Putting Optimization into process control

Sigurd Skogestad

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

Bratislava
03 May 2024
Putting Optimization into process control. Abstract

- How can you control a complex plant effectively using simple elements with a minimal amount of modelling?
- How can you put optimization into the control layer?

Industry has been using simple and effective as “advanced regulatory control” (ARC) schemes based on PID controllers for almost 100 years. The objective of my work is to provide a systematic approach for designing such control systems.

- The main competitor to ARC is MPC (model predictive control), but the costs of implementing and maintaining MPC solutions are high. Moreover, in most cases ARC solutions (including cascade, ratio, split range and selector control) are more flexible and easier to tune. The main problem right now is that the knowledge and competence about ARC strategies is very low, especially in academia, but also in industry the knowledge is dying out. The result is that people turn off good ARC applications, simply because they don’t understand what they are doing.

- The reason for the lack of training and knowledge is that there has a been belief in academia since the 1980s, that ARC solutions (and PID control) are old-fashioned and will soon be replaced by MPC. However, MPC has now been around for 50 years, and yet the use of MPC is far from increasing as expected. The latest hype is that, if MPC is too complex, then machine learning is the solution. No, it is not, because or the lack of rich data (with sufficiently large input excitations) in most control applications, in particular in process control.

In summary, there is a need to change the mindset of people, both in academia and industry, People need to realize that ARC solutions should be a central part of the future. MPC of course has its place, but mainly as an improvement for large-scale applications that can afford the effort.

The talk will emphasize the above points and in addition present a systematic approach to ARC methods based on my recent paper (which is open access).


Sigurd Skogestad is a Professor in chemical engineering at the Norwegian University of Science and Technology (NTNU) in Trondheim. He received his PhD from Caltrech in 1987 and he is the principal author together with Ian Postlethwaite of the book “Multivariable feedback control” published by Wiley in 1996 (first edition) and 2005 (second edition). The goal of his research is to develop simple yet rigorous methods to solve problems of engineering significance. Research interests include the use of feedback as a tool to (1) reduce uncertainty (including robust control), (2) change the system dynamics (including stabilization), and (3) generally make systems more well-behaved (including self-optimizing control). Other interests include limitations on performance in linear systems, control structure design and plantwide control, interactions between process design and control, and distillation column design, control and dynamics. His other main interests are mountain skiing (cross country), orienteering (running around with a map) and grouse hunting.
Midnight or midday?
About Sigurd Skogestad

- 1955: Born in Flekkefjord, Norway
- 1956-1961: Lived in South Africa
- 1974-1978: MS (Siv.ing.) studies in chemical engineering at NTNU
- 1979-1983: Worked at Norsk Hydro co. (process simulation)
- 1987-present: Professor of chemical engineering at NTNU
- 1994-95: Visiting Professor UC Berkeley
- 2001-02: Visiting Professor UC Santa Barbara
- 1999-2009: Head of ChE Department, NTNU
- 2015-...: Director SUBPRO (Subsea research center at NTNU)

Non-professional interests:
- mountain skiing (cross country)
- orienteering (running around with a map)
- grouse hunting
“The goal of my research is to develop simple yet rigorous methods to solve problems of engineering significance.”
EXPERIENCE IN NORSK HYDRO WITH CUBIC EQUATIONS OF STATE

SIGURD SKOGESTAD
Norsk Hydro, Research Centre, N3901 Porsgrunn, Norway

ABSTRACT

The paper presents some specific applications of cubic equations of state (EOS) in Norsk Hydro and points out some aspects of such equations that one should be aware of when using them or when developing new equations. It is emphasized that the use of EOS to calculate vapor-liquid equilibrium is inherently empirical. Activity coefficients predicted for some systems by the Soave-Redlich-Kwong (SRK) equation of state are presented. The limitations of the van Laar equation for activity coefficients which may be derived from SRK at infinite pressures does not necessarily apply at finite pressures. The shortcomings of the SRK equations of state are pointed out and suggestions are given on how to develop an extended SRK-equation.
1983-87: Caltech

1. Robust control
2. Distillation
3. PID (IMC)

STUDIES ON ROBUST CONTROL OF DISTILLATION COLUMNS

Thesis by
Sigurd Skogestad
California Institute of Technology
Pasadena, California
1987
(Submitted January 26, 1987)
October 1984
Robust control takes off!
Robust Control of Ill-Conditioned Plants: High-Purity Distillation

SIGURD SKOGESTAD, MANFRED MORARI, MEMBER, IEEE, AND JOHN C. DOYLE

Fig. 13. LV-configuration. $\mu$-plots for $\mu$-optimal controller $C_\mu(s)$. 
Robust control

Berkeley, Dec. 1994

1996

2005
At home doing moonshine distillation (1979)
Internal Model Control. 4. PID Controller Design

Daniel E. Rivera, Manfred Morari,* and Sigurd Skogestad

Chemical Engineering, 206-41, California Institute of Technology, Pasadena, California 91125

For a large number of single input–single output (SISO) models typically used in the process industries, the Internal Model Control (IMC) design procedure is shown to lead to PID controllers, occasionally augmented with a first-order lag. These PID controllers have as their only tuning parameter the closed-loop time constant or, equivalently, the closed-loop bandwidth. On-line adjustments are therefore much simpler than for general PID controllers. As a special case, PI- and PID-tuning rules for systems modeled by a first-order lag with dead time are derived analytically. The superiority of these rules in terms of both closed-loop performance and robustness is demonstrated.

\[ g(s) = \frac{k}{(\tau_1 s + 1)(\tau_2 s + 1)} e^{-\theta s} \]

Tuning parameters:
\[ K_c = \frac{1}{k(\tau_c + \theta)} \]
\[ \tau_1 = \min\{\tau_1, 4(\tau_c + \theta)\} \]
\[ \tau_D = \tau_2 \]

\[ \tau_c \geq \theta \]


Abstract
The aim of this paper is to present analytic rules for PID controller tuning that are simple and still result in good closed-loop behavior. The starting point has been the IMC-PID tuning rules that have achieved widespread industrial acceptance. The rule for the integral term has been modified to improve disturbance rejection for integrating processes. Furthermore, rather than deriving separate rules for each transfer function model, there is a just a single tuning rule for a first-order or second-order time delay model. Simple analytic rules for model reduction are presented to obtain a model in this form, including the “half rule” for obtaining the effective time delay.

*SIMC = Simple/Skogestad IMC
Chemical Engineering
Process control: Hierarchical decision system based on time scale separation

The central issue to be resolved ... is the determination of control system structure. Which variables should be measured, which inputs should be manipulated and which links should be made between the two sets? There is more than a suspicion that the work of a genius is needed here, for without it the control configuration problem will likely remain in a primitive, hazily stated and wholly unmanageable form. The gap is present indeed, but contrary to the views of many, it is the theoretician who must close it.
The central issue to be resolved ... is the determination of control system structure. Which variables should be measured, which inputs should be manipulated and which links should be made between the two sets? There is more than a suspicion that the work of a genius is needed here, for without it the control configuration problem will likely remain in a primitive, hazily stated and wholly unmanageable form. The gap is present indeed, but contrary to the views of many, it is the theoretician who must close it.
Well, I’m not a genius, but I didn’t give up. I started on this in 1983. 40 years later:
How we design a control system for a complete chemical plant?

• Where do we start?
• What should we control? and why?
• etc.
• etc.
Optimal steady-state operation (economics)

• Typical cost function*:

\[ J [$/s] = \text{cost feed} + \text{cost energy} - \text{value products} \]

*No need to include fixed costs (capital costs, operators, maintainance) at “our” time scale (hours)

Note: J=-P where P= Operational profit
Subject to Constraints:

Purity D: For example, \( x_{D, \text{impurity}} \leq \text{max} \)

Purity B: For example, \( x_{B, \text{impurity}} \leq \text{max} \)

Flow constraints: \( \text{min} \leq D, B, L \text{ etc.} \leq \text{max} \)

Column capacity (flooding): \( V \leq V_{\text{max}}, \text{etc.} \)

Cost \( J [\$s] \) to be minimized (economics):

\[
J = -P \quad \text{where} \quad P = p_D D + p_B B - p_F F - p_V V
\]

Optimal operation: Minimize \( J \) with respect to steady-state degrees of freedoms (inputs \( u \))

\( u = [\text{reflux } L; \text{ heat input } V] \)
Process control: Hierarchical decision system

**Objective**

- **Manager**
  - Scheduling (weeks)
  - Site-wide optimization (day)
  - Local optimization (hour)
  - Supervisory control (minutes)
  - Regulatory control (seconds)

- **Process engineer**
  - CV1s
  - CV2s

- **Operator/RTO (usually steady-state)**
  - Supervisory control layer
    - "Advanced control"/MPC
  - Regulatory (basic) control layer
    - PID-control

- **Gain margin**

**Responsible**

- **Min J (economics)**
  - hour
- **Setpoint control**
  - (y - y_s)^2 + Δu^2
  - minute
- **Stabilize + avoid drift**
  - Gain margin...
  - second

**CV = controlled variable (with setpoint)**
**Process control: Hierarchical decision system**

**Objective**

- Minimize $J$ (economics)
  - **Manager**
  - **Process engineer**

- **Site-wide optimization**
  - (day)

- **Economic MPC (EMPC)**
  - or
  - **Dynamic RTO**

- $J = J_{econ} + \Delta u^2$

- Stabilize + avoid drift
  - (look after other variables)

- $J_c = (y - y_s)^2 + \Delta u^2$

- Gain margin...

- $u$ (MV) = valves

- (weeks)

- (day)
Process control: Hierarchical decision system

**Objective**

- **Manager**
  - Scheduling (weeks)
  - Site-wide optimization (day)

- **Process engineer**
  - Economic MPC (EMPC)
  - Dynamic RTO

**Economic MPC (EMPC)**

- **Objective**
  - Minimize $J$ (economics)
  - $J = (y - y_s)^2 + \Delta u^2$

**Setpoint control**

- (+ look after other variables)

**Stabilize + avoid drift**

- Gain margin...
  - Minute

- Second

**Control variables**

- $u (MV) = valves$

- CV2s

- Regulatory control (seconds)

- Layer
Process control: Hierarchical decision system

u (MV) = valves
• **Real-time optimization layer (RTO):**
  - Optimize setpoints CV1s based on detailed nonlinear model (usually steady state)

• **Supervisory/"Advanced” control:**
  - Follow set points for CV1
  - Switch between active constraints (change CV1)
  - Look after regulatory layer (avoid that MVs saturate, etc.)

**Implementation:**
Alternative 1: “Advanced PID” (ARC) based on “simple elements”
Alternative 2: MPC (model predictive control)

• **Regulatory control (PID):**
  - Stable operation (CV2)

**CV = controlled variable.**  **MV = manipulated variable.**  **ARC = Advanced regulatory (PID) control**

**RTO**
**MPC or PID**
**PID**

u (MV) = valves
Move optimization into the control layer

- Try to eliminate RTO-layer
- = «Feedback-optimizing control»
- Unconstrained case: Select CV1 using «self-optimizing» control
  - Ideal: Control cost gradient to zero, CV1=J_u=0
- Changing active constraints: More complicated (but I think we now have solved it 😊)
  - I. Primal-dual optimizing control (with control of constraints on slow timescale)
    - Can use PID control
    - May add fast override control for constraints
  - II. Region-based control (fast control of constraints)
  - III. More inputs (MVs) than constraints: Can use PID with selectors
  - III. General case: MPC with changing cost function (switch CV1)
Optimal steady-state operation

\[
\min_u J(u,d) \\
\text{s.t. } g(u,d) \geq 0 \text{ (constraints)}
\]

- \( J \) = economic cost \([$/s]\)
- Unconstrained case: Optimal to keep gradient \( J_u \) \( \frac{\partial J}{\partial u} = 0 \)

- Constrained case: KKT-conditions: \( g_A = 0, \quad L_u = J_u + \chi^T g_u = 0 \)
Optimal steady-state operation

Want tight control of active constraints for economic reasons

- Active constraint: $g_A = 0$
- Tight control of $g_A$ minimizes «back-off»

• How can we identify and control active constraints?
• How can we switch constraints?
I. Primal-dual control based on KKT conditions: Feedback solution that automatically tracks active constraints by adjusting Lagrange multipliers (= shadow prices = dual variables) $\lambda$

$$L_u = J_u + \lambda^T g_u = 0$$

Inequality constraints: $\lambda \geq 0$

Primal-dual feedback control.
- Makes use of «dual decomposition» of KKT conditions
- Selector on dual variables $\lambda$
- Problem: Constraint control using dual variables is on slow time scale
II. Region-based feedback solution with «direct» constraint control (for case with more inputs than constraints)

\[ \mathbf{L}_u = \mathbf{J}_u + \lambda^T g_u = 0 \]

Introduce \( N: N^T g_u = 0 \)

**Control**

1. Reduced gradient \( N^T J_u = 0 \)
   - «self-optimizing variables»)
2. Active constraints \( g_A = 0 \).

- Bernardino and Skogestad, Decentralized control using selectors for optimal steady-state operation with changing active constraints, J. Process Control, Vol. 137, 2024
Static gradient estimation: 
Very simple and works well!

From «exact local method» of self-optimizing control:

\[ H^J = J_{uu} \left[ G^y_T \left( \tilde{F} \tilde{F}^T \right)^{-1} G^y \right]^{-1} G^y_T \left( \tilde{F} \tilde{F}^T \right)^{-1} \]

where \( \tilde{F} = [FW_d \quad W_{ny}] \) and \( F = \frac{dy_{opt}}{dd} = G^y_d - G^y J_{uu}^{-1} J_{ud} \).
### III. Region-based MPC with switching of cost function (for general case)

Standard MPC with fixed CVs: Not optimal


Proposed: With changing cost (switched CVs)

Figure 1: Typical hierarchical control structure with standard setpoint-tracking MPC in the supervisory layer. The cost function for the RTO layer is $J^0$ and the cost function for the MPC layer is $J_{MPC}$. With no RTO layer (and thus constant setpoints $CV^{sp}$), this structure is not economically optimal when there are changes in the active constraints. For smaller applications, the state estimator may be used also as the RTO estimator.

$$J^0 = \sum_{k=1}^{N} \| CV_k - CV^{sp} \|_Q^2 + \| \Delta u_k \|_R^2$$

Figure 2: Proposed region-based MPC structure with active set detection and change in controlled variables. The possible updates from an upper RTO layer ($y^*, \lambda^*_t$ etc.) are not considered in the present work. Even with no RTO layer (and thus with constant setpoints $CV^{sp}_A$, see (13) and (13), in each active constraint region), this structure is potentially economically optimal when there are changes in the active constraints.

$$J_{MPC}^A = \sum_{k=1}^{N} \| CV_A - CV^{sp}_A \|_{Q_A}^2 + \| \Delta u_k \|_{R_A}^2$$

$$H_0 = [I_{uu} \quad J_w'] [G^T \quad G_2]'$$

\[14\]
Process control layers

- **Real-time optimization layer (RTO):**
  - Optimize setpoints CV1\_s based on detailed nonlinear model (usually steady state)

- **Supervisory/"Advanced" control:**
  - Follow setpoints for CV1
  - Switch between active constraints (change CV1)
  - Look after regulatory layer (avoid that MVs saturate, etc.)

  **Implementation:**
  Alternative 1: “Advanced PID” (ARC) based on “simple elements”
  Alternative 2: MPC (model predictive control)

- **Regulatory control (PID):**
  - Stable operation (CV2)

CV = controlled variable. MV = manipulated variable. ARC = Advanced regulatory (PID) control
«Advanced» control

• This is a relative term
• Usually used for anything than comes in addition to (or in top of) basic PID loops
• Main options
  – ARC using advanced control elements
    • PID + Cascade, feedforward, selectors, etc.
    • This option is preferred if it gives acceptable performance and it’s not too complicated
  – Model predictive control (MPC)
    • Requires more effort to implement and maintain

ARC = Advanced regulatory (PID) control
Academia: MPC

- MPC
  - General approach, but we need a dynamic model
  - Also: MPC is usually implemented only after some time of operation
  - Furthermore: Not all problems are easily formulated using MPC

- So we need something in addition to MPC
Research question: Alternative simpler solutions to MPC

• Would like: Feedback solutions that can be implemented without a detailed models

• Machine learning?
  – Requires a lot of data
  – Can only be implemented after the process has been in operation

• Solution: “Classical advanced control“ (ARC) based on single-loop PID
  – Extensively used by industry
  – Problem for engineers: Lack of design methods
    • Has been around since 1930’s
    • But almost completely neglected by academic researchers
  – Main fundamental limitation: Based on single-loop (need to choose pairing)
QUIZ
What are the three most important inventions of process control?

• Hint 1: According to Sigurd Skogestad
• Hint 2: All three are from the 1930’s

SOLUTION
1. PID controller, in particular, I-action
2. Cascade control
3. Ratio control
Standard Advanced control elements

- Each element links a subset of inputs with a subset of outputs
- Results in simple local tuning

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

E1*. Cascade control
E2*. Ratio control
E3*. Valve (input) 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, 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

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.
“Classical Advanced control” (ARC) using simple control elements

E1. Cascade control
   • Have Extra output (state) measurements

E2. Ratio and feedforward control
   • Have measured disturbance

E12. Decoupling elements
   • Have interactive process

E13. Linearization elements / Adaptive gain
   • Have Nonlinear process

E5-E7. Split-range control (or multiple controllers or VPC)
   • Need extra inputs (MV) to handle all conditions (steady state) (MV-MV switch)

E3. Valve position control (VPC) (Input resetting/Midranging control)
   • Have extra inputs dynamically

E4. Selectors
   • Have changes in active constraints (CV-CV switch)

Often static nonlinear «function block»
One unifying approach is «Transformed inputs» (similar to feedback linearization)

ARC = Advanced regulatory (PID) control
How design classical APC elements?

• Industrial literature (e.g., Shinskey).
  Many nice ideas. But not systematic. Difficult to understand reasoning

• Academia: Very little work
  – I feel alone
Constraint switching (because it is optimal at steady state)

- **CV-CV switching**
  - Control one CV at a time

- **MV-MV switching**
  - Use one MV at a time

- **MV-CV switching**
  - MV saturates so must give up CV
    1. Simple («do nothing»)
    2. Complex (repairing of loops)
MV-MV switching

For cases with one CV (y) and many inputs (MVs)
- Need several MVs to cover whole steady state range
- Example 1: Not both heating ($u_1$) and cooling ($u_2$) to control temperature ($y=T$)
- Example 3: Need both gas ($u_1$) and brake ($u_2$) to control car speed ($y$)

Three alternatives
E5. Split range control
E6. Multiple controllers with different setpoints
E7. Valve position control
Example split range control (E5): Room temperature with 4 MVs

MV-MV switching

MVs (two for summer and two for winter):
1. AC (expensive cooling)
2. CW (cooling water, cheap)
3. HW (hot water, quite cheap)
4. Electric heat, EH (expensive)

SR-block:

$C_{PI} -$ same controller for all inputs (one integral time)
But get different gains by adjusting slopes $\alpha$ in SR-block
Alternative: Multiple Controllers with different setpoints (E6)

Disadvantage (comfort):
- Different setpoints

Advantage (economics):
- Different setpoints (energy savings)
Simulation Room temperature

- Dashed lines: SRC (E5)
- Solid lines: Multiple controllers (E6)

 SRC = split range control

<table>
<thead>
<tr>
<th>Input ($u_k$)</th>
<th>Description</th>
<th>Nominal</th>
<th>Min</th>
<th>Max</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>$u_1 = Q_{AC}$</td>
<td>air conditioning</td>
<td>0</td>
<td>0</td>
<td>4.5</td>
<td>kW</td>
</tr>
<tr>
<td>$u_2 = Q_{HW}$</td>
<td>heating water</td>
<td>0</td>
<td>0</td>
<td>3.0</td>
<td>kW</td>
</tr>
<tr>
<td>$u_3 = Q_{EH}$</td>
<td>electrical heating</td>
<td>0</td>
<td>0</td>
<td>4.0</td>
<td>kW</td>
</tr>
</tbody>
</table>
CV-CV switching

• Only one input (MV) controls many outputs (CVs)
  – Typically caused by change in active constraint
  – Example 1: Control car speed ($y_1$) - but give up if too small distance ($y_2$) to car in front.
  – Example 2: Control power ($y_1$) - but give up if too high engine temperature ($y_2$).

• Use max- or min-selectors (E4)
E4. Selector: One input (u), several outputs (y₁, y₂)

- Note: The selector is on the input u, even though the setpoint/constraint is on the output y
- Sometimes called “override”
  - OK name for temporary dynamic fix, but otherwise a bit misleading
- Selectors are used for output-output (CV-CV) switching
- Selectors work well, but require pairing each constraint with a given input (not always possible)
**Furnace control with safety constraint**

**Input (MV)**
- \( u \) = Fuel gas flowrate

**Output (CV)**
- \( y_1 \) = process temperature \( T_1 \)
  - (desired setpoint or max constraint)
- \( y_2 \) = furnace temperature \( T_2 \)
  - \((T2_{max}= 700\text{C})\)

*Rule: Use min-selector for constraints that are satisfied with a small input*

\[
\begin{align*}
\text{Input (MV)} & : u = \text{Fuel gas flowrate} \\
\text{Output (CV)} & : y_1 = \text{process temperature } T_1 \\
& \quad \text{(desired setpoint or max constraint)} \\
& \quad y_2 = \text{furnace temperature } T_2 \\
& \quad \text{(} T_{2\text{max}}= 700\text{C})
\end{align*}
\]
Design of selector structure

Rule 1 (max or min selector)

• Use max-selector for constraints that are satisfied with a large input
• Use min-selector for constraints that are satisfied with a small input

Rule 2 (order of max and min selectors):

• If need both max and min selector: Potential infeasibility
• Order does not matter if problem is feasible
• If infeasible: Put highest priority constraint at the end

“Systematic design of active constraint switching using selectors.”
Example. Maximize flow with pressure constraints

Input $u = z_1$
Want to maximize flow, $J = -F$:

Optimization problem is:

$$\max_{z_1} F$$

subject to:

$$F \leq F_{\text{max}}$$
$$p_1 \leq p_{1,\text{max}}$$
$$p_1 \geq p_{1,\text{min}}$$
$$z_1 \leq z_{1,\text{max}}$$

(15)

where $F_{\text{max}} = 10$ kg/s, $z_{1,\text{max}} = 1$, $p_{1,\text{max}} = 2.5$ bar, and $p_{1,\text{min}} = 1.5$ bar. Note that there are both max and min- constraints on $p_1$. De-
Disturbances in $p_0$ and $p_2$ (unmeasured)
Example «simple» MV-CV switching (no selector)

Anti-surge control (= min-constraint on F)

Minimize recycle (MV=z) subject to
- CV = F \geq F_{\text{min}}
- MV \geq 0

- No selector required, because MV=z has a «built-in» max-selector at z=0.
- Generally: «Simple» MV-CV switching (with no selector) can be used if we satisfy the input saturation rule: «Pair a MV that may saturate with a CV that can be given up (when the MV saturates at z=0)”
Suggest a solution which achieves

- \( p < p_{\text{max}} = 37 \text{ bar} \) (max delivery pressure)
- \( P_0 > p_{\text{min}} = 30 \text{ bar} \) (min. suction pressure)
- \( F < F_{\text{max}} = 19 \text{ t/h} \) (max. production rate)
- \( F_0 > F_{\text{min}} = 10 \text{ t/h} \) (min. through compressor to avoid surge)

Rule CV-CV switching: Use max-selector for constraints that are satisfied by a large input (MV) (here: valve opening \( z \))
Complex MV-CV switching

• = CV-CV switch followed by MV-MV switch
• Example inventory control: Avoid «long loop» (dynamic issue)
Example: Inventory control

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

“Long loop”

(d) Inventory control with undesired “long loop”, not in accordance with the “radiation rule” (for given product flow, TPM= $F_3$)
Example. Very smart selector strategy: **Bidirectional inventory control**
Reconfigures automatically with optimal buffer management!!

\[
\text{Max flow: } F = \infty
\]

F.G. Shinskey, «Controlling multivariable processes», ISA, 1981
C. Zotica, S. Skogestad and K. Forsman, Comp. Chem. Eng, 2021

![Diagram of multivariable processes](image-url)
\[ F^* = \infty \quad H \quad L \quad F^* = \infty \quad H \quad L \quad F^* = \infty \quad H \quad L \quad 1 \]

\[ F_0 = 0.5 \quad \text{Unit 1} \quad F_1 = 0.5 \quad \text{Unit 2} \quad F_2 = 1 \quad \text{Unit 3} \quad F_3 = 1 \]
Figure 12: Simulation of a 19 min temporary bottleneck in flow $F_1$ for the control structures in Fig. 3d with the TPM downstream of the bottleneck.
Challenge: Can MPC be made to do his? Optimally reconfigure loops and find optimal buffer?

- Yes, possible with standard setpoint-based MPC if we use
  - Trick: All flow setpoints = infinity (unachoevable setpoint)
- What about Economic MPC? Cannot do it easily; may try scenario-MPC
Example adaptive cruise control: CV-CV switch followed by MV-MV switch

Note: This is not Complex MV-CV switching, because then the order would be opposite.
Important insight

• Many problems: Optimal steady-state solution always at constraints
• In this case optimization layer may not be needed
  – if we can identify the active constraints and control them using selectors
E8. Anti-windup

- All the controllers shown need anti-windup to «stop integration» during periods when the control action \(v_i\) is not affecting the process:
  - Controller is disconnected (because of selector)
  - Physical MV \(u_i\) is saturated

Anti-windup using back-calculation. Typical choice for tracking constant, \(K_T = 1\)
Challenges selector design

• Standard approach requires pairing of each active constraint with a single input
  – May not be possible in complex cases
• Stability analysis of switched systems is still an open problem
  – Undesired switching may be avoided in many ways:
    • Filtering of measurement
    • Tuning of anti-windup scheme
    • Minimum time between switching
    • Minimum input change
When use MPC?

When conventional APC performs poorly or becomes complex

- Cases with many changing constraints (where we cannot assign one input to each constraint)
- Interactive process
- Know future disturbances and setpoint changes (predictive capability)
Conclusion Advanced process control (APC)

• Classical APC, aka «Advanced regulatory control» (ARC) or «Advanced PID»:
  – Works very well in many cases
  – Optimization by feedback (active constraint switching)
  – Need to pair input and output.
    • Advantage: The engineer can specify directly the solution
    • Problem: Unique pairing may not be possible for complex cases
  – Need model only for parts of the process (for tuning)
  – Challenge: Need better teaching and design methods

• MPC may be better (and simpler) for more complex multivariable cases
  – But MPC may not work on all problems (Bidirectional inventorycontrol)
  – Main challenge: Need dynamic model for whole process
  – Other challenge: Tuning may be difficult
Academic process control community fish pond

Simple solutions that work ($\text{ARC} = \text{PID}++$)

Optimal centralized Solution (\text{EMPC})

Please join me, I feel a little alone