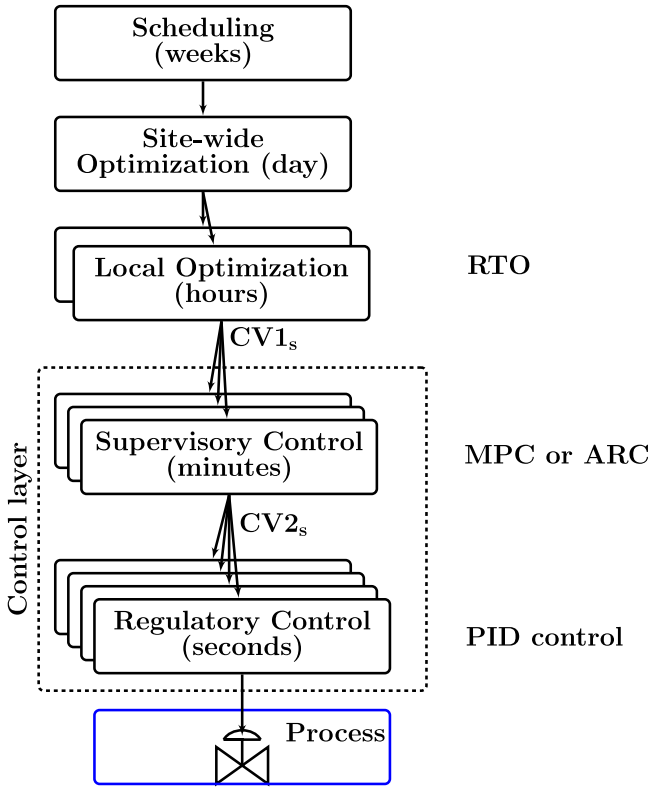


Fig. 4. Decomposition of "overall control system" for optimal operation in typical process plant. This involves a vertical (hierarchical) decomposition (Richalet et al., 1978) into decision layers based on time scale separation, and a horizontal decomposition into decentralized blocks/controllers, often based on physical distance. There is also feedback of measurements (y, w, d, CV1, CV2) (possibly estimates) from the process to the various layers and blocks but this is not shown in the figure. This paper considers the three lowest layers, with focus on the supervisory control layer.
CV1 = Economic controlled variables
CV2 = Regulatory/stabilizing controlled variables
RTO = Real-time optimization
MPC = Model predictive control
ARC = Advanced regulatory control
PID = Proportional–Integral–Derivative.

2.3. Structural decisions

To be able to decompose the control system into smaller blocks (Fig. 4), the engineer needs to make *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 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 corresponding 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, 2005).

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".

2.4. I. Vertical (hierarchical, layered, cascade) decomposition

Chiang et al. (2007) state about the vertical (layered) decomposition (in a paper mostly focusing on data networks but with direct relevance to control systems):

> "Layered architectures form one of the most fundamental structures of network design. They adopt a modularized and often distributed approach to network coordination. Each module, called layer, controls a subset of the decision variables, and observes a subset of constant parameters and the variables from other layers. Each layer in the protocol stack hides the complexity of the layer below and provides a service to the layer above. Intuitively, layered architectures enable a scalable, evolvable, and implementable network design, while introducing limitations to efficiency and fairness and potential risks to manageability of the network".

For process control applications, the three layers of main interest are (Fig. 4) (Richalet et al., 1978):

1. *Optimization layer* (real-time optimization, RTO). 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

$$J_\$ = p_F F - p_P P + p_Q Q \quad [\$/s] \tag{1}$$

   Here, F denotes feed streams (raw material) [kg/s], P denotes product streams [kg/s], Q utility (energy) usage [W=J/s], and p denotes the corresponding prices (in [$/kg] or [$/J]). The degrees of freedom (MVs) for the optimization layer are the setpoints ($CV1_s$) to the supervisory control layer.
2. *Supervisory ("advanced") control layer*. This layer is the main focus of this paper and it has three main objectives:
   - Follow the setpoints ($CV1_s = y_s$) coming from economic optimization layer. With MPC, we typically use a cost function of the form (for more details, see (23) below):

$$J_c = \sum_{k=0}^{k=N} (y_k - y_{s,k})^T Q (y_k - y_{s,k}) + \Delta u_k^T R \Delta u_k \tag{2}$$

   - Switch between active constraints (change CV1-variables)
   - Look after the regulatory layer (avoid that the physical inputs saturate, etc.)

   The degrees of freedom for the supervisory control layer include the setpoints ($CV2_s$) to the basic control layer and possibly some of the physical process inputs u.
3. *"Basic" regulatory control layer* (PID layer). The main objective of this layer 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. The degrees of freedom for this layer are the remaining physical process inputs u.

In Fig. 4, the setpoints $CV1_s$ and $CV2_s$ connect the layers and a key decision is what these variables should be. For example, consider a marathon runner. For the "economic optimization" (minimizing time), is it better to set the setpoint for the speed or heart rate of the runner (that is, should CV1 be speed or heart rate)?

In practice, the distinction between the various layers may not be so clear. In some cases, there is further vertical decomposition, for example, using cascade control. In other cases, especially in academic studies, the two control layers are combined. In industry, there is usually 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.

For *automatic* supervisory control, which is the focus of this paper, the process industry uses at present 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.

As indicated, in many implementations there is no formal separation between the regulatory and supervisory control layers, and in the process 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 the stabilizing (regulatory) and economic (supervisory) control tasks.

It is sometimes claimed that the vertical decomposition in Fig. 4 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 Fig. 4 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 vertical decomposition in Fig. 4 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).

2.5. 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

τ_{c1} (large) = closed-loop time constant of upper layer (outer loop)
τ_{c2} (small) = closed-loop time constant of lower layer (inner loop)

and define

$$\text{Time scale separation} = \tau_{c1}/\tau_{c2} \tag{3}$$

A large time scale separation is desired to allow for independent design of the layers (loops) and to avoid potential undesired interactions ("fighting") between them. Shinskey (1981) (page 12) recommends a time scale separation of at least 4, whereas Skogestad and 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) (page 310), such that performance degrades even nominally.

A large time scale separation also 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 τ_{c2} which 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 gain variations, it is often recommended to have a time scale separation of 10 (or larger). 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 (minimum) to 10 (preferable).

With a sufficient time scale separation the lower layer converges before the upper layer makes a new change (e.g. Chiang et al. (2007)). To understand the basis for the value of 4 in the rule of thumb, 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 to steady state is 98.2%, and convergence has for practical purposes been achieved.

Another justification for the lower value of 4, which is especially relevant for cascade control, follows by requiring that the interactions between the upper (slow) and lower (slow) control loops should not result in oscillations. Consider the series cascade control system in Fig. 10. For the linear case, all closed-loop transfer functions contain the "sensitivity" $S = (1 + L)^{-1}$ where $L = G_2C_2 + G_1G_2C_2C_1$. Assuming that both loops (layers) are approximated as first-order systems, we have in the Laplace (s) domain that $G_1C_1 = \frac{1}{\tau_{c1}s}$ and $G_2C_2 = \frac{1}{\tau_{c2}s}$. The closed-loop poles are found as the solution to $1 + L(s) = 0$, which gives 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 gives the requirement $\tau_{c1}/\tau_{c2} \geq 4$.

The limiting case of infinite time scale separation corresponds to $\epsilon = (\tau_{c1}/\tau_{c2})^{-1} \to 0$, 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.

2.6. II. Horizontal decomposition (distributed/decentralized control)

The second way of decomposing the control problem, is to divide each layer into separate blocks, each using only a subset of the input and output variables (see Fig. 4). The objective of the decomposition is usually to make it possible to use decentralized control with single-loop PID controllers. The most important decision is the input (MV) - output (CV) pairing, for which the two most important *pairing rules* are Minasidis et al. (2015):

- **"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 we also want 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 we can "give up" (stop controlling) when the MV saturates.[4]

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; Skogestad & Postlethwaite, 2005).

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.

In addition to decentralized PID 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).

2.7. What to control (CV1 and CV2)?

As seen from Fig. 4, 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.

2.7.1. Choice of economic controlled variables for supervisory control layer (CV1)

From an economic point of view, the following variables should be controlled (Skogestad, 2003)(Skogestad, 2015);

- **CV1=Active constraints**. Here "active" means that it is (economically) optimal to operate at this constraint. The setpoint is normally the constraint value. For hard constraints, one must add a "backoff" to avoid dynamic violation.
- **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).

There are usually many options for unconstrained self-optimizing degrees of freedom, and it is easy to make a bad choice.

- Bad choice for self-optimizing variable CV1: One should *never* control the cost J or any variable that reaches its maximum or minimum value at the optimum. Violation of this rule gives either

[4] The term "saturates" is used when a physical input (MV, u) reaches its minimum or maximum constraint value; for example, a closed (0) or fully open (1) valve position. A block diagram is shown in Fig. 20.

infeasibility (e.g., if attempting to control J at a lower setpoint than the minimum) or non-uniqueness (e.g., if attempting control J at a higher setpoint).

As an example, consider optimizing a marathon runner where we want to minimize the time, $J = T$ [s]. In this case, we should not control the speed CV1=v [m/s] at a constant setpoint, because it reaches a maximum value at the optimum. Also note that fixing v is indirectly the same as fixing the cost $J = T$ since $T = L/v$ where $L = 42195$ m is the length of the marathon. A good self-optimizing variable for a marathon runner may be the heart rate (Skogestad, 2004b).

- Ideal choice: The ideal self-optimization variable (CV1) 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 possibly a measurement combination (usually a linear combination, CV1 = Hw). The goal is that the optimal setpoint (CV1$_s$) is (almost) constant, that is, it depends 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 (find optimal matrix H) is the "nullspace method", but this only takes into account that the setpoint should be independent of disturbances. To take into account also the measurement error/noise (which effect is reduced if the gain from u to CV1 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).

2.7.2. Choice of controlled variables for regulatory control layer (CV2)

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. In addition, the measurement should have a small effective time delay and be robust. The choice of CV2 may not be critical economically because the setpoint CV2$_s$ is set by the layer above. Typical choices for the lower-layer controlled variables (CV2) in process control are levels, flows, pressures and temperatures. In mechanical systems, typical choices may be acceleration and rate/velocity/speed.

2.8. Active constraint switching

From an economic point of view, the control of the active constraints (CV1) is usually the most important. The reason is that there may be a large economic penalty imposed by having a "backoff" from the optimal constraint value. For this reason it is advisable to move the handling of active constraints down into the faster layers:

- If an active constraint needs to be tightly controlled (typically, for hard constraints) it is usually moved from the supervisory (CV1) to the regulatory control layer (CV2).
- The identification and switching between active constraints is usually handled by the supervisory layer and not by the optimization (RTO) layer.

The latter may seem surprising, because one may think that identifying active constraints requires optimization. Also, the number of possible active constraints regions (combinations) can be very large; up to 2^{n_c}, where n_c is the number of constraints (Jacobsen & Skogestad, 2011). In practice, especially when n_c is large, there are usually much fewer possible or relevant regions (Reyes-Lúa & Skogestad, 2020b). In any case, as discussed next, it is usually not necessary to identify possible constraints regions or use RTO, 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.

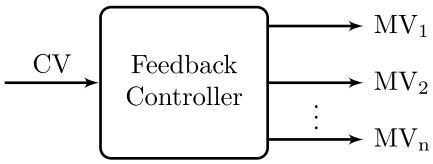


Fig. 5. MV-MV switching (input sequencing) is used when we have multiple MVs to control one CV, but only one MV should be used at a time. The block "feedback controller" usually consists of several elements, for example, a controller and a split range block.

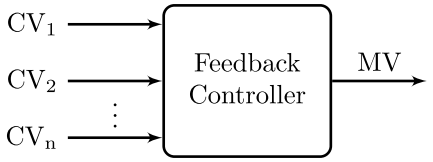


Fig. 6. CV-CV switching ("override") is used when we have one MV to control multiple CVs, but the MV should control only one CV at a time. The block "feedback controller" usually consists of several elements, typically several PID-controllers and a selector.

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 strategy. 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 choice and pairing of MVs and CVs. There is a fundamental difference between MV and CV constraints because we need a controller to handle a CV constraint, whereas an MV can simply be set at its optimal constraint value.

In turns out that we may distinguish between four different switching cases as described below: MV-MV, CV-CV, simple MV-CV and complex MV-CV switching (Reyes-Lúa & Skogestad, 2020b). Block diagrams for the two first cases are shown in Figs. 5 and 6, respectively. Note here that, the "Feedback controller" block may be a combination of simpler control elements (e.g., PID controller, selector and split range) and also note that setpoints (CV$_s$) have been omitted for simplicity. There is no separate figure for MV-CV constraint switching because one either does not need to do anything ("simple" MV-CV switching) or one needs to combine MV-MV and CV-CV switching to make a repairing of loops ("complex" MV-CV switching).

2.8.1. MV-MV switching (Fig. 5).

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 consider three alternative approaches (control elements):

1. Split range control with one controller (E5)
2. Separate controllers (with different setpoints) for each MV (E6)
3. Valve position control (E7)

2.8.2. CV-CV switching (Fig. 6).

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. It is also known as override or selective control. CV-CV switching is frequently used for satisfying inequality constraints.

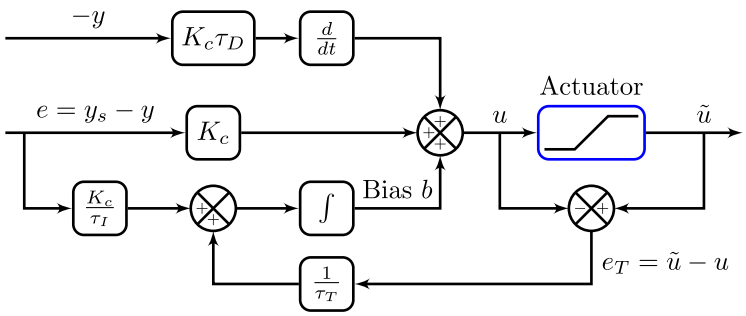


Fig. 7. Recommended PID-controller implementation with anti-windup using tracking of the actual controller output ($\tilde{u}$), and without D-action on the setpoint (Åström & Hägglund, 1988). The block "Actuator" does not need to be a saturation element, it could represent any element that may break the link between u and $\tilde{u}$, for example, a selector.
$\int$ = integral = $\frac{1}{s}$ in Laplace domain
$\frac{d}{dt}$ = derivative = s in Laplace domain
u = value computed by the controller.
$\tilde{u}$ = measured (or estimated) actual value applied to the process
K_c = controller gain (tuning parameter)
τ_I = integral time [s, min] (tuning parameter)
τ_D = derivative time [s, min] (tuning parameter)
τ_T = tracking time constant for anti-windup [s, min] (tuning parameter)
τ_F = filter time constant for measurement y [s, min] (tuning parameter) (not shown in the Figure).

When a CV constraint is encountered, one "gives up" (stops) controlling the present CV. CV-CV switching is implemented using selectors (E4).

2.8.3. 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.

2.8.4. 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 pairing rule", then it is not necessary to do anything (except for anti-windup).

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

This applies to the case where the CV that should be given up (when we encounter the MV constraint) is controlled with another MV. That is, we have paired an MV which may saturate (may reach a minimum or maximum constraint) 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 (E4, E5 or E6), and then we use CV-CV switching (a selector, E3) to give up the other CV.

We discuss in the next section these switches in detail.

3. Important advanced control elements

This section describes in more detail some of the "classical" or "standard" advanced control elements which, based on the author's experience, are widely used in the process industries (and in most other control application areas),

3.1. PID controller (E8)

The most important standard control element is the PID feedback controller, also see Appendix C. A recommended implementation of a PID controller with anti-windup (E8) using tracking is shown in Fig. 7 (Åström & Hägglund, 1988).

The most important for a PID-controller to work well is to have a good "pairing" between the MV (u) and the CV (y) (see the two main pairing rules in Section 2.6).

Next, it is important to choose good values for the PID tuning parameters (K_c, τ_I, τ_D), and rather than using "trial and error" online tuning, it is recommended to use a model-based tuning approach, such at the SIMC PID rules (Appendix C.3.1). In process control, derivative action if rarely used, and when it is used it is usually not applied to the setpoint (see Fig. 7). In addition, the designer may want to add a measurement filter (with tuneable filter time constant τ_F; see Appendix C.3.5) and for the anti-windup scheme in Fig. 7, the tracking time τ_T is also a tuning parameter (Appendix C.5.3). Importantly, the SIMC PID tuning rule has a single adjustable tuning parameter:

τ_c = desired closed-loop time constant [s, min] (4)

It is recommended to base the controller tuning (i.e., choice of K_c, τ_I and τ_D) on τ_c. First, it is systematic and the SIMC PID rules are simple to use and work well. Second, τ_c is needed for analyzing the time scale separation for cascade control and also for designing the measurement filter F (the rule is to select $\tau_F \leq \tau_c/2$; see Appendix C.3.5).

Choice of tuning parameter and squeeze and shift rule

For the SIMC PID rule in (C.13), the recommended value of the tuning parameter is

$\tau_c \geq \theta$ = effective time delay for process

For what loops do we need "tight" control with the smallest value $\tau_c = \theta$? The answer is that tight control is usually most important when the output y has a constraint which should not be violated dynamically. The extreme is a "hard"' constraint which never should be violated. In general, for output constraints where dynamic violations should be avoided, we need to introduce a "backoff" between the setpoint y_s and the constraint value, and by "tightening" control we may reduce the backoff and save money. This is illustrated in Fig. 8 and is known as the *Squeeze and shift rule: Use improved control to squeeze (reduce) the variance and shift 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 (e.g., Krishnamoorthy and Skogestad (2020)). The implications for controller

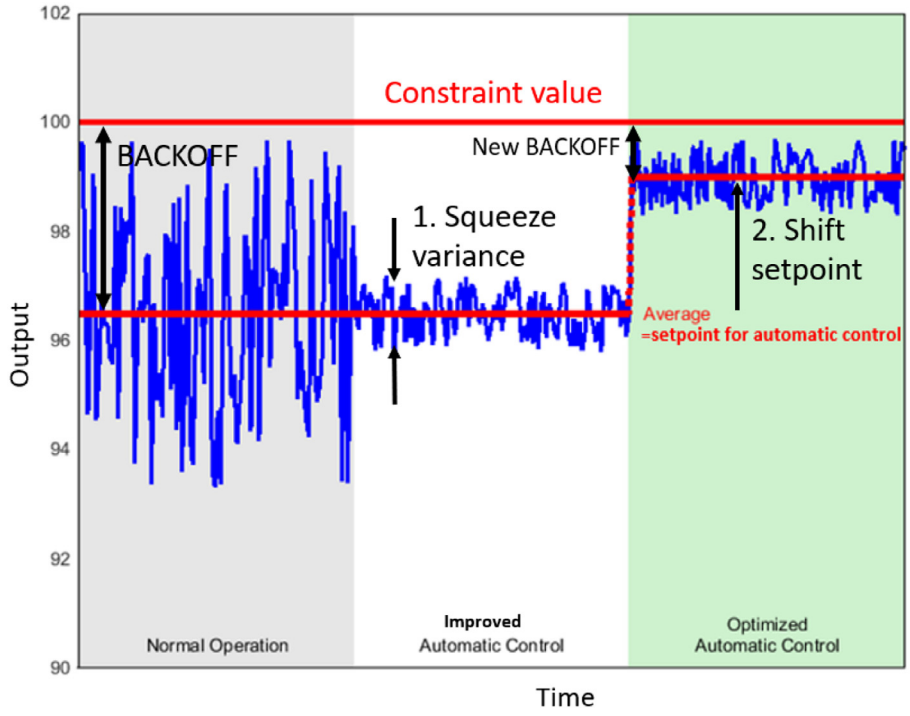


Fig. 8. Squeeze and shift rule: Squeeze the variance by improving control and shift the setpoint closer to the constraint (i.e., reduce the backoff) to optimize the economics (Richalet et al., 1978).

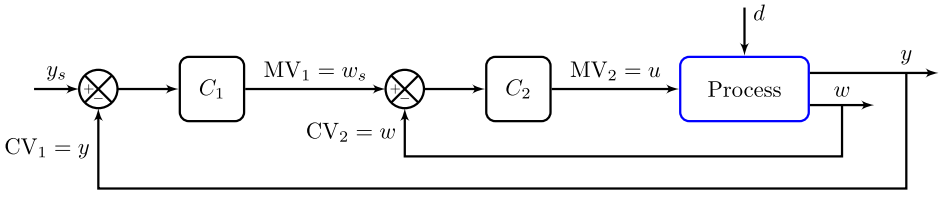


Fig. 9. General cascade control scheme with primary (outer, master) controller C_1 (slow) and secondary (inner, slave) controller C_2 (fast). All blocks are possibly nonlinear. The objective of the control system is to keep the output y close its setpoint y_s in spite of disturbances d. The extra (secondary) process measurement w (sometimes called y_2) is controlled on a fast time scale, with the objective of improving the control of y.

tuning is that it is important to have tight control (small τ_c) 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.

Note that we may also have "soft" output constraints where only the steady-state average matters. For such constraints, the integral action in the controller is sufficient and no backoff is needed. Of course, also for soft constraints we would like to improve control and reduce dynamic variations, because there are many other disadvantages with dynamic variations, including propagation of disturbances and equipment wear.

More details about the PID controller, including SIMC PID tuning, measurement filter, alternative anti windup schemes, bumpless transfer and on/off control are given in Appendix C.

3.2. Cascade control (E1)

A general cascade implementation is shown in Fig. 9. The outer (primary, master) controller C_1 has as its manipulated variable (MV_1) the setpoint (w_s) to the inner (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 main 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 case with a series process in Fig. 10. Here w is an intermediate (secondary) measurement which directly affects the primary output y through the primary process G_1.

An early and very good description of the benefits of cascade control is given by Shinskey (1967). With reference to Fig. 10, he writes (page 154) (this is a direct quote, except for the addition of symbols): "The principal advantages of cascade control are these:

1. Disturbances arising within the secondary loop (d_2) 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."

The third advantage is related to the important linearizing effect of "high-gain" feedback, which is usually not mentioned in control textbooks. Specifically, consider the inner loop in Fig. 10 with a feedback controller C_2 and process model G_2. For the linear case, the inner loop transfer function is $L_2 = G_2C_2$ and the closed-loop response from the

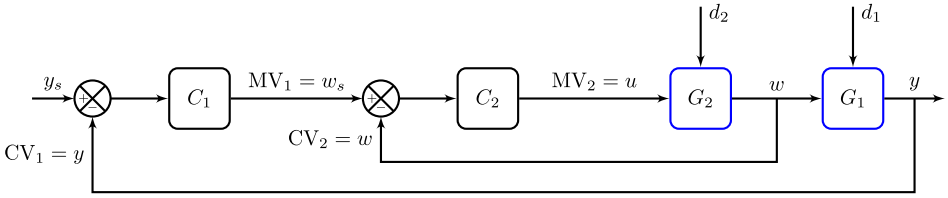


Fig. 10. Cascade control for series process where w (sometimes called y_2) is an intermediate process measurement. All blocks are possibly nonlinear.
C_1=primary/outer/master controller (slow), G_1 =primary process
C_2=secondary/inner/slave controller (fast). G_2 =secondary process.

setpoint w_s to the output w becomes $w = T_2 w_s$ with

$$T_2 = L_2(I + L_2)^{-1}$$

Without the inner loop, the process transfer function (from u to y) is G_1G_2. However, with the inner loop closed, the transformed process (from w_s to y) for tuning the outer controller C_1 becomes G_1T_2. With high-gain feedback in C_2, we get $\|L_2\| \gg 1$ and we have $T_2 \approx I$ (perfect linear response), or equivalently $w \approx w_s$, independent of the model G_2. Thus, we have the (seemingly incredible) fact that the response is independent of the model G_2, so it does not matter if G_2 varies, for example, due to nonlinearity. A typical example is when G_2 is a valve with a nonlinear gain characteristic, u is the valve position and w is the flow measurement. However, it should be noted that gain variations in G_2 translate into changes in the dynamics (response time) in T_2. This illustrated in Appendix B.2 where we find that a process gain increase of 50% translate into a corresponding reduction in the closed-loop time constant τ_{c2} (from 4 s to 4/1.5=2.67 s for the specific example).

A potential problem with high-gain feedback is that it may result in instability, but according to the Bode stability condition this is solved by using integral action (which gives infinite controller gain at low frequency) and reducing the controller gain at higher frequencies (e.g., Skogestad and Postlethwaite (2005), page 24), for example, by using a PI-controller or even a pure I-controller, $C_2(s) = K_I/s$.

Tuning of the two controllers C_1 and C_2 should be done sequentially, and it is 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 tuning parameters. The faster inner (secondary) controller C_2 (with a "small" τ_{c2}) is tuned first based on the process model G_2, and with this loop closed, we tune the slower outer (primary) controller C_1 (with a "large" τ_{c1}). For the case with a series process (Fig. 10), the tuning of C_1 may be done based on the process model G_1 with an added effective delay $\tau_{c2} + \theta_2$ to represent the inner loop (that is, the inner loop closed-loop transfer function is approximated as $T_2 \approx 1$ plus an effective delay). Here θ_2 is the effective delay in G_2 and τ_{c2} is the closed-loop time constant for the controller C_2.

As given by the rule of thumb in Section 2.5, 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.

It is possible to extend with more layers of cascade control. For example, if we want to control composition in a distillation column using reflux flow, then we typically have a cascade with three layers: A slow composition controller (CC) sends a temperature setpoint to a temperature controller (TC) which again sends a flow setpoint to a flow controller (FC) which finally manipulates the physical valve position. We may even have a fourth layer, because the composition setpoint may be set by a downstream units which produces the final product. In principle, this works well, but the problem is that it may "eat up" the available time window. For example, with four layers of cascade (C_1, C_2, C_3, C_4) and a time scale separation of 10 between each layer, the slowest outer control loop (C_1) will be $10^3 = 1000$ times slower than the fastest inner loop (C_4). If the fastest loop (C_4) is a flow controller with a closed-loop time constant of 10 s, then the slowest outer loop (C_1) will have a closed-loop time constant of 10^4 s = 2.7 h. To avoid eating up the time window (for example, if 2.7 h is too slow for the outer loop with C_4), the solution is either to reduce the time scale separation or to tune the inner loop more tightly (i.e., with a small τ_{c2}).

If it is not possible to achieve the desired time scale separation of about 4 or larger, then it is still possible to use cascade control (where the outer controller sets the setpoint to inner controller), but the above four advantages of cascade control are then lost, at least to some degree. Specifically, if we tune the controllers C_1 and C_2 sequentially (e.g., C_1 is tuned based on the model G_1T_2), then it is even possible to have a time scale separation less than 1, that is, the outer loop is the fastest. However, this is not recommended because the tuning of C_1 is then no longer independent of the tuning of C_2, and if there are gain variations in $L_2 = G_2C_2$ for the inner loop, then these will affect the output y.

An alternative to cascade control for cases where it is not possible to get a sufficiently large time scale separation, is to design a two-input (y and w) single-output (u) controller, for example, using standard optimal control (e.g., using state feedback with LQG control or MPC). In the linear case, this may give a controller of the form

$$u = C_1'(s)(y_s - y) - C_2(s)w \tag{5}$$

where C_1' and C_2 are designed simultaneously. Note that with cascade control (Figs. 9 and 10) we have $C_1' = C_2C_1$, where we first design C_2 and then C_1, but this sequential design is a good approach only if the time scale separation is sufficiently large. Also note that with cascade control, both C_1 and C_2 usually have integral action, whereas in the more general case in (5) usually only C_1' has integral action.

3.3. Ratio control (E2)

Ratio control involves keeping a constant ratio R, either between a manipulated variable u and a disturbance d (to be used for feedforward control),

$$R = u/d \tag{6}$$

or between two manipulated variables (to be used for decoupling control),

$$R = u_1/u_2 \tag{7}$$

3.3.1. Implementation with multiplication element

A typical ratio control scheme for a mixing process is shown in the flowsheet[5] in Fig. 11. Based on physical insight, the viscosity (y) of the product will be constant if we keep a constant ratio $R = F_2/F_1$ between the flowrates of water ($u = F_2$) and solids ($d = F_1$). To implement this, we measure the solid flowrate $d = F_1$ (a disturbance, sometimes called

[5] In a flowsheet, a controller is written as XC where X tells what kind of variable is being controlled, for example, FC for flow control, PC for pressure control, TC for temperature control, LC for level control, IC for inventory control (which usually is level or pressure) and VC for viscosity control. These are typically single-variable PID controllers.

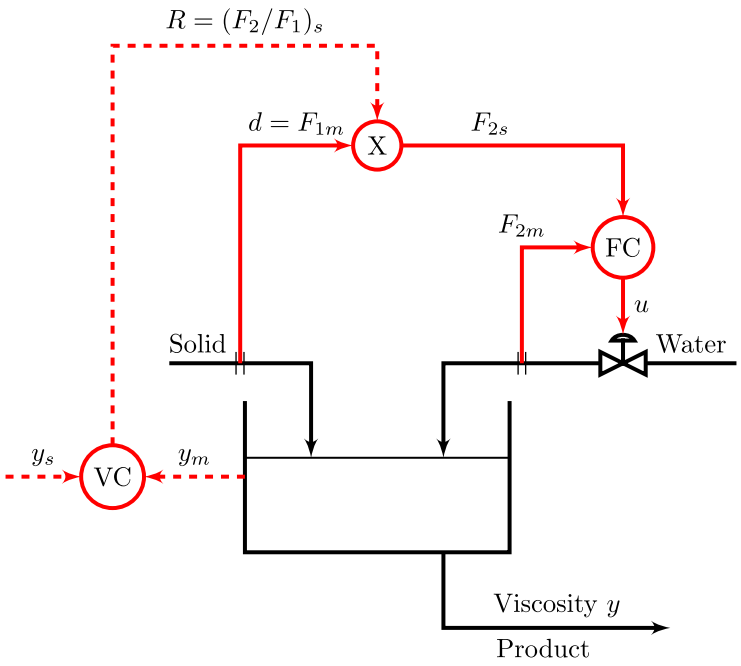


Fig. 11. Flowsheet of ratio control with feedback correction (trim).
The ratio control is shown with red solid lines. The ratio block (x) multiplies the measured flow disturbance $d = F_1$ with the desired flow ratio R to get the flow MV=F_{2s}. An inner flow controller (FC) with $u = z$ (valve position = physical input) and $w = F_2$ is used to implement the desired flowrate $w_s = F_{2s}$. The outer feedback viscosity controller VC (red dashed lines) corrects the ratio setpoint $R = (F_2/F_1)_s$ in order to get $y = y_s$ at steady state.
To satisfy the steady-state mass balance, the product outflow should be set by a level controller (not shown on the flowsheet).

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 Fig. 11 this is done by 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.

In Fig. 11 the controlled variable (y) is viscosity, but for the use of ratio control it does not matter what the controlled property variable is; it could be concentration, density, boiling point, pH, color and so on. The reason for choosing viscosity was to illustrate this point, namely that it works also for a property where the blending model may be nonlinear or even unknown. This is different from conventional feedforward control (and MPC) where a blending model is required. Importantly, with ratio control, the use of feedback trim (discussed next) based on measuring the property variable y eliminates the need for a blending model.

3.3.2. Feedback trim for ratio

In Fig. 11 we also have included a feedback adjustment (trim) of the ratio R. We use an outer viscosity controller (VC) which finds by feedback ("trial and error") the correct ratio R which makes the measured viscosity y equal to its setpoint y_s. 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 a measured mixture property such as taste, color, texture, turbidity (haziness) or "thickness" (viscosity). In summary, the use of a feedback correction ("feedback trim") is very powerful and common as *it replaces the need for a model for how y depends on the inputs and disturbances.*

3.3.3. Theoretical basis for ratio control

Ratio control is most likely the oldest control approach (think of recipes for making food), but despite this, no theoretical basis for ratio control has been available until recently (Skogestad, 2023). Importantly, 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 temperature or pressure. On the other hand, the two variables included in the ratio R are implicitly assumed to be *extensive variables*.

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 extensive variables (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.

However, the scaling assumption may not hold so there are also some restrictions with ratio control:

1. The scaling property does not hold for many process units. For example, heat exchangers (where the heat transfer depends on heat exchange area which is usually constant) and non-equilibrium reactors (where the conversion depends on reactor volume which is usually kept constant). 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.
2. The scaling property requires that *all extensive variables* (flows, heat rates, sizes of certain equipment) must be scaled by the same factor in order to keep the intensive variables (including

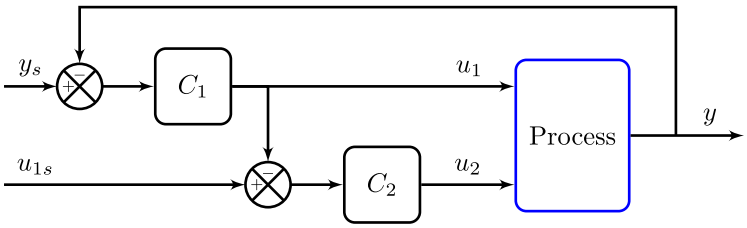


Fig. 12. Valve (input) position control (VPC) for the case when an "extra" MV (u_1) is used to improve the dynamic response. A typical example is when u_1 is a small fast valve and u_2 is a large slower valve.
C_1 = fast controller for y using u_1.
C_2 = slow valve position controller for u_1 using u_2 (always operating).
u_{1s} = steady-state resting value for u_1 (typically in mid range. e.g. 50%).

y) constant. Thus, ratio control should not be used if we have saturation in a flow (even if this is a process unit where the scaling property holds), because then it is not possible to scale (change) all the extensive variables by the same amount.

3. To have perfect ratio control, we must require that all independent intensive variables (e.g. feed composition and temperature) are kept constant. However, this is not a critical requirement if we have an outer feedback trim which adjusts the ratio R.

For a distillation column, the third restriction implies that the scaling assumption holds if we assume that the pressure and stage efficiency (number of theoretical stages) is constant, but this can be overcome with an outer feedback trim which adjusts the ratios. However, the second restriction may be more serious, for example, we should not apply ratio control to a distillation column with a fixed heat input. 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 conventional feedforward control and decoupling (and more generally for any model-based scheme, including MPC).

3.3.4. 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 processes. Since ratio control is difficult to implement with MPC (see Section 7.6.6), it should always be included in the regulatory layer, and having the ratio as a manipulated variable for MPC (MV=R).

3.4. Input (valve) position control (VPC) to improve the dynamic response (E3)

3.4.1. VPC with two MVs (for mid-ranging control)

Consider a "multi-input single-output" (MISO) process with two MVs (inputs; u_1 and u_2) and one CV (y), but only one MV (u_2) is used for steady-state control. The other MV (u_1) is an "extra" input (for example, u_1 is a bypass stream or a small valve in parallel to the main valve u_2) 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}. This may be realized using input (valve) position control (VPC) as shown in Fig. 12. The term "valve" position control (VPC) is common, although a better name would be "input resetting control" because the extra MV (u_1) does not need to be a valve; it may even be the setpoint to another loop. Another common term is mid-ranging control (Allison & Isaksson, 1998; Åström & Hägglund, 2006). Hägglund (2021) provides a review of alternative solutions for mid-ranging control.

In Fig. 12, 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

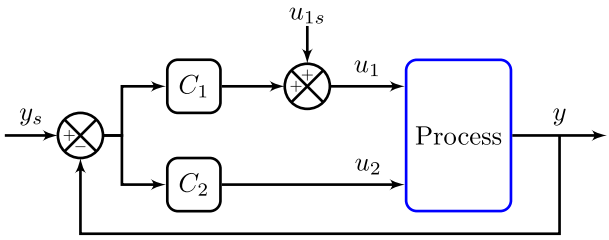


Fig. 13. Parallel control to improve dynamic response – as an alternative to the VPC solution in Fig. 12.
The "extra" MV (u_1) is used to improve the dynamic response, but at steady-state it is reset to u_{1s}. The loop with C_2 has more integral action and wins a steady state.

should typically be in the order 4 to 10. Allison and 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) note 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.

3.4.2. Parallel control: Alternative to VPC for improving the dynamic response

An alternative solution to VPC is to use two-input single-output (TISO) control (also known as "parallel control") (Fig. 13) where both C_1 and C_2 control the same y.

$$u_1 = C_1(y_s - y) + u_{1s}; \quad u_2 = C_2(y_s - y)$$

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, so that u_2 will change to make $(y_s - y) = 0$. Usually, this means that C_2 is a PID-controller and C_1 is a P- or PD-controller in Fig. 13.

The advantage with valve position control compared to parallel control is that the two controllers in Fig. 12 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.

Hägglund (2021) presents an alternative parallel scheme with "feedforward" action to coordinate the manipulated variables u_1 and u_2, for example, for cases with stiction for the main input u_2.

3.4.3. VPC with one MV (stabilizing control with resetting of MV)

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 (Fig. 14) (Storkaas & Skogestad, 2004). The inner fast controller (C_2)

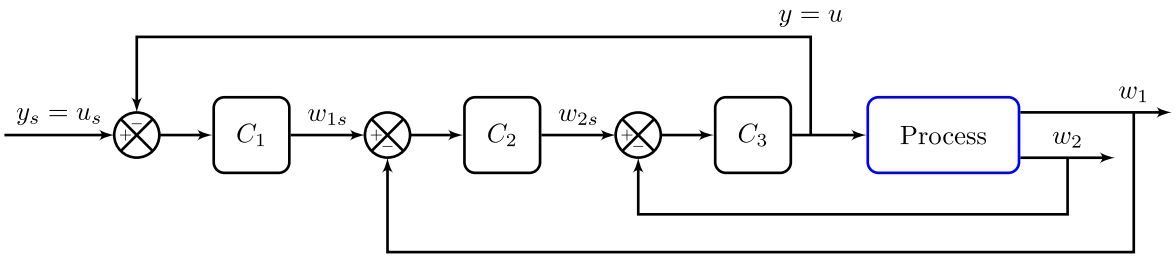


Fig. 14. Stabilizing control of variable w_1 combined with valve position control (VPC) for u (=valve position) and inner flow controller ($w_2 = F$).
It corresponds to the flowsheet in Fig. 15 with $w_1 = p$ (pressure), C_1 = outer VPC (slow), C_2 = stabilizing controller (fast), C_3 = inner flow controller (very fast).
Note that the process variables (w_1, w_2) have no fixed setpoint, so they are "floating".

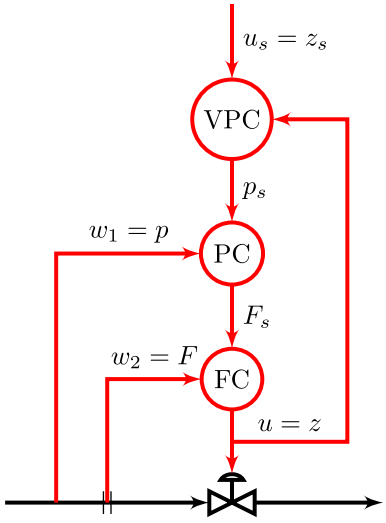


Fig. 15. Anti-slug control where the pressure controller (PC) is used to stabilize a desired non-slugging flow regime. The inner flow controller (FC) (fast) provides linearization and disturbance rejection. The outer valve position controller (VPC) (slow) resets the valve position to its desired steady-state setpoint ($u_s = z_s$). It corresponds to the block diagram in Fig. 14.

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 Fig. 14 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 $w_{1s} = p_s$. Applications of "floating pressure" operation are found in steam systems and distillation columns (Shinskey, 1979; Wade, 2004). Another application is discussed next.

Example VPC with one MV: Anti-slug control

An application for stabilizing multiphase flow (Storkaas & Skogestad, 2004) is shown in the flowsheet in Fig. 15. It corresponds to the block diagram in Fig. 14. 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 s (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), which will limit the achievable speed (bandwidth) for resetting the valve to its desired position $u_s = z_s$.

A common related example is stabilizing a bicycle. Here, u is the vertical position (tilt) of the body, w_1 is the vertical position (tilt) of the bicycle, and there is no variable w_2 (Storkaas & Skogestad, 2004).

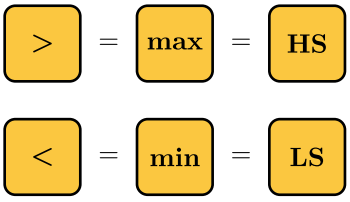


Fig. 16. Alternative symbols for selector block. Each selector block has two or more inputs, but only one output. HS= high select, LS = low select.

3.5. Selective (limit) control (E4)

Selectors are used for CV-CV switching (Fig. 6), which is when one MV (u) is used to control many CVs ($y_1, y_2, \ldots$), but only one CV should be controlled at a time. Some alternative symbols for selectors are shown in Fig. 16. 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.

CV-CV switching is sometimes called "override" control, but this term may be misleading because it gives the impression that it is some ad-hoc industrial method where we make a "fix" to the solution. On the contrary, as discussed in Section 7.5, the result from CV-CV switching ("override") is usually optimal, at least at steady state.

[6] Anti-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. Stabilizing laminar flow may still be possible, for example, with distributed actuators that manipulate locally the diameter of a flexible pipeline.

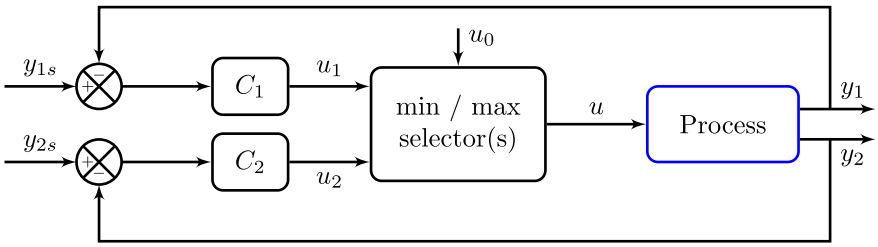


Fig. 17. CV-CV switching with selector on MV (input u).
Here, y_{1s} and y_{2s} may be constraint values or desired setpoints, whereas u_0 (if used) may be a desired value which may be given up. The block "min/max selector(s)" may be a max- or a min-selector (Rule 1), or a max- and min-selector in series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

3.5.1. Selector on input (MV)

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 Fig. 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.

Note in Fig. 17 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 and Middleton (1999). Rather, in CV-CV switching we have multiple controllers which are working one at a time.

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

Selector Rule 1. Max or Min selector *(applies to selector on MV, see Fig. 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 Fig. 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. However, if the constraints require both a max and min selector, we have to be more careful:

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 selector block.*

For example, in Fig. 18 we use a max-selector followed by a min-selector, $u = \min(u_2, \max(u_0, u_1))$, since 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.

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 the same 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 Fig. 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 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 a model-based solution (MPC) may be simpler.

3.5.2. Selector on setpoint (cascade)

An alternative (and somewhat less general) implementation of CV-CV switching is the cascade implementation with the selector on the *setpoint*, as shown in Fig. 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. The reason why the cascade implementation is said to be "somewhat less general" is because the design of the outer controller depends 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 5.3) where the cascade implementation is *not* recommended.

If the setpoint y_{2s} to the inner loop is a constant (for example, a constraint), then it may be convenient to replace the selector block in Fig. 19 by a saturation element (limiter) (Cao, 2004).

3.5.3. Auctioneering selector

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, if we want limit the maximum temperature ("hotspot") in a reactor, then we may use a single controller with $y = \max(y_1, y_2, \dots)$ and $y_s = y_{max}$. This solution is sometimes referred to as *auctioneering* (Shinskey, 1967).

3.5.4. Built-in selectors in final control elements (actuators)

All physical inputs are generated by final control elements (actuators) such as valves, pumps and motors, and they have a maximum and minimum value (constraint) which cannot be violated. This may be represented by a saturation element (limiter) with a max- and min-value as shown in Fig. 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 (Fig. 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 Fig. 20 is equivalent to a max- and min-selector in series (in any order) or to a mid-selector:

$$\tilde{u} = \max(u_{min}, \min(u_{max}, u)) = \min(u_{max}, \max(u_{min}, u)) = \text{mid}(u_{min}, u, u_{max}) \tag{8}$$

The order of the "built-in" max- and min-selector in (8) 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

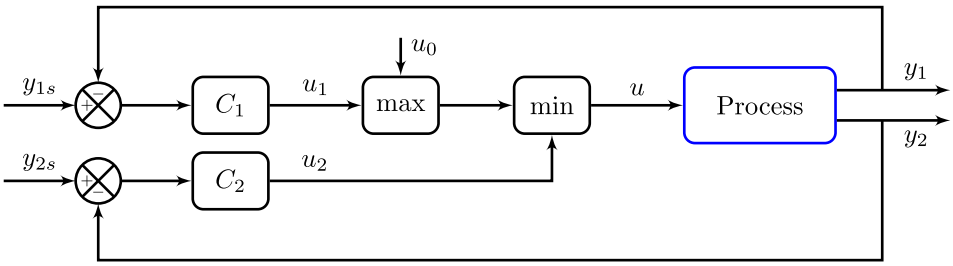


Fig. 18. CV-CV switching for case with possibly conflicting constraints. In this case, constraint y_{1s} requires a max-selector and constraint y_{2s} requires a min-selector. The selector block corresponding to the most important constraint (here y_{2s}) should be at the end (Rule 2).
To understand 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 (Rule 3) at the end because of the possible saturation of u, so if the MV (input) saturates, then all variables (u_0, y_{1s}, y_{2s}) will be given up.

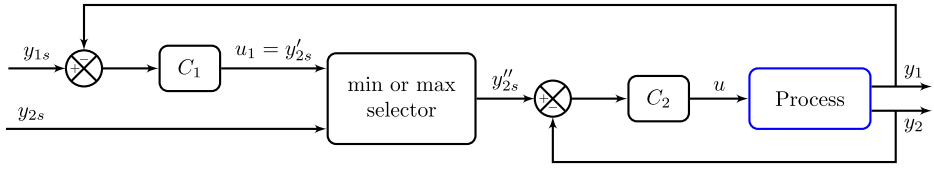


Fig. 19. Alternative cascade CV-CV switching implementation with selector on the setpoint. In many cases, y_{1s} and y_{2s} are constraint limits.

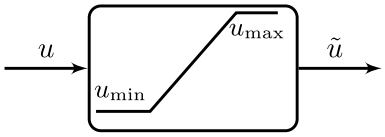


Fig. 20. Saturation element (limiter) to represent amplitude limits (constraints), for example, for a valve. It is equivalent to a min- and a max-selector in series or to a mid-selector, see (8).

(valve) always comes 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 a saturation element in the block diagram for such cases, e.g. see Fig. 21.

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 and Hägglund (2006).

3.6. 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 to cover the entire region of steady-state operation, 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 (Fig. 5), for which the oldest approach is split-range control (Eckman, 1945) as shown in Fig. 21.[7] An example is when 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 Fig. 21. In Fig. 21, 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.

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

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 (Fig. 22) (Smith, 2010) (Reyes-Lúa & Skogestad, 2019).

The setpoints $(y_{s1}, y_{s2}, \dots)$ should in the same order as we want to use the MVs. The setpoint differences (e.g., $\Delta y_s = y_{s2} - y_{s1}$ in Fig. 22) should be large enough so that, in spite of disturbances and measurement noise for y, only one controller (and its associated MV) is active at a given time (with the other MVs at their relevant limits). The solution in Fig. 22 has two important advantages compared to split range control in Fig. 21. First, the controllers $(C_1, C_2, \dots)$ can be designed independently (.g., with different integral and derivative times) for each MV, whereas in split range control there is a single controller C. Second, and probably more importantly, one avoids in Fig. 22 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 Fig. 21. Instead, any saturation limit (or similar) is detected indirectly by feedback through the loss

[7] Note the blue saturation elements for the inputs in Fig. 21 and other block diagrams. Saturation can occur for any physical input, but they are explicitly shown for cases where the saturation is either the reason for or part of the control logic. For example, in Fig. 21, the reason for using u_2 is that u_1 may saturate.

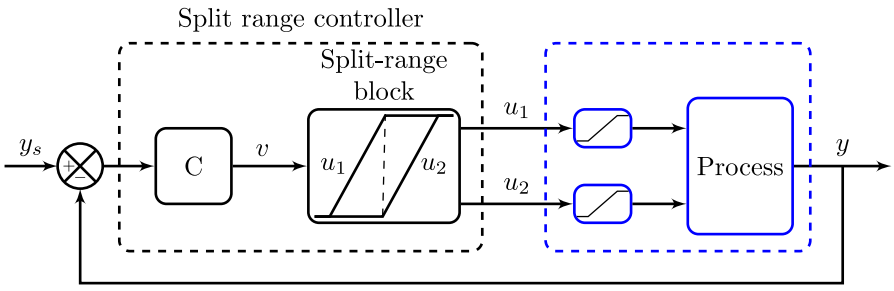


Fig. 21. Split range control for MV-MV switching.
A typical example is when u_1 are u_2 are two sources of heating and y is temperature.
In some cases there is a small overlap where both u_1 and u_2 are used simultaneously, for example, to correct for valve nonlinearity.

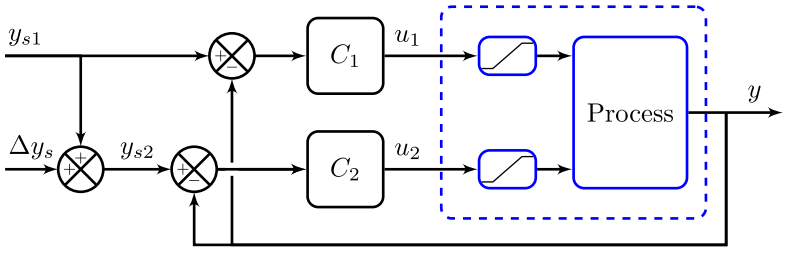


Fig. 22. Separate controllers with different setpoints for MV-MV switching.

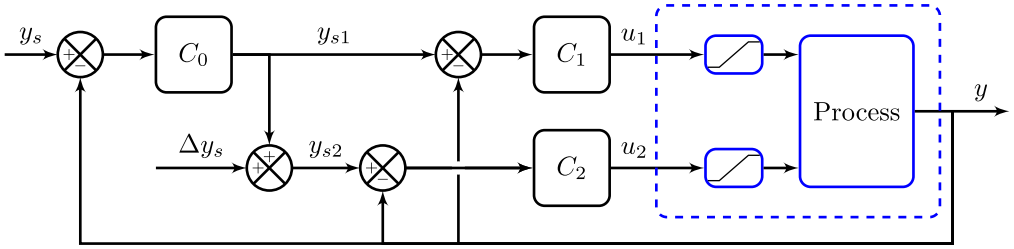


Fig. 23. Separate controllers for MV-MV switching with outer resetting of setpoint.
This is an extension of the scheme in Fig. 22, with a slower outer controller C_0 that resets y_{1s} to keep a fixed setpoint $y = y_s$ at steady state.

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 Fig. 22 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 temporarily 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 $y = y_{s2} = y_{s1} + \Delta y_s$ when we use u_2. The last disadvantage can be avoided (at least at steady state) by using the implementation in Fig. 23. 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, $y_{s1} = 21$C) in the winter than in the summer (say, $y_{s2} = 23$C) to save energy (and money) for heating and cooling (Reyes-Lúa & Skogestad, 2019).

3.8. VPC for MV-MV switching (E7)

Consider yet again MV-MV switching, and assume that for dynamic reasons we 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 u_1 saturating. This can be realized using valve position control as shown in Fig. 24.

The main advantage with the VPC scheme (Fig. 24) compared to the two alternative schemes for MV-MV switching (split range control in Fig. 21 and multiple controllers in Fig. 23) is that the same input (u_1) is always used to control y. The disadvantage is that when u_2 is used, we need to keep using 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 (Fig. 24) requires that we use a little heating also when we need cooling.

The VPC solution for MV-MV switching (Fig. 24) 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 Figs. 12 and 24 seem to be the same, but actually their behavior is very different. In Fig. 12 (VPC for improved dynamic control) we expect no

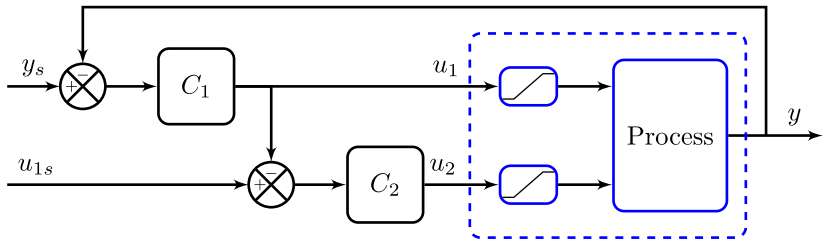


Fig. 24. Valve (input) position control for MV-MV switching. A typical example is when u_2 is needed only in fairly rare cases to avoid that u_1 saturates.
u_{1s} = value of u_1 where we switch to using u_2 (usually close to constraint, e.g., at 10% or 90%).
C_2 = valve position controller (only operating when u_1 reaches u_{1s}; otherwise u_2 is at its constraint, typically $u_2 = u_{2,min} = 0$.).
The VPC schemes in Figs. 12 and 24 seem to be the same, but their behavior is very different. In Fig. 12 both inputs are used all the time (u_2 is the main steady-state input, and u_1 is used to improve dynamics), whereas in Figs. 24, u_1 is the main input and u_2 is only used when u_1 approaches saturation.

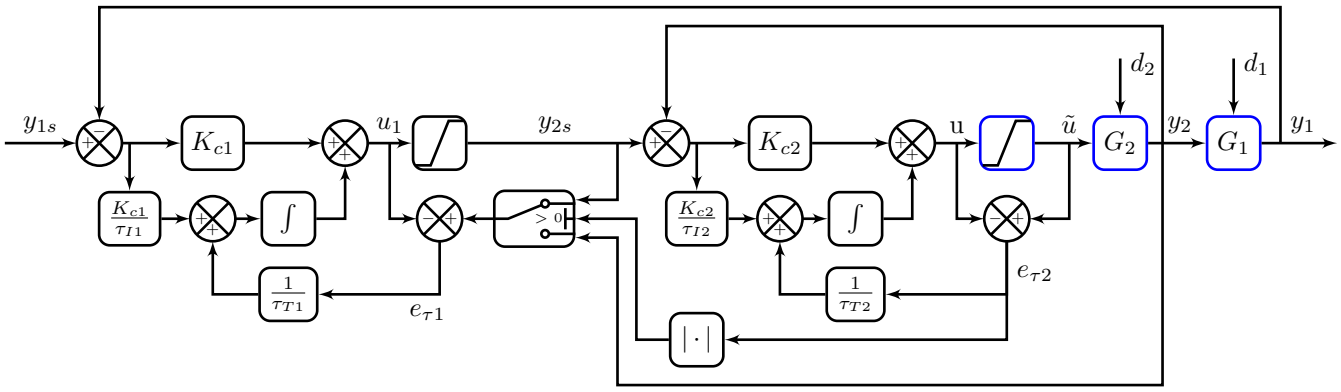


Fig. 25. Cascade control with anti windup using tracking, using an industrial switching approach (Leal et al., 2021).

saturation of the inputs u_1 and u_2. On the other hand, in Fig. 24 (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., $u_1 = u_{1s} = 10\%$).

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. A valve positioner is usually a high-gain P-controller which ensures that the actual measured valve position ($w = z$) is equal to the desired valve position.

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

In this paper, we recommend anti-windup with back-tracking as given in Fig. 7 and (C.18). In general, anti-windup needs to be implemented for controllers with integral action for cases where the MV (= controller output = u in Fig. 7) 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 Fig. 7 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 (Figs. 9 and 10), the question is how we should apply anti windup in the outer loop (C_1). Saturation for $\text{MV}_2 = u$ in the inner loop will give an offset for $\text{MV}_1 = w_s$ in the outer loop, which will result in "wind up" of the integrator for C_1.

To avoid this, one may use the "industrial switching approach" (Leal et al., 2021) in Fig. 25 (note that we in this figure have written $y_2 = w$ and $y_1 = y$). The switch between y_2 and y_{2s} when computing e_{T1} avoids that the anti-windup for C_1 corrects for the expected "normal" dynamic control error $y_{2s} - y_2$ when there is no saturation in u.

3.10. Linear feedforward control (E11)

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.

Consider 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 Fig. A.42 becomes

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

The Laplace variable s is included here to show that the feedforward controller is generally dynamic. There are two main problems here. The first is that $C_{Fd,ideal}(s)$ may not be realizable, for example, if the time delay in G_u is larger than in G_d. However, this problem is relatively easy to avoid by obtaining a realizable approximation (e.g. Guzmán and Hägglund (2021)). The simplest approximation is to use a constant gain (static feedforward compensator), that is, $C_{Fd}(s) = K_{Fd} = C_{Fd,ideal}(0)$.

A second, and more fundamental problem is that the model may be wrong, and feedforward control is generally sensitive to model errors, also at steady state. Specifically, as proved next, if the gain in G_u increases by more than a factor 2, then the resulting input u will be too large so that the output y overshoots more (in magnitude) in the opposite direction than with no control ($u = 0$), making feedforward control worse than no control.

Proof of sensitivity of feedforward control 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

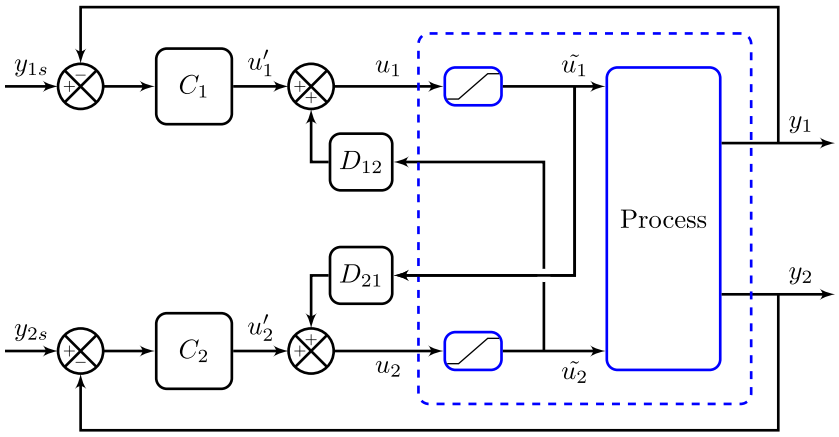


Fig. 26. Linear decoupling with feedback (reverse) implementation of Shinskey (1979).

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 $G_u' = 2G_u(\alpha_u = 2)$ and $G_d' = G_d$ ($\alpha_d = 1$) we get $y = -G_d d$ (with feedforward) which is identical in magnitude (but in the opposite direction) to no control ($y = G_d d$). With $\alpha_u = 2.5$ it is 50% worse in magnitude (again in the opposite direction) than with no control. In another example, let $G_u' = 1.5G_u(\alpha_u = 1.5)$ 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 compared to no control ($y = G_d' d$).

To reduce the potential "overshooting" in the opposite direction with feedforward control, one may introduce a "chicken factor" f and choose for example $C_{Fd} = f \cdot C_{Fd,ideal}$, 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 4) to avoid changes in the linear model caused by nonlinearity.

Finally, it should be noted that the design of feedforward and feedback control may require coordination to avoid that they "fight" against each other (Guzmán & Hägglund, 2011). Model predictive control may provide a good solution for more complex cases.

3.11. Linear decoupling (E12)

The feedforward idea can also be applied to decoupling as illustrated in Fig. 26. For the 2 × 2 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 Fig. 26 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 of G. 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 Fig. 26 uses the feedback decoupling scheme of Shinskey (1979) which is called inverted decoupling (Wade, 1997). Compared with 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 Fig. 26 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. In other words, the tuning of C_1 and C_2 can be based on the open loop models (G_{11} and G_{22}).
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 the stability 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 important, one may select $D_{21} = 0$.

The scheme in Fig. 26 can easily be extended to 3 × 3 systems and higher. Again one may simplify by 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 then special case of nonlinear feedforward and decoupling as discussed next.

4. Nonlinear feedforward, decoupling and linearization (E14)

A fairly general control structure with combined feedforward and feedback control is shown in Fig. 27. 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 inverting the process model. Many control schemes can be rewritten in this form, for example Internal Model Control (IMC), MPC (where the block "feedback controller" is actually the estimator), feedback linearization and the use of transformed inputs v (Skogestad et al., 2023).

In this paper, we consider the use of transformed inputs v, where the "feedforward block" is static and nonlinear and may include decoupling and linearization.

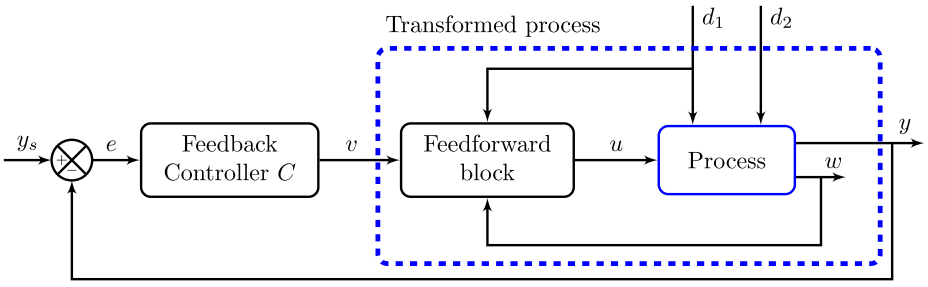


Fig. 27. Block diagram of control system with combined feedforward (often nonlinear) and feedback control (often linear). The outer feedback controller C uses the "transformed input" v to provide a feedback correction to the feedforward part.
Comment: The figure shows the possibility of treating a process measurement w in a feedfirward manner (like a measured disturbance), although strictly speaking this introduces feedback. Typically, w is a flow measurement. The main idea is that the Feedforward block is based on model inversion; e.g., see Fig. 29.

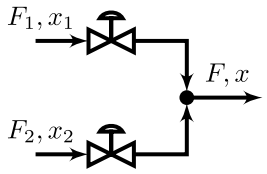


Fig. 28. Flowsheet of in-line blending process (mixer) where F is the flowrate [kg/s] and x is the mass fraction of component A [kg A/kg].

4.1. Introductory example: Blending 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 $y_1 = x$ (mass fraction of A) and total flow $y_2 = F$ [m^3/s]; see Fig. 28. For example, A could be water and B could be methanol, that is, we have $x_1 = 1$ (pure A in stream 1) and $x_2 = 0$ (no A in stream 2). An equivalent process from a control point of view, would be a shower process where we mix hot (1) and cold (2) water and want to control temperature and flowrate. The inputs, outputs and disturbances for the process are

$$u = \begin{pmatrix} F_1 \\ F_2 \end{pmatrix}; \quad y = \begin{pmatrix} x \\ F \end{pmatrix}; \quad d = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \tag{9}$$

This is a coupled process and if we want to use single-loop control it may be difficult to decide on good pairings between the manipulated variables u and controlled variables y. However, based on physical insight (or a steady state model), with $x_1 = 1$ and $x_2 = 0$, the system becomes decoupled if we use as "transformed" manipulated variables the flow ratio and the sum (McAvoy, 1983) (page 136),

$$v_1 = \frac{F_1}{F_1 + F_2} \tag{10a}$$

$$v_2 = F_1 + F_2 \tag{10b}$$

Note here that $v_1 = x$ and $v_2 = F$. The resulting model from transformed inputs to outputs then becomes extremely simple:

$$y_1 = v_1 \tag{11a}$$

$$y_2 = v_2 \tag{11b}$$

Based on (11), Seborg et al. (2016) (page 343) write about the choice of transformed manipulated variables in (10): "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 not correct. If we assume that the disturbances are measured, then it is always possible to derive ideal transformed manipulated variables (inputs) 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).

4.2. Ideal transformed inputs

Consider the steady-state model

$$y = f_0(u, d) \tag{12}$$

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

$$v_0 = f_0(u, d) \tag{13}$$

For implementation, one needs to invert the model by solving (13) with respect to u for given values of v_0 and d. We can formally write the solution as

$$u = f_0^{-1}(v_0, d) \tag{14}$$

At steady state, the resulting transformed system then trivially becomes

$$y = v_0 \tag{15}$$

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 Fig. 29. Here we have allowed for treating some measured states w as disturbances because this may allow for simpler models (Skogestad et al., 2023).

The method in (14) and Fig. 29 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).

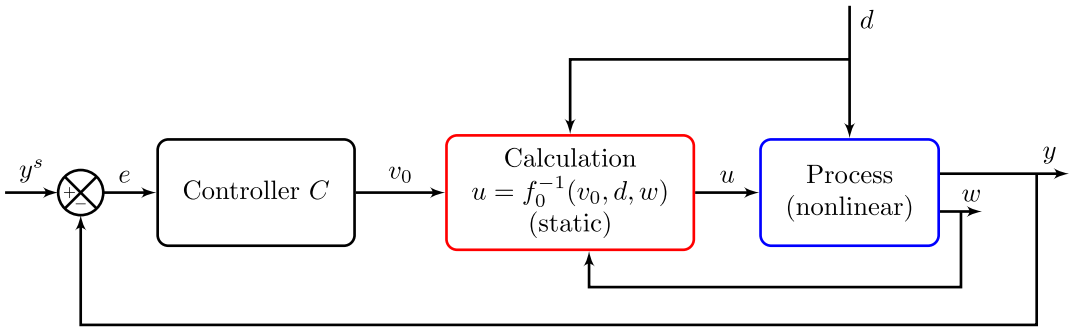


Fig. 29. Feedforward, decoupling and linearization (red calculation block) using transformed inputs $v_0 = f_0(u, d, w)$ based on static model $y = f_0(u, d, w)$. In the ideal case with no model error, the transformed system from v_0 to y (as seen from the controller C) becomes $y = I v_0$ at steady state.
d = measured disturbance
w = measured process state variable.

4.3. Example: Ideal transformed inputs for blending process

Consider again the blending process in Fig. 28 where x_1 and x_2 represents the mass fraction of A in the two feed streams. In Section 4.1, we assumed $x_1 = 1$ and $x_2 = 0$, but we here remove this restriction. We want to blend feed 1 (with flowrate $u_1 = F_1$ and composition $d_1 = x_1$) with feed 2 ($u_2 = F_2, d_2 = x_2$) to make a product with composition $y_1 = x$ [kg A/kg] and total flow $y_2 = F$ [kg/s]. The steady-state model (component mass balance for A and total mass balance) gives

$$x = \underbrace{(F_1 x_1 + F_2 x_2)/(F_1 + F_2)}_{v_{0,1}} \tag{16a}$$

$$F = \underbrace{F_1 + F_2}_{v_{0,2}} \tag{16b}$$

Note that the same model applies also for mixing of hot and cold feeds with constant heat capacity but then x is temperature.

The two ideal transformed inputs, $v_{0,1}$ and $v_{0,2}$, are simply the right-hand side f_0 of the model Eqs. . Note that with $x_1 = 1$ and $x_2 = 0$, they are identical to v_1 and v_2 in the introductory example (Section 4.1). To implement the transformed inputs, we need to invert the model Eqs. to get the inputs (independent variables)

$$F_1 = \frac{v_{0,2}(v_{0,1} - x_2)}{x_1 - x_2} \tag{17a}$$

$$F_2 = \frac{v_{0,2}(x_1 - v_{0,1})}{x_1 - x_2} \tag{17b}$$

(17) can be implemented as a nonlinear calculation block using Fig. 29. However, inspired by the linear feedback decoupling scheme in Fig. 26, an alternative implementation is shown in Fig. 30. To derive this scheme, we solve (16a) with respect to F_1 and we solve (16b) with respect to F_2, to get

$$F_1 = F_2 \frac{v_{0,1} - x_2}{x_1 - v_{0,1}} \tag{18a}$$

$$F_2 = v_{0,2} - F_1 \tag{18b}$$

These equations are coupled, but may be solved by feedback as shown in the simple control structure in Fig. 30. The resulting transformed system from v_0 to y is $y = I v_0$, so with no model error, we have perfect feedforward control, decoupling and linearization. The role of the two outer PID-controllers C_1 and C_2 in Fig. 30 is to correct for model uncertainty and unmeasured disturbances.

Besides being simple to understand and implement, the advantage with the implementation in (18) and Fig. 30, compared to an inversion using (17), 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) is the most important variable to control then we may want to give up $y_2 = F$ (flow) also when F_1 saturates. 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 $y_2 = F$. (I did not find this anti-windup scheme in the literature, so it should be tested in simulations before implementation).

5. Comparison of alternatives for switching

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

5.1. MV-MV switching

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

5.1.1. Split range control (E5, Fig. 21)

This solution has the advantage of being simple to understand, because of the nice visualization with the split range block. However, one disadvantage is that one must use the same integral and derivative time for all MVs. Split range control is therefore the ideal solution for cases where the dynamics with all MVs are similar, for example, for sequencing of multiple valves or pumps when a wide throughput range is required.

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 and Skogestad (2020a).

Another (and usually more serious) disadvantage is that split range control 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 adding fairly complex programming and/or logic. Note that this problem arises when a min (or max) selector is placed on an output from a split range block. Placing a min (or max) selector before the split range block does not cause this problem, for example, see the adaptive cruise control example in Section 5.3 (Fig. 31).

5.1.2. Multiple controllers with different setpoints (E6, Fig. 22)

This is often 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.

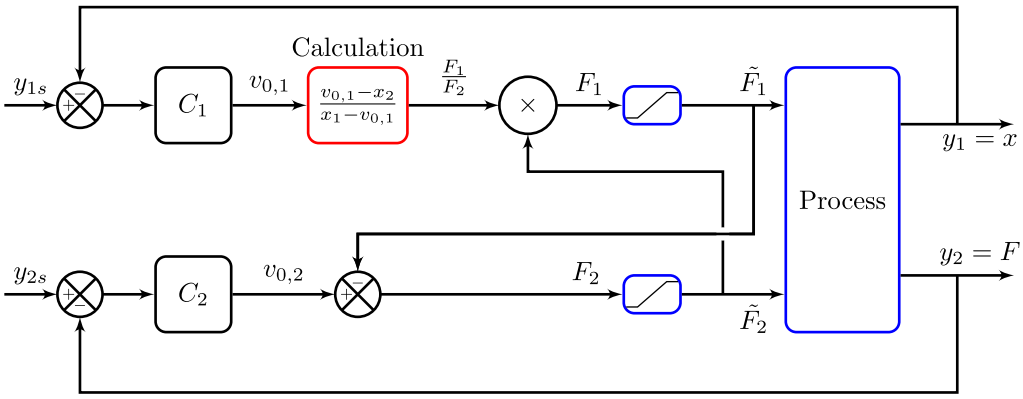


Fig. 30. Simple control structure which provides decoupling, feedforward control and linearization for the mixing process (blending system) in Fig. 28.
The output from the feedback controller C_1 is the ideal transformed input $v_{0,1}$. From this and measured disturbances (inlet compositions x_1 and x_2), the feedforward calculation element (red) uses (18a) to compute F_1/F_2. The decoupling is given by one multiplication element and one subtraction element. To work also in the case of input saturation, it uses the actual measured flowrates ($\tilde{F}_1, \tilde{F}_2$) a 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 we need two inner flow controllers (for F_1 and F_2) which are not shown in the figure.

The CV setpoints needs to be different for each MV, which may be seen as a disadvantage, but in some cases this may be an economic advantage. Reyes-Lúa and Skogestad (2019) discusses the example of temperature control using heating and cooling where we save energy by having a lower setpoint for heating (used in the winter) than for the cooling (used in the summer). 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 different setpoints, the consumption of inert gas is less than with split range control.

The main disadvantage with different setpoints is that we lose control for some time during switching. We cannot make the setpoint difference too small, because this will result in undesired switching because of disturbances and noise. Therefore, multiple controllers with different setpoints should not be used for applications where it is necessary to keep a constant setpoint, for example, for a critical reactor temperature control (Smith, 2010) (p. 102).

5.1.3. Input (valve) position control for MV-MV switching (E7, Fig. 24)

The advantage with the VPC solution 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 it 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 with VPC 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.

5.2. CV-CV switching

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

1. Selector on the MV (input u) (most general) (Fig. 17)
2. Selector on a CV setpoint if we use a cascade implementation (Fig. 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 when 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 ($u_{1,0}, u_{2,0}$) 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.

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

This is not a process control example, but nevertheless it should be known to most readers. 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 Fig. 31. 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:

- 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 (Fig. 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

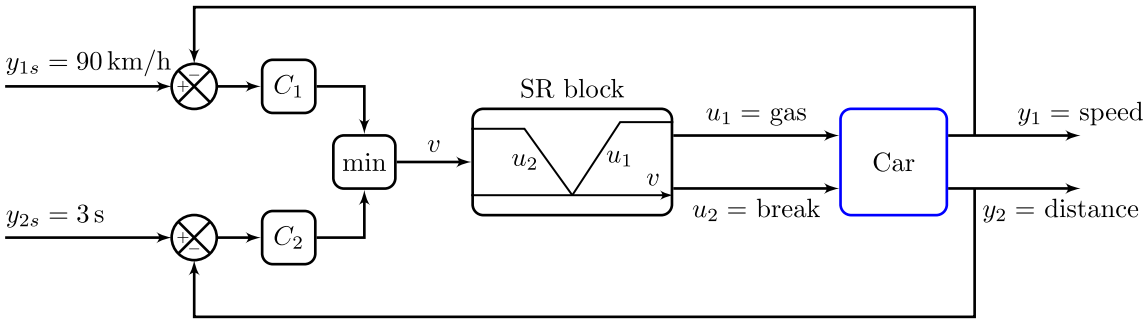


Fig. 31. Adaptive cruise control with selector and split range control.

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 is selected to control speed at all times, it requires using both gas (u_1) and breaking (u_2) at the same time for the downhill case 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.

5.4. 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.

5.5. 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 keeping 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, as discussed in the next example.

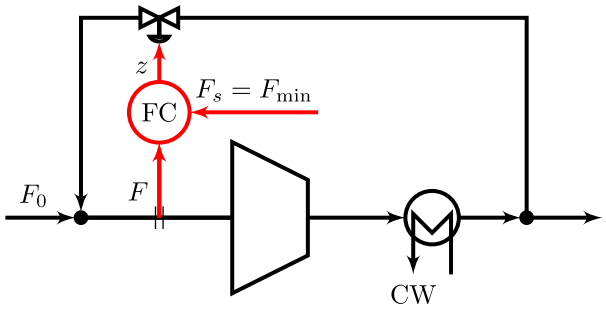


Fig. 32. Flowsheet of anti-surge control of compressor or pump (CW = cooling water). This is an example of simple MV-CV switching: When MV=z (valve position) reaches its minimum constraint ($z = 0$) we can stop controlling CV=F at $F_s = F_{min}$, that is, we do not need to do anything except for adding anti-windup to the controller. Note that the valve has a "built in" max selector.

5.5.1. 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 (Fig. 32). 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_{min}), that is, the constraint is $F \geq F_{min}$. With the simple feedback scheme in Fig. 32, 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_{min}), and at this point the constraint becomes over-satisfied, so it is optimal to stop controlling CV=F at $F_s = F_{min}$.

Let us check that the solution in Fig. 32 follows our selector rules. The minimum flow constraint is satisfied by a large valve opening (MV) so it requires a max-selector (Rule 1 for selectors). However, we have no selector in Fig. 32. The reason is that we make use of the "built-in" selector in the valve. Let us explain why it works: The low input constraint ($z = 0$) for the valve corresponds to a "built-in" max-selector (Rule 3 for selectors). Since both constraints give a max-selector, there is no conflict, and we can use the "built-in" selector for both constraints.

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

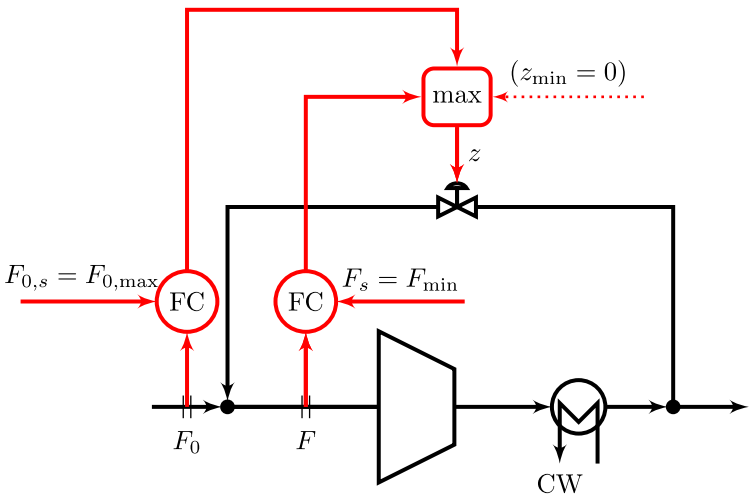


Fig. 33. Anti-surge compressor control with two CV constraints. This is an example of simple MV-CV-CV switching. MV = z, $CV_1 = F$, $CV_2 = F_0$ (all potentially active constraints).

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 Fig. 33.

The MV constraint ($z_{min} = 0$) is included as an input to the max-selector in Fig. 33 to show clearly that it is consistent with the other two constraints. However, because of the "built-in" max-selector in the valve, it is not really needed and this is why it shown with a parenthesis and dashed 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 will have to give up the constraint on F or F_0 (whichever is active at the moment).

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

We need anti-windup in both flow controllers in Fig. 33. If one uses back-calculation as in (C.18) then for both controllers, $\tilde{u}$ is the output z 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, which may be an advantage, 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).

5.6. *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).

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.

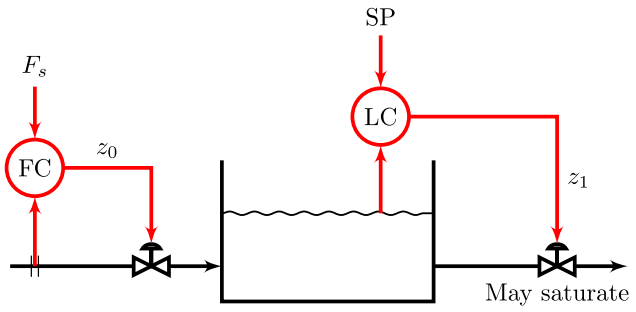
5.7. *Example complex MV-CV switching: Bidirectional inventory control*

Consider inventory (level) control of a single unit (tank) for the case where the inflow is given. The level (CV) then needs to be controlled using the outflow as shown in Fig. 34a. 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 switch to using the inflow (alternative MV), which means that we can no longer keep the inflow F at the desired setpoint F_s.

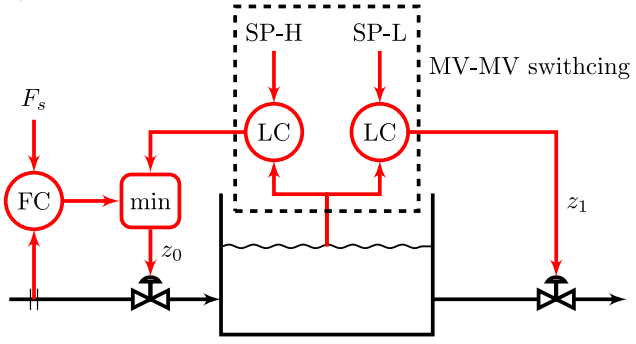
The required repairing of loops is a case of complex MV-CV switching which can be realized by a combination of MV-MV switching (using two level controllers with different setpoints, SP-L and SP-H) and CV-CV switching for the inflow (using a min-selector) as shown in Fig. 34b. This solution is also known as bidirectional inventory control (Shinskey, 1981) (Zotică et al., 2022).

6. Design of regulatory control layer with focus on inventory control

The main focus in this paper is on the "advanced" supervisory control layer which aims at keeping the "economic" controlled variables (active constraints and self-optimizing variables; CV1 in Fig. 4) at given setpoints. The supervisory layer sits on top of a basic regulatory PID control layer, and the design of the controllers (e.g., selecting PID tunings) starts from the bottom, usually with the inventory control system, which is the focus of this section. 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. Thus, inventory control involves control of liquid levels and certain pressures (CV2 in Fig. 4).



(a) Normal inventory control (cannot handle saturation in the outflow valve z_1).



(b) Bidirectional inventory control (handles saturation in outflow valve z_1 by complex MV-CV switching).

Fig. 34. Normal inventory control (cannot handle saturation in the outflow valve z_1).

The task of designing the inventory control system is greatly simplified by identifying or choosing the "throughput manipulator" (TPM). The "pair-close" pairing rule then results in the "radiation rule" for inventory control as discussed next.

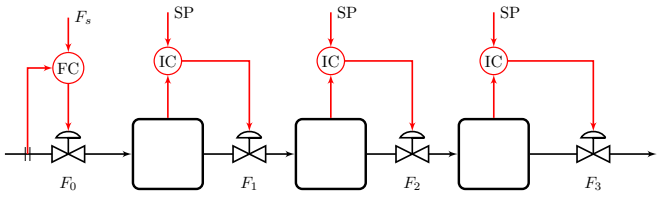
6.1. *Throughput manipulator and radiation rule*

Consider inventory control of units in series (Fig. 35), which is an extension to the single tank example in Section 5.7 (Fig. 34). The task of designing the inventory control system is greatly simplified by identifying or choosing the "throughput manipulator" (TPM). Here is a simple definition:

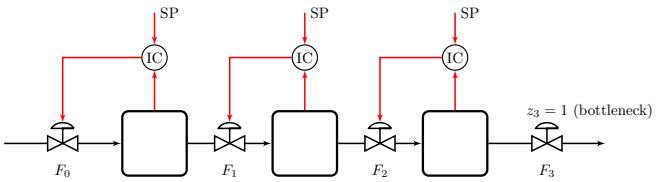
> *TPM = Variable used for setting the throughput/production rate (for the entire process).*

The TPM is often an MV but it can also be a disturbance. Usually the TPM is a flowrate, but it can in some cases even be an 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 an 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. The most common TPM location is at the feed (process inflow) or at the product (process outflow). In terms of economics and maximizing production, a good choice, in order to minimize the back-off from active constraints, is to locate the TPM close to the production bottleneck (Downs & Skogestad, 2011). This could be at the feed or at the product, but it is more generally inside the process.

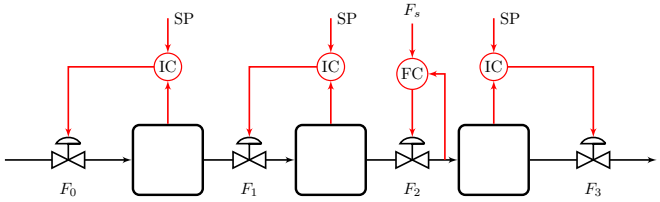
For the units in series, consider first the common case in Fig. 35(a) where the feed flow is given, which means that TPM=F_0. In this case, the inventories need to be controlled using their outflows, that is, inventory control is in the direction of flow. However, if the inflow becomes too large then we may encounter a bottleneck, for example, the outflow valve of the last unit may saturate at fully open ($z_3 = 1$).



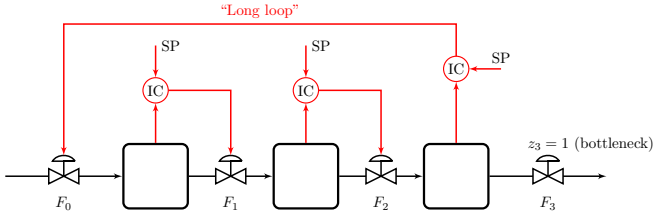
(a) Inventory control in direction of flow (for given feed flow, TPM = F_0)



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



(c) Radiating inventory control for TPM in the middle of the process (shown for TPM = F_2)



(d) Inventory control with undesired "long loop", not in accordance with the "radiation rule" (for given product flow, TPM= F_3)

Fig. 35. Inventory control for units in series. Cases (a), (b) and (c) are in accordance with the "radiation rule".

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 last inventory. This results in the control structure in Fig. 35(d) 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 to the direction of flow as shown in Fig. 35(b).

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 (Aske & Skogestad, 2009; Buckley, 1964; Price et al., 1994):

Radiation rule (Fig. 35): 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 Fig. 35(a) (TPM at feed) to Fig. 35(c) (TPM at product), requires rearranging three loops. Fortunately, it turns out that the reuse of the bidirectional inventory control structure discussed in Fig. 35(b) solves the problem in an elegant way. This is the topic of Section 6.3 (Fig. 36), but let us first consider controller tuning.

6.2. What is the purpose of having inventories (buffer tanks)? Fast or slow level control?

Buffer tanks are put between process units to provide mass or energy holdup (inventory) by storing liquid, gas or solids. To design the inventory control system and choose inventory setpoints, we need to know the reason for installing the tank. The two main purposes for installing buffer tanks are (Faanes & Skogestad, 2003; Lindholm et al., 2010):

A. To reduce propagation of disturbances between units during continuous operation (surge tank).
B. To allow for independent operation of process units by using a variable inventory to isolate units from each other, for example, during a temporary shutdown of a unit or for processes with both continuous and batch operation.

The two purposes often give opposite demands on the level control tuning. Faanes and Skogestad (2003) focus of category (A) where to "average out" flowrate disturbances, we want to use slow (non-aggressive) inventory control ("averaging level control"). On the other hand, the focus of the next section on bidirectional inventory control is on category (B), where to make use of the storage capacity of the tank, a setpoint close to the top or bottom of the tank is often desired, which calls for tight (aggressive) level control (Lindholm et al., 2010).

6.3. Bidirectional inventory control for units in series

We consider operation of units in series where the main reason for installing the buffer tanks is to maximize the throughput by keeping the production going also during temporary stops of a unit. Thus, this belongs to category (B) where tight level control (close the full or empty) is desired to make maximum use of the available storage capacity.

The bidirectional inventory control in Fig. 36 (Shinskey, 1981) achieves two goals (Zotică et al., 2022):

- Rearrange the loops according to the "radiation rule" when the bottleneck moves.
- Maximize the throughput over time by using the inventories dynamically and switching optimally between high and low inventory setpoints when the bottleneck moves.

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. Typically, we may set SP-H=90% and SP-L=10%. This accomplishes MV-MV switching between inflow and outflow. For each flow (valve) the decision on what to control (CV-CV switching) is made by a min-selector.

The inventory controllers should then be fairly tightly tuned. This is to avoid overflowing (inventory=100%) or emptying (0%) the units (tanks). We have also introduced flow setpoints ($F_{0s}, F_{1s}, F_{2s}, F_{3s}$) to be able to set the flow (or valve position) at each location, but since it enters a min-selector, the setpoint is in reality a maximum flowrate constraint. If a flow setpoint is set at a sufficiently low value, then the corresponding valve 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 Fig. 36 will automatically make use of the inventories to maximize the throughput, identify the bottleneck, and give a radiating control system around this bottleneck (Zotică et al., 2022). Yes, it is almost like magic! (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 [by changing the setpoints F_s] without loss of inventory control. 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 controller [SP-L for Unit 1] 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 system also accommodates constraints at intermediate points. Suppose a filter into unit 2 began to clog, reducing flow into tank [unit] 2. Its falling level would cause [SP-L for Unit 2] and eventually [SP-L for Unit 3] to manipulate downstream flows. Meanwhile, the level in tank [unit] 1 would rise, causing [SP-H for Unit 1] to reduce the feed to match the rate of outflow.... 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".

Zotică et al. (2022) demonstrates the effectiveness by simulations and find that the solution makes optimal use of available storage for isolating temporary bottlenecks.

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

In the bidirectional inventory control scheme in Fig. 37, we have added (in red color) some extra selector logic to avoid a minimum flow constraint on the intermediate flow F_2 (Bernardino & Skogestad, 2023). This may be desirable, for example, if unit 3 cannot operate at a low load. To be able to keep a large flow F_2 also when F_1 or F_3 are small (at least temporary), we increase the low inventory setpoint in the upstream unit 2 (from L to M_L) and decrease the high inventory setpoint in the downstream unit 3 (from H to M_H). The setpoint values for M_L and M_H depend on the nature of future disturbances and whether it is most important to keep production at its maximum or to F_2 keep large. As a starting point one may set, for example, $L = 10\%, M_L = 40\%, M_H = 60\%$ and $H = 90\%$.

The control structure in Fig. 37 may easily be dismissed as being too complicated so MPC should be used instead. At first this seems reasonable, but a closer analysis shows that MPC may not be able to solve the problem (Bernardino & Skogestad, 2023).[8] Besides, is the

[8] It seems difficult to design an MPC that achieves the objective of the structure in Fig. 37, which is to maximize throughput for cases with temporary bottlenecks, while at the same time protecting against a minimum flow constraint. The response of the simpler control structure in Fig. 36, 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 unachievable high setpoint for the four flowrates (F_0, F_1, F_2, F_3). However, this trick does not seem possible to apply for the more complex case in Fig. 37 because of the minimum flow constraint. Without using the trick of unachievable high flow setpoints, it is not even clear if MPC can handle the simpler case in Fig. 36. We can tell MPC about the minimum and maximum level constraints, but how does MPC know where to keep the level during steady state operation? It seems MPC would need to know the future disturbances (which is impossible), or a least MPC needs a scenario of expected disturbances, which makes the problem definition and solution complicated. The further study of this is left as a challenge to the MPC community.

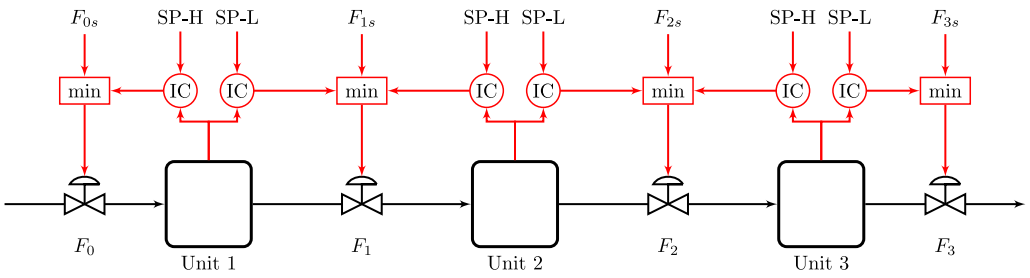


Fig. 36. Bidirectional inventory control scheme for automatic reconfiguration of loops (in accordance with the radiation rule) and maximizing throughput (Shinskey, 1981) (Zotică et al., 2022).
SP-H and SP-L are high and low inventory setpoints, with typical values 90% and 10%. Strictly speaking, since there are setpoints on the (maximum) flows ($F_{i,s}$), the four valves should have slave flow controllers (not shown). However, one may instead have setpoints on valve positions (replace $F_{i,s}$ by $z_{i,s}$), and then flow controllers are not needed.

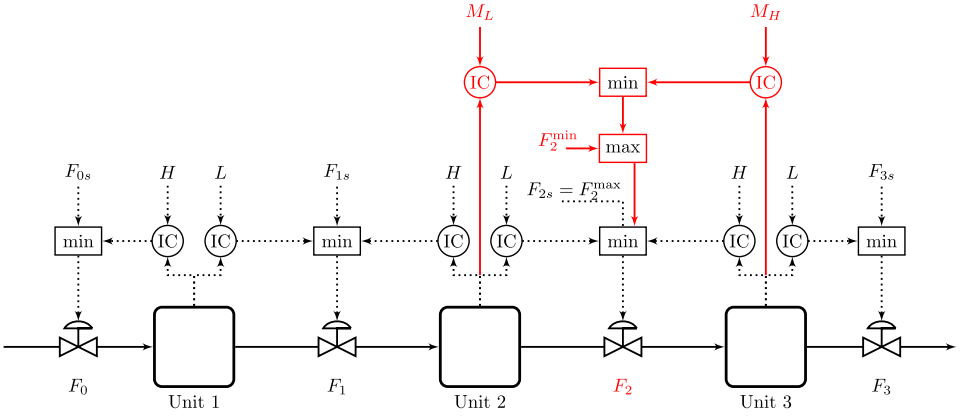


Fig. 37. Bidirectional inventory control scheme for maximizing throughput (dashed black lines) while attempting to satisfy minimum flow constraint on F_2 (red lines). H, L, M_L and M_H are inventory setpoints.

control structure in Fig. 37 really that complicated? Of course, it is a matter of how much time one is willing to put into understanding and studying such structures. Traditionally, people in academia have dismissed almost any industrial structure with selectors to be ad hoc and difficult to understand, but this view should be challenged.

To this end, we provide an explanation for the red selector logic in Fig. 37. 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 (Fig. 35(a)), 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 M_L 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 $F_2 \geq F_2^{min}$. If this is not allowed, then we either need to stop Unit 3 (and set $F_2 = 0$) or alternatively we can introduce recycle around Unit 3 (if possible). 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 inventory in unit 3 will be at H. So if we can increase F_0 again 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.

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

Fig. 38 shows another seemingly complex bidirectional inventory control structure for an industrial feed water treatment plant (case study provided by Krister Forsman at the Perstorp company). We here give an explanation of how it works.

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 rate (F_{4s}) and finally there are maximum and minimum constraints on all six 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 operates in an on/off fashion. This is the reason why the two corresponding inventory controllers in Fig. 38 are on/off hysteresis controllers which, depending on which of the two on/off controllers 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 when this cannot be achieved (because 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

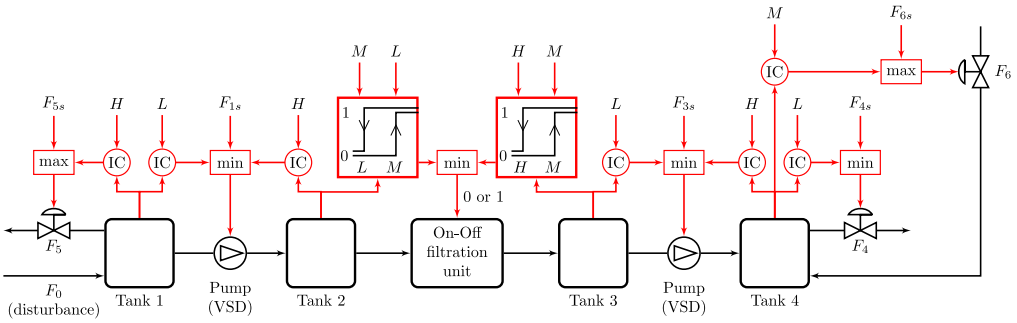


Fig. 38. Bidirectional inventory control structure for industrial plant with on/off (1/0) control of filtration unit. H, L and M are inventory setpoints with typical values 90%, 10% and 50%. If it is desirable to set a flowrate (F_s) somewhere in the system, then flow controllers must be added at this location.

temporary reductions in these flows 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 a steady-state offset.

7. Discussion

7.1. Design of the overall control system

The aim of this paper is to present the various standard control elements and illustrate their use, with particular emphasis on how to handle changes in active constraints (MV-MV, CV-CV and MV-CV switching).

A much more complex topic is the design of an overall decomposed control system for a given process, which involves the structural decision of selecting variables (inputs, controlled variables, measurements) and interconnecting the variables using the standard control elements (E1–E18). This topic has been discussed in a few papers, including (Morari et al., 1980), Skogestad (2004a), Downs and Skogestad (2011) and Minasidis et al. (2015), but a lot more work remains to be done.

With reference to Fig. 4, Skogestad (2004a) proposes a top-down analysis (steps 1–4) followed by a bottom-up design (steps 5–7) with the following main steps:

1. Define operational objectives, including identifying constraints and a cost function J to be minimized.
2. Identify dynamic and steady-state degrees of freedom.
3. Choose primary (economic) controlled variables (CV1), including active constraints and self-optimizing variables for the unconstrained degrees of freedom.
4. Select the location of the throughput manipulator (see Section 6.1 for details).
5. Basic regulatory control layer: Identify stabilizing controlled variables (CV2) and select how to pair these with manipulated variables (u).
    - The two main pairing rules are the "pair-close" and "input saturation" rules (Section 2.6)
    - For inventory control use the "radiation rule" (Section 6.1).
6. Supervisory control layer which controls economic variables (CV1), tracks changes in active constraints (CV-CV switching) and avoids saturation in the basic control layer (MV-MV switching): Make use of cascade control, ratio control, valve position control, feedforward control, decoupling, selectors, MPC etc. as needed to get acceptable control performance in face of disturbances. The design of this layer is the focus of the present paper.
7. Optimization layer: Compute optimal setpoints and identify active constraints. The focus in this paper is to move these tasks into the control layers whenever possible.

Finally, step 8 is to validate the proposed control strategy using dynamic simulation, whenever possible. Minasidis et al. (2015) present some additional guidelines and rules for this procedure. For example, for the economic controlled variables (CV1), two rules are to *"never control a variable which is optimally at a maximum or minimum, like the cost function J"* (also see Section 2.7.1) and *"always control the purity constraint of a valuable product"* (because it is always an active constraint).

7.2. Understanding and improving advanced industrial control solutions

An important contribution of this paper is to provide a systematic overview of the "advanced" control elements used in industry. With this knowledge, it should be possible to understand most industrial solutions and also to propose alternatives and improvements.

When I started studying the advanced control solutions used by industry, only a few years ago, I was rather confused. I did not understand what the various control strategies where attempting to do, especially in regards to constraint switching. We then realized that there are two main cases of constraint switching, namely CV-CV switching (where one always uses a selector) and MV-MV switching (where there are three alternatives) (Reyes-Lúa & Skogestad, 2020b). In addition, there is the simple MV-CV switching where one does not need to do anything (one just "gives up" the CV constraint when the MV saturates), and finally we have complex MV-CV switching (which is the "repairing of inputs and outputs" or "rearranging of loops" case) where one must combine MV-MV and CV-CV switching. Note that in complex MV-CV switching, the MV-MV switch comes before the CV-CV switch. If the order is reversed, as for the adaptive cruise control in Fig. 31, then we have a different case where MV-MV and CV-CV switching are used independently.

Here is a summary of some additional insights from this paper:

- If the industrial solution has a selector (sometimes realized using a saturation element, 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 (E4), although there may be exceptions as seen in 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 (Fig. 17), or as a cascade implementation with a selector on the CV setpoint (Fig. 19).

  - If there are several selectors (max and min) in series then we know that the constraints are potentially conflicting and that the highest priority constraint should be at the end (Fig. 18).

- If the industrial solution has a valve position controller (VPC) then there may be two quite different problems that it is addressing (see E3 and E7 in Table 1), and it may not be immediately clear which.

  1. If we have an extra MV for dynamic reasons (E3; Fig. 12) then the two controllers (and MVs) are used all the time. The MV manipulated by the VPC (MV_1 in Fig. 12) is then used on the longer time scale, whereas the MV linked to the CV (MV_2 in Fig. 12) is used for dynamic reasons (fast control). Here, an alternative is to use parallel control (Fig. 13).
  2. There is also another possibility, namely, when the VPC makes use of an extra MV to avoid that the primary MV saturates at steady-state (E7; Fig. 24). This is then a case where the VPC is used for MV-MV switching and the VPC is only active part of the time.

- For MV-MV switching there are three alternatives.

  1. A common solution is split range control (E5; Fig. 21) which is usually easy to identify.
  2. Another common solution is multiple controllers with different setpoints (E6; Fig. 23). It may be a bit more difficult to identify.
  3. Finally, there is VPC (E7), as just discussed, which is probably the least common solution for MV-MV switching

  One should have all these three alternatives in mind when choosing the best solution for MV-MV switching, as there is not one alternative which is best for all problems (see Section 5.1 for details).

7.3. Cross-limiting control and other special structures

Industry also makes use of other smart solutions, which do not follow from the standard structures presented in this paper.

One example is cross-limiting control for combustion, where the objective is to mix air (A) and fuel (F) in a given ratio, but during dynamic transients, when there will be deviations from the given ratio, one should make sure that there is always excess of air. The scheme in Fig. 39 with a crossing min- and max-selector achieves this. It 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 Fig. 39 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 Fig. 39 without the two selectors. The only "strange" thing to notice about this structure (without the selectors) is that also 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 at steady state it is a ratio controller.

Now let us look at how it works with the two selectors included, which has an effect on the transient behavior. 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

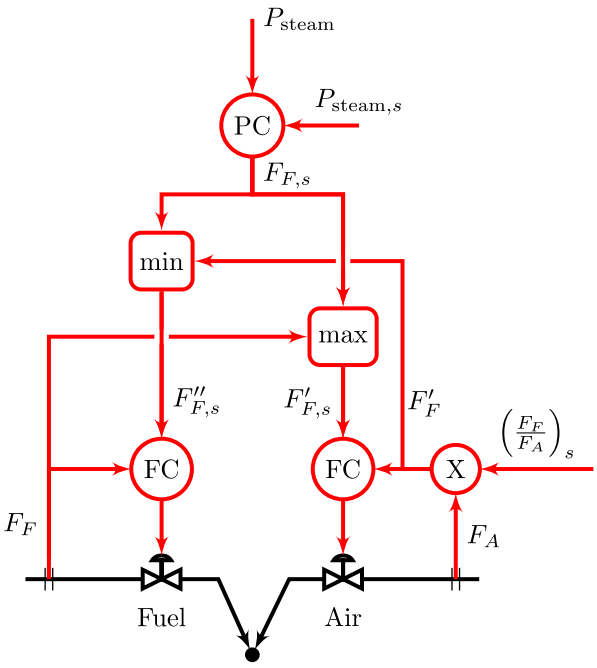


Fig. 39. Cross-limiting control for combustion where air (A) should always be in excess to fuel (F).

the controller (PC) demands lower flows, the fuel flow decreases 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 additional smart structures ("inventions") which are not discussed in this paper, for example, some are found in the books by Shinskey. Also (Liptak, 1999) shows 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 Fig. 39, 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.

7.4. Smith predictor

Note that the Smith Predictor (Smith, 1957) is not included in the list of 18 control elements given in the Introduction, although it is a standard element in most industrial control systems to improve the control 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 (Grimholt & Skogestad, 2018b; Ingimundarson & Hägglund, 2002). The exception is cases where the true time delay is known very accurately. 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 with the Smith Predictor (and also with IMC and MPC) 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

$$\text{DM} = \frac{\text{PM}}{\omega_c} \tag{19}$$

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 (and IMC) may have multiple crossover frequencies, resulting in very large values for ω_c and thus in a very small delay margin.

7.5. Optimization with constraints and theoretical basis for selectors

For real-time optimization (RTO), we have proposed using simple feedback implementions, including CV-CV switching with selectors. Is this optimal? As shown below, the use of selectors itself is optimal, so this is not just some ad-hoc industrial fix used by engineers. However, it is not always possible to use selectors directly on the inputs as shown in Fig. 17; in the more general case the selectors should be on the Lagrange multipliers λ as shown in Fig. 40.

Consider a static constrained optimization problem (RTO poblem),

$$\min_u J(u.d), \quad \text{subject to } g(u,d) \le 0 \tag{20}$$

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

$$\min_{u,\lambda} \underbrace{(J(u,d) + \lambda g(u,d))}_{\mathcal{L}(u,d,\lambda)} \tag{21}$$

with the following necessary optimality (KKT) conditions

$$\nabla_u \mathcal{L} = 0, \quad \lambda \ge 0, \quad g \cdot \lambda = 0 \tag{22}$$

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 $\lambda \ge 0$ 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 Fig. 40 (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 $\text{CV}_s = 0$) by manipulating the dual variable (MV=$\tilde{\lambda}$). This value is send to a max-selector, $\lambda = \max(\tilde{\lambda}, 0)$, 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 Fig. 40 this problem is solved by feedback using a "gradient controller" but it could alternatively be solved numerically using a calculation block (plus a dynamic filter or a lower-level control layer for implementing u). Importantly, the max-selector in Fig. 40 provides the optimal transition between optimal constrained and unconstrained steady-state operation (and the reverse), in a similar way to the selectors elements used in this paper.

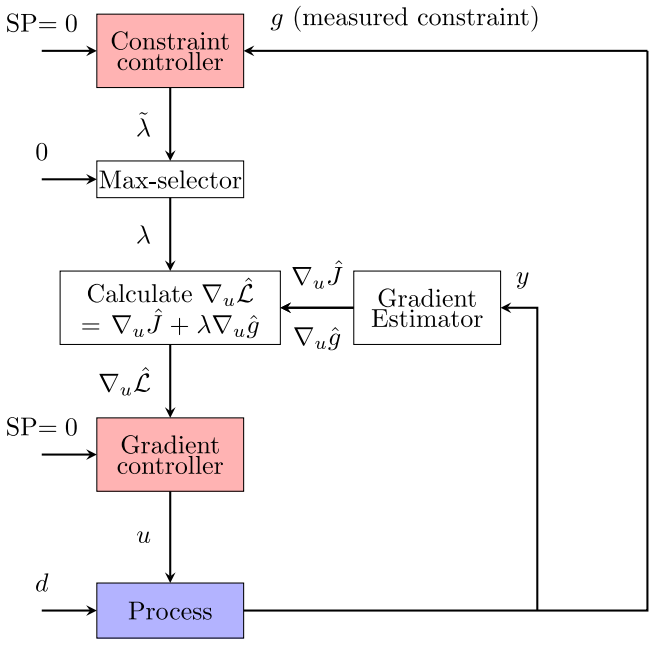


Fig. 40. Dual decomposition of constrained optimization with upper (slow) master constraint controller and max-selector on the dual variable λ (Lagrange multiplier).

7.6. Critique of MPC

The defining feature of model predictive control is a repeated optimization of an open-loop performance objective over a finite horizon extending from the current time into the future (Eaton & Rawlings, 1992). In this discussion section, shortcomings, advantages and more fundamental limitations of MPC are pointed out. It may seem strange to discuss and criticize MPC in a paper about advanced regulatory control (ARC). However, a discussion about MPC shortcomings is included because many engineers and researchers think that the industrial approaches (ARC) are outdated and ad hoc and will be replaced by MPC.

7.6.1. 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 k, the optimal input u_k 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. In discrete form, the objective is to minimize the aggregated cost J from the present time ($k = 0$) and to the end of the prediction horizon ($k = N$):

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

The cost J is minimized subject to given model equations, e.g. $dx/dt = f(x,u,d)$ (appropriately discretized), and operational constraints, $g_k \le 0$. 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 to the dynamic control performance, so the total cost is $J = J_\$ + J_c$. However, EMPC is rarely used in practice, both because it may be 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.

7.6.2. Conventional MPC (with setpoints)

Conventional MPC is setpoint-based, so it should ideally be combined with an upper real-time optimization layer (RTO, usually static) which computes the optimal setpoints y_s. A good introduction to conventional MPC, which emphasizes its predictive capabilities (when we have knowledge about future changes for setpoints, disturbances or prices) compared to standard feedback control, is given by Eaton and Rawlings (1992).

Conventional MPC tracks the setpoints in an "optimal" way by minimizing at each sample time $k = 0$ the following quadratic cost function

$$J_c = \sum_{k=0}^{k=N} (y_k - y_{s,k})^T Q (y_k - y_{s,k}) + \Delta u_k^T R \Delta u_k \tag{23}$$

Here, Δu_k represents the input change between samples, and Q and R are weight matrices. Also here only the first input change (Δu_0) is implemented and there is a moving horizon where the optimization problem is resolved at each sample time. 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, but 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 and 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 become very large, and the original repeated open-loop solution based on (23) 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.

7.6.3. 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 has 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. Rather, MPC will be applied on top of cascade (PID) and ratio control.

7.6.4. Integral action and MPC

Consider the simple setpoint tracking problem in Appendix B. Fig. B.43 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? The answer is that with some measurement error (which must be included in the estimator problem), MPC will give the feedforward solution. 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 of Cutler and Ramaker (1980)) was to let the 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 avoids 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. In more complex cases, the strength of MPC is that one can easily coordinate the use of feedback and feedforward control, for example, by changing the weight (magnitude) of the integrating disturbance.

7.6.5. Cascade control and MPC

MPC is not the right tool when cascade control (Fig. 9) is the preferred solution. The problem with MPC is that it cannot make use of an extra process measurement (w) unless it has a model of how the output y and the measurement w are related. In addition, even with such a model, it is not clear how MPC should 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 (Fig. 9) an engineer can easily make use of an extra measurement w, by just using the physical insight that fast control of w will indirectly benefit the control of y. Here, w_s becomes the new manipulated variable (to replace u) for control of y. The tuning of the two controllers may be done online in a sequential manner, starting with the fast inner controller for w.

As an example, assume that we want to use the outflow valve to control the level in a tank, and we want to use of a flow measurement to replace an uncertain or unknown valve model. Here, $u = z$ (valve position), $w = F$ (extra flow measurement) and y = tank liquid volume (measured). The uncertain nonlinear valve model may be written $w = f(u, d_w)$ (static), and the mass balance for the tank gives $dy/dt = d_y - w(u)$ where d_y is the inflow (disturbance) and $w(u) = F$ is the outflow of the tank. With conventional cascade control (ARC), we may tune a flow controller (e.g., an I-controller with only one tuning parameter) online without using the valve model, and the setpoint w_s will be a degree of freedom (MV) for the level controller. With MPC we need an estimator for MPC to make use of the flow measurement (w), and it is not clear how the estimator can be tuned to avoid that MPC makes too much use of the uncertain valve model. Probably, we would need to assume that the disturbance d_w affecting the flow w is very large (use a large weight for d_w) and that the noise on the flow measurement is very small (use a small weight for n_w). In any case, a valve model will need to be supplied to MPC, even if it is not important for tuning the controller.

In practice, the preferred solution with MPC is to first implement a slave flow controller (e.g., an I-controller) and let the flow setpoint be the MV for MPC. However, as pointed out by Kumar et al. (2023), there are some difficulties here, especially related to the fact that controllers with integral action need anti-windup. One solution is to include in MPC a model of the slave controllers (Kumar et al., 2023).

7.6.6. Ratio control and MPC

A typical application of ratio control is for mixing, where the manipulated flowrate (u) should be increased proportionally to a given measured flowrate (d) such that their ratio $R = u/d$ is kept constant, see (6). 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 the viscosity. On the other hand, a simple ratio control implementation (e.g., Fig. 11) does not require a model for how y depends on u and d; we just need the physical insight that y remains constant if we keep the ratio u/d constant (see Section 3.3.3).

7.6.7. 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 from the control engineer (designer), the three main inventions of process control; namely integral control, ratio control and cascade control (see above).
3. Since a dynamic model is usually not available at the startup of a new process plant, we need initially a simpler control system, typically based on advanced regulatory control elements. 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. For example, it may be difficult to tune MPC to have fast feedforward control for disturbances because it may affect negatively the robustness of the feedback part (Pawlowski et al., 2012).
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 and Skogestad (2012), e.g., see (C.14) for the gain margin.
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 7.6.9).

Shortcomings List 1, Lists 4 and 5 are related and become more serious for larger problems. Thus, even with MPC, the problem is often decomposed, for example, by using separate MPCs for each process unit, possibly with a coordinator MPC on top. There have been many academic efforts over the last 30 years to deal with shortcomings Lists 5 and 6, and significant progress has been made. However, these new approaches makes the problem even more difficult to formulate and solve.

7.6.8. 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. Pure feedforward control and coordination of feedforward and feedback control.
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.). Actually, for the Tennessee Eastman Challenge Process, Ricker (1996) found that ARC (using decentralized PID control) was better than MPC. Ricker (1996) writes in the abstract: "There appears to be little, if any, advantage to the use of NMPC (nonlinear MPC) in this application. In particular, the decentralized strategy does a better job of handling constraints – an area in which NMPC is reputed to excel". In the discussion section he adds: "The reason is that the TE problem has too many competing goals and special cases to be dealt with in a conventional MPC formulation".

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 for a given problem, there may be cases where MPC is preferred because it is simpler to implement and understand.

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

With MPC, the optimal input is obtained by repeatedly solving an open-loop (feedforward) control problem (see Section 7.6.2). Feedback is only introduced indirectly by updating the initial states x_0. In addition, and this is more serious, it is frequently assumed that the states x are perfectly measured, which is not realistic, especially not in process control applications.

If all states are not measured, the standard MPC 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 with no model uncertainty. Here, "model uncertainty" refers to changes and errors in the process model, including changes in the process model parameters, for example, gain and time delay variations, which may move the closed-loop poles and cause instability for linear systems. The term "model uncertainty" does not include uncertainty in the exogenous signals (noise n and disturbances d), for which the separation principle holds for linear systems.

The failure of the separation principle 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.

In summary, the assumption of separating the estimation and control tasks, greatly simplifies the overall mathematical problem and it is very much in line with the main theme of this paper, which is to split the control system (and its design) into smaller elements. However, since the separation principle does not hold for systems with model uncertainty, the conclusion is that model predictive control is not as "optimal" as one may believe.

7.6.10. Problems in designing MPC and ARC controllers

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 states) and model uncertainty. This is excellent work, but so far little of this effort has impacted 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 mainly based on linear experimental models, often derived from step responses, and using 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 is 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 probably that they are viewed as being optimal and general. However, as explained above (Section 7.6.9), this is not true, 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 is 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).

In general, to be optimal (without quotes), the tasks of control and estimation need to be combined into one controller block, that is, one needs 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. One possible solution is to use dynamic programming, but this is known to have serious problems with computational complexity and curse of dimensionality, so in practice approximations or alternative methods must be used, for example, reinforcement learning or model predictive control.

A fundamentally different approach to the repeated open-loop optimization (MPC) is to specify that we want to use a precomputed controller C from measurements (y) to inputs (u) and to restrict the set of allowed controllers (for example, by fixing the order of the controller). A special case of using precomputed controllers is to use ARC. The optimization problem is then to search for the best controller parameters, for example, PID tuning parameters. However, this gives a very hard mathematical problem. As an example, consider the simplest case where we use proportional control, i.e., $u = Ky$, and we want to find the optimal gain matrix $C = K$. However, even in the linear case with no uncertainty, this optimal static output feedback problem is unsolved and believed to be non-convex and NP-hard. (e.g., Sadabadi and Peaucell (2016)). The optimization problem becomes even more difficult if we impose structural restrictions, for example, decentralized control (distributed control, horizontal decomposition) where we specify that given elements in the controller C are zero (e.g., Anderson et al. (2019)).

The mathematical problem is therefore usually simplified by *removing* decomposition restrictions, for example, by combining the control and optimization layers in Fig. 4 into a single Economic MPC (EMPC). This makes it 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.

In conclusion, the reason for including this critique section on MPC, is not to say that people should stop research on MPC or EMPC. On the contrary, impressive progress has been made over the last 30 years to make MPC a practical way of solving many important control, for example, by improving the numerical efficiency and robustness of nonlinear MPC. Rather, the discussion is included to point out that MPC is not the best solution all control problems. Therefore, it is worthwhile for the academic control community to focus research on the "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 improved design methods.

7.7. 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 uses the word "fragility" to describe this sensitivity of an optimized solution to unforeseen events, and he has coined the phrase "robust yet fragile" (Doyle et al., 2005). Carlson and 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 not acceptable, for concluding that MPC is the preferred solution.

8. Challenges to the academic control community

The topic this paper is the use of standard elements for control of complex industrial processes, here denoted advanced regulatory control (ARC). These industrial solutions are based on decomposing the overall controller. The engineer directly specifies the control structure and required control elements. An important advantage compared to more centralized solutions is that each tuning parameter usually has a direct and clear effect on the system responses, and that it may obtained experimentally or based on very simple models. Thus, 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 to propose a decomposed control structure. 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 using 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 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. The first goal of this research would be to develop rigorous design methods for each element (which should be relatively easy). The second, and much more difficult goal, would be to study system decomposition and how to put together an overall control system based on simple elements.

It is also worthwhile to look into some of the old industrial literature for ideas. Many specific control 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 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.

8.1. A list of specific research tasks

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

1. Vertical decomposition including time scale separation in hierarchically decomposed systems (considering performance and robustness)
2. Horizontal decomposition including decentralized control and input/output pairing
3. Selection of variables that link the different layers in the control hierarchy, for example, self-optimizing variables (CV1 in Fig. 4) and stabilizing variables (CV2).
4. Selection of intermediate controlled variables (w) in a cascade control system.[9]
5. Tuning of cascade control systems (Figs. 9 and 10)
6. Structure of selector logic
7. Tuning of anti-windup schemes (e.g., optimal choice of tracking time constant, τ_T) for input saturation, selectors, cascade control and decoupling.
8. How to make decomposed control systems based on simple elements easily understandable to operators and engineers
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 or library of special (smart) control structures (inventions) that solve specific control problems, for example, cross-limiting control

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 have been used in the lower-layer control loops. In addition, both for unstable and oscillating processes, a better approach may be to use cascade control on top of a fast inner P- or PD-controller which stabilizes or removes oscillations (see footnote 4). 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.

8.2. The harder problem: Control structure synthesis

The above list of research topics deals mainly with the individual elements. A much harder research issue is *the synthesis 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.

[9] 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 (e.g., one fast PD-controller for stabilization and one slow PI-controller for steady-state control which 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. See also Fig. 15.

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 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.

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 prove to 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 some 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. It is important that such a tool has a graphical interface, both for presenting the problem and for proposing and improving solutions.

8.3. MPC and summary challenges

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 both MPC and ARC should be in the toolbox of control engineers. However, a discussion about MPC shortcomings is included because many engineers and researchers think that the industrial approaches (ARC) are outdated and ad hoc and will be replaced by MPC. As argued in this paper, this should not happen, partly because MPC is itself is 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 analysis of switched systems (for example, with selectors, anti-windup and split range control) is mathematically very difficult. 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.

9. Conclusion

Control engineers rely on many tools, and although some people may think that in the future there will be one general universal tool that solves all problems, like economic model predictive control (EMPC), this is not likely to happen. The main reason is that the possible benefits of using more general tools may not be worth the increased implementation costs (including modeling efforts) compared to using simpler "classical" advanced regulatory control (ARC) solutions. In particular, this applies to process control, where each process is often unique. In addition, for a new process, a model may not be available, so at least for the initial period of operation a classical ARC scheme must be implemented.

Since its introduction in the 1940's, about 80 years ago, advanced regulatory control (ARC) has largely been overlooked by the academic community, yet it is still thriving in industrial practice, even after 50 years with model-based multivariable control (MPC). So it is safe to predict that ARC (including PID control) will not be replaced by MPC, but will remain in the toolbox along with MPC. Thus, it is time to give classical ARC a "new beginning" in terms of strengthening its theoretical basis and training engineers and students on how to use it in an effective manner. Classical ARC includes the standard control elements (Table 1) that industry commonly uses to enhance control when simple single-loop PID controllers cannot achieve the desired control performance. Examples of such control elements are cascade control, ratio control, selectors, split range control, valve position control (VPC), multiple controllers (and MVs) for the same CV, and nonlinear calculation blocks.

This paper takes a systematic view on how to design classical ARC system. The starting point is usually optimal steady-state economic operation. The process may have many manipulated variables (MVs) for control (typically valves), but usually most of these are used to control "active" constraints, which are the constraints which optimally should be kept at their limits at steady state. For the remaining unconstrained degrees of freedom, we should look for self-optimizing variables, which are measured variables for which the optimal values depend weakly on the disturbances.

In terms of control system design, we usually start by designing a good control system for the normal (nominal) operating point, preferably based on single-loop PID controllers where each manipulated variable (MV), which is not optimally at a constraint, is paired with a controlled variable (CV). To handle interactions, disturbances and nonlinearity, one may add cascade control and calculation blocks. However, during operation one may reach new (active) constraints, either on MVs or CVs, which may be easily observed from measurements of the potential constraints. Since the number of control degrees of freedom does not change, we will need to give up the control of another variable, which will either be another constraint (on CV or MV) or an unconstrained CV (self-optimizing variable). The key is then to know which variable give up, and in many cases we may determine this based on physical insight, and implement it using standard ARC elements, for example, using selectors. Thus, active constraint switching and close-to-optimal economic operation under varying conditions can usually be achieved without real-time optimization (RTO).

A key new observation is that there are only four cases of switching and these may be handled by using standard ARC control elements (Sections 2.8 and 5): For CV-CV switching we use selectors (overrides), for MV-MV switching we use split range control or similar, for simple MV-CV switching (where the two variables are already paired) we do not need to do anything (except for including anti-windup in the controller) and for complex CV-MW switching we need to combine CV-CV and MV-MV switching.

The main disadvantage with ARC compared to MPC is that it is based on single-loop controllers, so one needs to pair outputs (CVs) with inputs (MVs). For most processes this works well, but for more complex cases with many constraint switches one may get significant benefits and simplifications with MPC. Other cases where MPC may offer significant benefits compared to ARC is for interactive processes and for cases with known future disturbances.

In conclusion, excellent control performance and close-to optimal economic operation can in most cases be achieved by the use of simple classical ARC elements, but there is a lack of understanding, both in industry and academia, on how such control systems should be designed. The paper offers a new beginning in terms of providing a systematic approach.

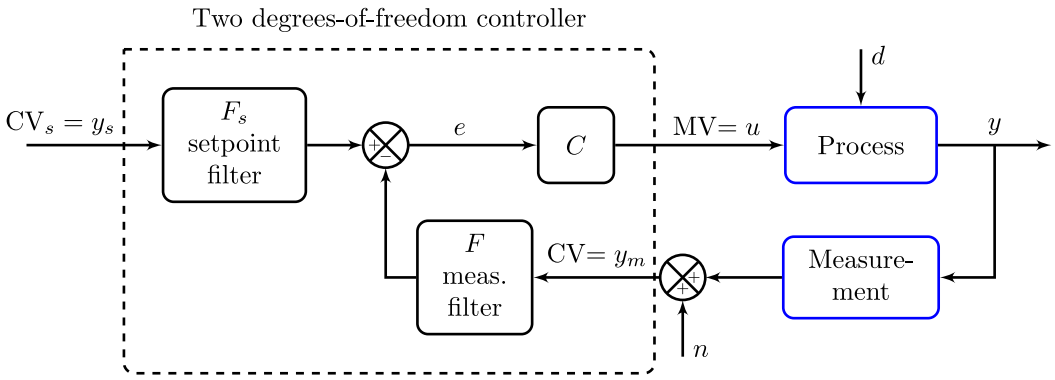


Fig. A.41. Two degrees-of-freedom control system with setpoint filter F_s and measurement filter F. All blocks are possibly nonlinear.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgments

I gratefully acknowledges fruitful discussions and inputs from Krister Forsman, Cristina Zotică, Adriana Reyes-Lúa, Lucas Ferreira Bernardino, Dinesh Krishnamoorthy, Risvan Dirza, Nicholas Alsop, Mark Darby, Ivar J. Halvorsen, Sigifredo Nino, James B. Rawlings, William Tubbs, Evren Turan, John C. Doyle, Jose Luis Guzmán and anonymous reviewers. I am grateful to Cristina Zotică for preparing the figures. Finally, I want to acknowledge Manfred Morari, George Stephanopoulos, Alan Foss and Greg Shinskey for the decisive impact their work and advice has had on my research in the area of control system structure over the last 40 years, since I started my Ph.D. work on process control at Caltech in 1983.

Appendix A. Feedback and feedforward control structures

Fig. 3 shows a simple feedback control system 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 control error $e = y_s - y_m$.

The more general two degrees-of-freedom controller in Fig. 2, 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 setpoint prefilter F_s for y_s is shown in Fig. A.41. Here, we have also added a measurement filter F for y_m. Instead of using a prefilter F_s, an alternative is to add, in parallel to C, a feedforward element C_{Fy} from the setpoint y_s to MV=u (Fig. A.42).

In Figs. 3 and A.41 we have included a measurement block and a measurement error signal (noise) n. Note that the signal n also includes the static measurement error (systematic error, bias). In process control, the measurement block is often represented by a time delay or a first-order process with a steady-state gain of identity. However, in this paper, we usually do not include the measurement block or the measurement noise (n), that is, we assume perfect measurement with $y_m = y$. Of course, this is not correct but it simplifies the block diagrams. In the linear case, the one degree-of-freedom feedback controller in Fig. 3 then becomes (with Laplace transforms and deviation variables)

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

and the two degrees-of-freedom feedback controller in Fig. A.41 becomes

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

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 measurement filter) or a first-order filter,

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

Here τ_F is the measurement filter time constant, and the inverse ($\omega_F = 1/\tau_F$) is known as the cutoff frequency. However, one should be careful about selecting a too large filter time constant τ_F as it acts as a effective delay as seen from the controller C; see also the recommendation $\tau_F \le \tau_c/2$ in (C.17).

King (2011) (page xii) writes in this respect: "Many engineers are guilty of installing excessive filtering to deal with noisy measurements. Often implemented only to make trends look better they introduce additional lag and can have a detrimental impact on controller performance". To reduce the effective delay (lag) introduced by filtering, Sigifredo Nino (personal email communication, 30 March 2023), who has extensive industrial experience, suggests using a second-order Butterworth filter,

$$F(s) = \frac{1}{\tau_F^2 s^2 + 1.414\tau_F s + 1} \tag{A.4}$$

As mentioned, an alternative to using F_s is to use a "feedforward" element C_{Fy}. For the linear case, an alternative to (A.2) is then:

$$u = C(s)(y_s - F(s)y) + C_{Fy}(s)y_s \tag{A.5}$$

Feedforward control is also used for measured disturbances, and a linear feedforward control system (with no feedback, i.e., $C = 0$) is shown in Fig. A.42. Here, we have

$$u = C_{Fd}(s)d + C_{Fy}(s)y_s \tag{A.6}$$

where d is a measured disturbance. The feedforward control system in Fig. A.42 is linear because the independent contributions from d and y_s are added together.

Appendix B. Example: Feedback versus feedforward control for uncertain processes

This example is important for understanding the advantage of being able to directly specify the desired control structure; in this case to use feedback rather than feedforward control to deal with gain uncertainty. Originally, the example was at the beginning of the paper, but it was moved to the Appendix to improve the flow of the paper.

Control makes use of two main principles, namely feedforward and feedback. Most engineers are (indirectly) trained to be "feedforward

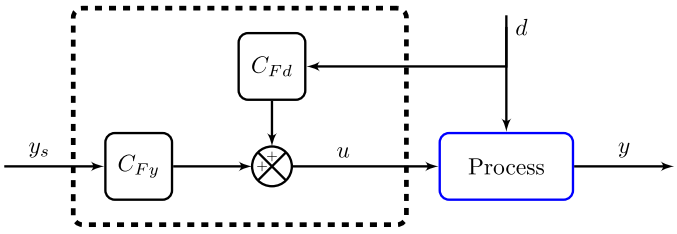


Fig. A.42. Block diagram of feedforward control system with linear combination of feedforward from measured disturbance (d) and setpoint (y_s) (E14).

thinkers" and they immediately think of "model inversion" when it comes to doing control. Thus, they prefer to rely on models instead of data, although feedback solutions in most cases are much simpler and more robust (e.g., Skogestad (2009)). Interestingly, as discussed next, feedforward and feedback solutions may in some cases yield identical nominal performance. However, given a choice, feedback solutions should be preferred because they are much less sensitive to model errors (including nonlinearity). This is illustrated in the following example where the main purpose is to demonstrate the advantage of feedback control, and in particular of integral action, in dealing with model error (uncertainty). A more general treatment is found in Skogestad and Postlethwaite (2005) (pages 203–205).

We consider a linear first-order process with a time constant $\tau = 6$ [in relevant time units; e.g. seconds or minutes] and steady state gain $k = 3$ [again in relevant unit]. The following linear model describes the dynamics:

$$\tau \frac{dy(t)}{dt} = -y(t) + ku(t) \tag{B.1}$$

However, for our purposes the Laplace (s) domain is more convenient, because it transforms differential equations into algebraic equations and makes it possible to derive transfer functions. The most important Laplace property is that derivation is replaced by multiplication with s, that is, the Laplace transform of $dy(t)/dt$ is $sy(s)$. Introducing deviation variables, we may then write (B.1) as $y(s) = G(s)u(s)$, where, independently of what kind of signal $u(t)$ is, the process transfer function is

$$G(s) = \frac{k}{\tau s + 1}, \quad k = 3, \ \tau = 6 \tag{B.2}$$

B.1. Nominal response

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 to "speedup" the original dynamics by a factor $\tau_c/\tau = 6/4 = 1.5$.

Feedforward solution. We use feedforward from the setpoint (Fig. A.42):

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

where we choose

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

The output response becomes as desired,

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

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

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

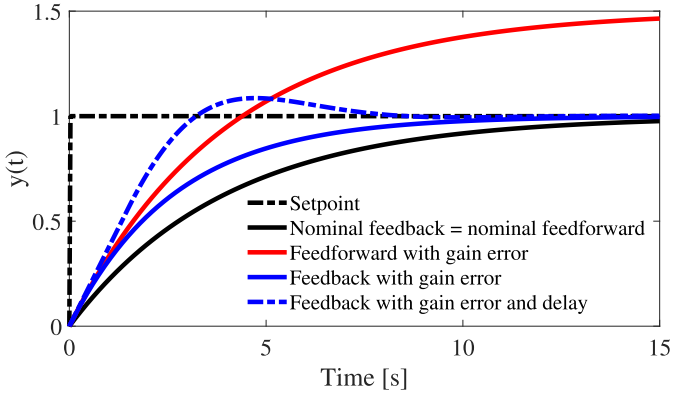


Fig. B.43. Setpoint response for process (B.2) demonstrating the advantage of feedback control for handling model error.

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 Appendix C.3.1):

$$C(s) = K_c \left(1 + \frac{1}{\tau_I s}\right) = 0.5 \frac{6s + 1}{6s} \tag{B.5}$$

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:

$$y = \frac{1}{\tau_c s + 1} y_s = \frac{1}{4s + 1} y_s \tag{B.6}$$

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

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 Fig. B.43).

B.2. Response with process gain change

As illustrated by the simulations in Fig. B.43, the feedback PI-control solution is a lot more robust than feedforward control. Consider an increase in the process gain by 50% (from $k = 3$ to $k' = 4.5$). With the feedforward controller (B.3), 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 (B.4) 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 (B.5), 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 (from $kK_c = 1.5$ to $k'K_C = 2.25$) may result in instability, in particular, 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 Fig. B.43, which shows that even by adding a measurement delay $\theta = 1.5$, the response with PI-control is still good. We see that some oscillations are beginning to appear, but the closed-loop system is still far from instability.[10] Note that instability cannot occur

[10] For the perturbed case (with $k' = 4.5$), a more detailed analysis using the Bode stability condition gives that the delay margin is DM $= \frac{\text{PM[rad]}}{\omega_c\text{[rad/s]}} = 4.19$s,

with feedforward control, at least not in the linear case, so this is an advantage of feedforward control.

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 feedforward control for cases where feedback control is a much better solution. The latter is relevant for some controller design methods, for example, model predictive control (MPC).

Appendix C. Basic single-variable feedback control

C.1. The PID controller

There exists many variants and parameterizations of the PID controller. The most common is the "ideal-form" PID controller given by

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

In the Laplace domain we get $u(s) = C(s)e(s) + u_0$ where

$$C(s) = K_c\left(1 + \tau_D s + \frac{1}{\tau_I s}\right) \tag{C.2}$$

Here u is the MV, $e = y_s - y$ is the setpoint error and y is the measured CV-value. This a one degree-of-freedom controller, since the controller only acts on the error e, see Fig. 3.

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

$$b = u_I + u_0 \tag{C.3}$$

With integral action, the value of u_0 only matters initially, when the controller is turned on or reactivated, because later the contribution u_I from the integral action 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 = controller gain

τ_I = integral time [s, min]

τ_D = derivative time [s, min]

To avoid a derivative "kick" for setpoint changes, it is common to *not* use derivative action on the setpoint (Fig. 7). Fig. 7 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 the PI-controller then has 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, for process control applications, it is recommended that the tuning is based on a first-order plus delay model (C.9), obtained from an experiment that excites the process, for example, a step response; see next.

where in this simple case with $\tau_I = \tau$, the phase margin is PM= 90° = 1.57 rad and the gain crossover frequency is $\omega_c = \frac{k' K_c}{\tau_I} = \frac{4.5 \cdot 0.5}{6} = 0.375$ rad/s. Thus, the system remains stable as long as the delay is less than $\theta = 4.19$s.

C.1.1. Discrete PID controller

A discrete approximation of (C.1) for practical implementation is given by

$$u_{I,k} = u_{I,k-1} + \frac{K_c \Delta t_s}{\tau_I} e_k \tag{C.4a}$$

$$u_k = K_c e_k + K_c \tau_D \frac{e_k - e_{k-1}}{\Delta t_s} + u_{I,k} + u_0 \tag{C.4b}$$

Here, Δt_s is the sampling time, which in process control applications often is 1 s, but for control purposes it should be as small as possible to reduce the effective delay. The effect of measurement noise on the derivative part may be handled by filtering the measurement. For example, the first-order filter in (A.3) can be approximated as

$$y_k = \alpha y_{m,k} + (1-\alpha) y_{k-1}, \quad \text{where } \alpha = \frac{1}{1 + \tau_F/\Delta t_s} \tag{C.5}$$

(this is known as the "exponentially moving average" in time series analysis). We then have $e_k = y_{s,k} - y_k$. If we do not want derivative action on the setpoint, then $e_k - e_{k-1}$ in (C.4b) is replaced by $y_{k-1} - y_k$.

C.2. PID tuning by direct synthesis or IMC

Design (tuning) rules for the PID controller were proposed more than 80 years ago by Ziegler and Nichols (1942), and these remained the dominant rules for the next 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; see Section 7.4) popularity of the Smith Predictor. The only other set of PID tuning rules that were available until about 1985, were the Cohen and 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. Eventually, in the 1980s academic researchers started showing interest in PID control. Åström and Hägglund (1988) considered the implementation of PID controllers and recommended the anti-windup scheme shown in Fig. 7.

For PID tuning, Rivera et al. (1986) proposed the Internal Model Control (IMC) rules and Smith and Corripio (1985) proposed their similar "direct synthesis" rules. The IMC and "direct synthesis" rules are both based on specifying the desired closed-loop response. It is not possible to eliminate a process time delay (θ) from the closed loop, so a typical setpoint specification is a first-order plus delay response, which in the Laplace domain may be written as

$$y(s) = T(s) y_s(s), \quad \text{where} \quad T(s) = \frac{e^{-\theta s}}{\tau_c s + 1} \tag{C.6}$$

Here, τ_c is the desired closed-loop time constant, which is the most important design parameter.

In the time domain, for a step setpoint change y_s occurring at $t = 0$, this corresponds to

$$y(t-\theta) = (1 - e^{-t/\tau_c})\, y_s \tag{C.7}$$

For a linear system, we have that

$$T(s) = \frac{GC}{1 + GC} \tag{C.8}$$

(see Fig. 3 with Process = $G(s)$ and Measurement = 1). From this one can with a given process model $G(s)$, find algebraically the corresponding controller C, which turns out to be a Smith Predictor controller. To obtain a fixed-order controller, we approximate the time delay in the Smith Predictor controller, e.g., using $e^{-\theta s} \approx 1 - \theta s$. For a first- or second-order process G, this gives a PI or PID controller , respectively (Skogestad, 2003; Smith & Corripio, 1985). 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, τ_c, which is the desired closed-loop response time. 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 (C.7)). In some papers τ_c is called λ, ans these direct synthesis (IMC) rules became very popular in the pulp & paper and mining industries 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.

C.3. SIMC PID controller

C.3.1. Derivation of SIMC PI-rule

Let us derive the SIMC PI-rule. The starting point 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):

$$G(s) = \frac{k}{\tau s + 1} e^{-\theta s} \tag{C.9}$$

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 three model parameters are

$$k = \text{steady-state gain} = \frac{\Delta \text{CV}}{\Delta \text{MV}} \tag{C.10a}$$

$$\tau = \text{first-order process time constant (63\%)} \tag{C.10b}$$

$$\theta = \text{effective time delay} \tag{C.10c}$$

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

Combining (C.6), (C.8) and (C.9), solving for $C(s)$ and using the approximation $e^{-\theta s} \approx 1 - \theta s$, results in a PI-controller with $K_c = \frac{1}{k}\frac{\tau}{\tau_c+\theta}$ and $\tau_I = \tau$ (Skogestad, 2003). This works well for setpoint changes and disturbances at the process output. However, with $\tau_I = \tau$, we essentially turn off the integral action for slow or integrating processes with a large τ. To get acceptable rejection of disturbances entering at the process input (which are very common), we need to reduce the integral time τ_I for such processes. We should not reduce it too much because otherwise we get "slow" oscillations caused by having two integrators in series (one from the process and one from the controller). We select the minimum value as $\tau_I = 4(\tau_c + \theta)$, which is the smallest τ_I that avoids the complex poles associated with the undesirable slow oscillations. It is also useful to introduce[11]

$$k' = \frac{k}{\tau} = \text{initial slope of step response} \tag{C.11}$$

C.3.2. SIMC PI-rule

The final SIMC PI-rule for a first-order plus delay process (C.9) then becomes (Skogestad, 2003):

$$K_c = \frac{1}{k'}\frac{1}{\tau_c + \theta} \tag{C.12a}$$

$$\tau_I = \min\left(\tau, 4(\tau_c + \theta)\right) \tag{C.12b}$$

Let us look at two limiting cases for the process time constant τ. For an integrating with delay process, $G(s) = \frac{k'}{s}e^{-\theta s}$, we have $\tau = \infty$, and the SIMC-rule gives a PI-controller with integral time $\tau_I = 4(\tau_c + \theta)$. For integrating processes, it is common in industrial practice to use too much integral action (choose too small τ_I) which results in the "slow" oscillations just mentioned. If the process starts cycling, the intuition of the operator is to reduce K_c. However, from the relationship $K_c\tau_I = 4/k'$ (which follows from Eq. (C.12) and avoids slow oscillations for integrating processes), this is exactly the opposite of what the operator should do. The result is that the process cycles even more, and the operator gives up and leaves the process cycling.

For a static process ($\tau = 0$) with delay, $G(s) = ke^{-\theta s}$, the SIMC-rule gives a pure I-controller ($C(s) = K_I/s$ or $u(t) = K_I\int_0^t e(t)dt$ in the time domain) 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.

For intermediate values of process time constant τ, the recommendation is to select $\tau_I = \tau$. This is the "lambda tuning rule" and generally works well for setpoint responses. However, the modification in Eq. (C.12b) is needed to handle input ("load") disturbances for processes with a large τ.

C.3.3. Choice of tuning parameter τ_c

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

$$\tau_c \geq \theta \tag{C.13}$$

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 practical 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 (C.12) this gives $K_c = (1/3)(6/4) = 0.5$ and $\tau_I = \min(6, 16) = 6$, as used earlier in (B.5).

C.3.4. Gain margin for SIMC rule

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 (C.12b), we have an exact linear relationship (Grimholt & Skogestad, 2012):

$$\text{GM} = \frac{\pi}{2}\left(\frac{\tau_c}{\theta} + 1\right) \tag{C.14}$$

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 (C.12b)), 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 (C.14) for all values of τ_c/θ. Similar linear relationships apply to the delay margin (Grimholt & Skogestad, 2012).

C.3.5. Derivative action

Derivative action is normally only recommended for dominant second-order processes (defined as processes for which $\tau_2 \geq \theta$ (Skogestad, 2003)), where the SIMC-rule gives [12]

$$\hat{\tau}_D = \tau_2 \tag{C.15}$$

[11] Note that k' is used in two different meanings in this paper, so the slope gain k' in (C.11) should not confused with the perturbed gain k' in Appendix B.2.

[12] The hat on τ_2 is used because this is for the series-form PID, $C(s) = \hat{K}_c\left(1 + \frac{1}{\hat{\tau}_I s}\right)(\hat{\tau}_D s + 1)$. With D-action, the values for K_c and τ_I in Eq. (C.12)) are actually for the series-form PID (and should have hats). With D-action, all three controller parameters need to be modified by the factor $f = \left(1 + \frac{\hat{\tau}_D}{\hat{\tau}_I}\right)$ when going from the series form to the the "ideal" form in (C.2): $K_c = \hat{K}_c f, \tau_I = \hat{\tau}_I f, \tau_D = \hat{\tau}_D/f$ (Skogestad, 2003).

However, there is an exception. If it is important with very tight control for a first-order plus delay process (C.9), then one may use the "improved" SIMC PID-rule and add derivative action with

$$\hat{\tau}_D = \theta/3 \tag{C.16}$$

(again, 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 (see also Section 7.4). 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.

C.3.6. Measurement filter

For noisy processes, one may add a filter F on the measurement of y (Fig. A.41), for example, a first-order filter (A.3) (discrete (C.5)) or a Butterworth filter (A.4) with a tuneable time constant τ_F. To avoid that the filter adds too much lag to the control loop, one should choose

$$\tau_F \le \tau_c/2 \tag{C.17}$$

Preferably, an even smaller value should be chosen.

C.4. Comment on industrial PID implementations

Note from Eq. (C.12a) that the controller gain K_c should have the same sign as the process gain (k, k'). This means that K_c should be negative if the process gain is negative. However, most commercial control systems only allow for positive controller gains and then instead distinguish between "direct" and "reverse"' control action. Commercial process control vendors use the following definition:

- "Reverse acting" control is used in the "normal" case when the process gain is positive (because the MV (u) should then be reduced when the CV (y) is too high).
- "Direct acting"' control is used when the process gain is negative.

Also note that process control vendors may use different parametertizations of the PID-controller. For example, it is common to use the integral gain $K_I = K_c/\tau_I$ instead of the integral time τ_I. Previously (before about 1980), the use of proporotional band $= 100/K_c$ and reset rate $= 1/\tau_I$ was common. Finally, note that most industrial control systems work with scaled variables, typically 0% to 100%. Indeed, this is where the number 100 comes from in the definition of proportional band. The use of scaled variables has many advantages, including allowing for the use of default tunings, which is particularly useful during startup.

C.5. Anti-windup (E8)

In the following let u denote the controller output (MV). "Windup" is when the integrator term u_I in (C.1) 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) have no effect on the controlled variable (CV), usually because the controller output u is not equal to the actual (physical) input ($\tilde{u}$) (Fig. 7). The most common reason is saturation in the final control element (actuator) (which is usually a valve in process control), but it could also be because of a selector or user-set limits on the controller output.

C.5.1. Simple anti-windup schemes

Many industrial anti-windup schemes exist. The simplest is to limit u in (C.1) to be within specified bounds (by updating u_0), or to limit the bias $b = u_0 + u_I$ to be within specified bounds (also by updating 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 simple anti-windup approaches suffice (Smith, 2010) (page 21).

C.5.2. Anti-windup using external reset

A better and also common anti-windup scheme is "external reset" (e.g., Wade (2004) Smith (2010)) which originates from Shinskey. This scheme 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.[13] With this implementation, anti-windup is easily achieved by replacing the positive feedback from u with the actual applied value ($\tilde{u}$).

C.5.3. Recommended: Anti-windup with tracking

The "external reset" solution is a special case of the further improved "tracking" scheme in Fig. 7 which is recommended by Åström and Hägglund (1988). The tracking scheme (sometimes referred to as the "back-calculation" scheme (Åström & Hägglund, 2006)) 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 "tracking" scheme, note that we for a one degree-of-freedom PID controller have (see also Fig. 7)

$$u(t) = K_c e(t) + K_c \tau_D \frac{de(t)}{dt} + \overbrace{\underbrace{\int_{\hat{t}=t_0}^{t} \left(\frac{K_c}{\tau_I} e(\hat{t}) + \frac{1}{\tau_T} e_T(\hat{t}) \right) d\hat{t} + u_0}_{\text{bias}=b}}^{u_I(t)} \tag{C.18}$$

The tracking error

$$e_T(t) = \tilde{u} - u \tag{C.19}$$

is fed to the input of the integrator through the gain $1/\tau_T$. This error 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 created 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 saturation is no longer active. The disadvantage with a too small tracking time is that it may activate the controller unnecessary.

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

$$e_T = u - u_{lim} = K_c \frac{\tau_T}{\tau_I} e \tag{C.20}$$

Note that $e = y_s - y$ is nonzero (and out of our control) when u is saturated (or more generally, disconnected from the process). We see from (C.20) that a small τ_T means that tracking error e_T is smaller, with u (computed by the controller) 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 desirable because it may activate the controller when it is not necessary, because the proportional and derivative terms will always cause some "nervous" variations in $u(t)$ due to disturbances and measurement noise. 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), (C.18) gives $u = K_c e + b$ which combined with (C.20) and $\tau_T = \tau_I$ gives $b = u_{lim}$. For a PI-controller, (C.18) gives

[13] Note that with positive feedback we have $\frac{1}{1-\frac{1}{\tau_I s+1}} = \frac{\tau_I s+1}{\tau_I s} = 1 + \frac{1}{\tau_I s}$.

$u(t) = K_c e(t) + b$ (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 and 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.

C.5.4. 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 recommended tracking scheme in Fig. 7.

However, when switching from manual to automatic control, we may get a "bump". This may happen even with anti-windup using tracking, 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 (C.18) 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.

C.5.5. Velocity form

An alternative to the normal "position form" PID controller in (C.1) is the "velocity form" where the controller computes the MV change $\Delta u(t)$ ($\Delta u_k = u_k - u_{k-1}$ in discrete form), rather than $u(t)$. The velocity form inherently contains anti-windup (although it does not have a tuning parameter like τ_T) and bumpless transfer. However, a major disadvantage with the velocity form is that the integral mode *must* be included, for example, it cannot be used as a P-controller. For this reason, the position form in (C.1) and (C.18) is recommended.

C.6. 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 also fairly 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 include 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 Fig. 38.

References

Allison, B. J., & Isaksson, A. J. (1998). Design and performance of mid-ranging controllers. *Journal of Process Control*, *8*(5–6), 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*(1), 3–16.

Alstad, V., Skogestad, S., & Hori, E. (2009). Optimal measurement combinations as controlled variables. *Journal of Process Control*, *19*(1), 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. *Industrial and Engineering Chemistry Research*, *48*, 10892–10902.

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.

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*, (November), 10–15.

Bernardino, L. F., Krishnamoorthy, D., & Skogestad, S. (2022). Optimal operation of heat exchanger networks with changing active constraint regions. In *PSE Symposium in: Computer aided chemical engineering (Elsevier), vol. 49* (pp. 421–426).

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

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

Bode, H. (1945). *Network analysis and feedback amplifier design*. New York: Van Nostrand.

Bristol, E. (1966). On a new measure of interactions for multivariable process control. *IEEE Transactions on Automatic Control*, *AC-11*(1), 133–134.

Buckley, P. C. (1964). *Techniques of process control*. Wiley.

Cao, Y. (2004). Constrained self-optimizing control via differentiation. In *Adchem synposium, Hong Kong, Jan. 2004, IFAC Proceedings Volumes 37 no. 1* (pp. 63–70).

Carlson, J. M., & Doyle, J. C. (1999). Highly optimized tolerance: A mechanism for power laws in designed systems. *Physical Review E*, *60*(2), 1412–1427.

Chiang, M., Low, S. H., Calderbank, A. R., & Doyle, J. C. (2007). Layering as optimization decomposition: A mathematical theory of network architectures. *Proceedings of the IEEE*, *95*(1), 255–312. http://dx.doi.org/10.1109/JPROC.2006.887322.

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

Cutler, C., & Ramaker, B. (1980). Dynamic matrix control - A computer control algorithm. In *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*(3), 706–711.

Downs, J. J., & Skogestad, S. (2011). An industrial and academic perspective on plantwide control. *Annual Reviews in Control*, *35*(1), 99–110.

Doyle, J. (1978). Guaranteed margins for LQG regulators. *IEEE Transactions on Automatic Control*, *AC-23*(4), 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. In *Proceedings of the national academy of sciences of the United States of America (PNAS), vol. 102 no. 41* (pp. 14497–14502).

Eaton, J. W., & Rawlings, J. B. (1992). Model-predictive control of chemical processes. *Chemical Engineering Science*, *47*(4), 705–720.

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

Faanes, A., & Skogestad, S. (2003). Buffer tank design for acceptable control performance. *Industrial and Engineering Chemistry Research*, *42*, 2198–2208.

Forsman, K. (2016). Implementation of advanced control in the process industry without the use of MPC. *DYCOPS conference, Trondheim, IFAC papers online*, *49*(7), 514–519.

Foss, A. (1973). Critique of chemical process control theory. *AIChE Journal*, *19*(2), 209–214.

Freudenberg, J., & Middleton, R. (1999). Properties of single input, two output feedback systems. *International Journal of Control*, *72*(16), 1446–1465.

Grebe, I., Boundy, R., & Cermak, R. (1933). The control of chemical processes. *Trans. Am. Inst. Chem. Eng.*, *29*, 211–256.

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*(3), 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. *Journal of Process Control*, *70*, 36–46.

Grimholt, C., & Skogestad, S. (2018b). Should we forget the smith predictor? In *3rd IFAC Conference on Advances in PID Control, Ghent, Belgium, IFAC Papers Online, vol. 51 no. 4* (pp. 769–774).

Grosdidier, P., Morari, M., & Holt, B. R. (1985). Closed-loop properties from steady-state gain information. *Industrial and Engineering Chemistry, Fundamentals*, *24*(2), 221–235.

Guzmán, J., & Hägglund, T. (2011). Simple tuning rules for feedforward compensators. *Journal of Process Control*, *21*, 92–102.

Guzmán, J., & Hägglund, T. (2021). Tuning rules for feedforward control from measurable disturbances combined with PID control: A review. *International Journal of Control*.

Hägglund, T. (2021). A feedforward approach to mid-ranging control. *Control Engineering Practice*, *108*, Article 104713.

Hägglund, T., & Guzmán, J. (2018). Development of basic process control structures. *IFAC Papers Online*, *51*(45), 775–780.

Ingimundarson, A., & Hägglund, T. (2002). Performance comparison between PID and dead-time compensating controllers. *Journal of Process Control*, *12*(8), 887–895.

Jacobsen, M. G., & Skogestad, S. (2011). Active constraint regions for otimal operation of chemical processes. *Industrial and Engineering Chemistry Research*, *50*, 11226–11236.

Jäschke, J., Cao, Y., & Kariwala, V. (2017). Self-optimizing control–A survey. *Annual Reviews in Control*, *43*, 199–223.
King, M. (2011). *Process control: A practical approach*. Wiley.
Krishnamoorthy, D., & Skogestad, S. (2020). Systematic design of active constraint switching using selectors. *Computers and Chemical Engineering*, *143*, Article 107106.
Krishnamoorthy, D., & Skogestad, S. (2022). Real-time optimization as a feedback control problem – A review. *Computers and Chemical Engineering*, *161*, Article 107723.
Kumar, P., Rawlings, J. B., & Carrette, P. (2023). Modeling proportional–integral controllers in tracking and economic model predictive control. *Journal of Process Control*, *122*, 1–12.
Leal, M., Hoyo, Á., Guzmán, J. L., & Hägglund, T. (2021). Double back-calculation approach to deal with input saturation in cascade control problems. In J. A. Gonçalves, M. Braz-César, & J. a. P. Coelho (Eds.), *CONTROLO 2020* (pp. 200–209). Cham: Springer International Publishing.
Lindholm, A., Forsman, K., & Johnsson, C. (2010). A general method for defining and structuring buffer management problems. In *Proceedings of 2010 American control conference* (pp. 4397–4402).
Liptak, B. (Ed.), (1973). *Instrumentation in the process industries*. Chilton Book Co..
Liptak, B. G. (1999). *Optimization of industrial unit processes, 2nd edition* (Second). CRC Press.
Lundström, P., Lee, J., Morari, M., & Skogestad, S. (1995). Limitations of dynamic matrix control. *Computers & Chemical Engineering*, *19*(4), 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. In *Proceedings 20th European Conference on Modelling and Simulation*.
McAvoy, T. J. (1983). *Interaction analysis - principles and applications*. Instrument Society of America (ISA).
Minasidis, V., Skogestad, S., & Kaistha, N. (2015). Simple rules for economic plantwide control. *Computer Aided Chemical Engineering (from PSE/ESCAPE Symposium)*, *37*, 101–108.
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*(2), 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*(2), 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*(2), 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*(4), 1178–1189.
Pawlowski, A., Guzmán, J., Normey-Rico, J., & Berenguel, M. (2012). Improving feedforward disturbance compensation capabilities in generalized predictive control. *Journal of Process Control*, *22*, 527–539.
Price, R. M., Lyman, P., & Georgakis, C. (1994). Throughput manipulation in plantwide control structures. *Industrial and Engineering Chemistry Research*, *33*, 1197–1207.
Rawlings, J. B. (2000). Tutorial overview of model predictive control. *IEEE Control Systems Magazine*, *20*(3), 38–52.
Reyes-Lúa, A., & Skogestad, S. (2019). Systematic design of split range controllers. *Processes*, *7*(12), 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. *Industrial and Engineering Chemistry Research*, *59*(6), 2229–2241.
Reyes-Lúa, A., Zotică, C., Forsman, K., & Skogestad, S. (2019). Systematic design of split range controllers. *DYCOPS conference, IFAC-PapersOnLine*, *52*(1), 893–903.
Richalet, A., Testud, J., & Papon, J. (1978). Model predictive heuristic control: Apllications to industrial processes. *Automatica*, *14*, 413–428.
Ricker, N. L. (1996). Decentralized control of the Tennessee Eastman challenge process. *Journal of Process Control*, *4*, 205–221.
Rivera, D., Morari, M., & Skogestad, S. (1986). Internal model control. 4. PID controller design. *Industrial and Engineering Chemistry Research*, *25*(1), 252–265.
Rosenbrock, 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.
Samad, T., Bauer, M., Bortoff, S., Cairano, S. D., Fagiano, L., Odgaard, P. F., Rhinehart, R. R., Sánchez-Peña, R., Serbezov, A., Ankersen, F., Goupil, P., Grosman, B., Heertjes, M., Mareels, I., & Sosseh, R. (2020). Industry engagement with control research: Perspective and messages. *Annual Reviews in Control*, *49*, 1–14.
Seborg, D. E., Edgar, T. F., Mellichamp, D. A., & III, F. J. D. (2016). *Process dynamics and control* (Fourth). 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). 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. *Industrial and Engineering Chemistry Research*, *30*(4), 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*(4), 291–309.
Skogestad, S. (2004a). Control structure design for complete chemical plant. *Computers and Chemical Engineering*, *28*, 219–234.
Skogestad, S. (2004b). Near-optimal operation by self-optimizing control: from process control to marathon running and business systems. *Computers and Chemical Engineering*, *29*, 127–137.
Skogestad, S. (2009). Feedback: Still the simplest and best solution. *Modeling, Identification and Control*, *30*(3), 149–155.
Skogestad, S. (2015). Control structure selection. In *Encyclopedia of systems and control* (pp. 202–215). Springer.
Skogestad, S. (2023). The theoretical basis of ratio control. *Publication in progress*.
Skogestad, S., & Postlethwaite, I. (1996). *Multivariable feedback control: analysis and design* (1st). Wiley.
Skogestad, S., & Postlethwaite, I. (2005). *Multivariable Feedback Control: Analysis and Design* (Second). Wiley.
Skogestad, S., Zotică, C., & Alsop, N. (2023). Transformed inputs for linearization, decoupling and feedforward controller. *Journal of process Control*, *122*, 113–133.
Smith, O. (1957). Closed control of loops with dead time. *Chemical Engineering Progress*, *53*, 217–219.
Smith, C. L. (2010). *Advanced process control - beyond single-loop control*. New York: Wiley.
Smith, C. A., & Corripio, A. (1985). *Principles and practice of automatic process control*. New York: Wiley.
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*(2), 105–114.
Storkaas, E., & Skogestad, S. (2004). Cascade control of unstable systems with application to stabilization of slug flow. In *Adchem synposium, Hong Kong, Jan. 2004, IFAC Proceedings Volumes 37 no. 1* (pp. 335–340).
Wade, H. L. (1997). Inverted decoupling: A neglected technique. *ISA Transactions*, *36*(1), 3–10.
Wade, H. L. (2004). *Basic and advanced regulatory control: system design and application* (2nd). ISA.
Young, A. (1955). *An introduction to process control system design*. Longmans.
Ziegler, J. G., & Nichols, N. B. (1942). Optimum settings for automatic controllers. *Transactions of the ASME*, *64*, 759–768.
Zotică, C., Forsman, K., & Skogestad, S. (2022). Bidirectional inventory control with optimal use of intermediate storage. *Computers and Chemical Engineering*, *159*, Article 107677.