Part 2.
Plantwide process control
«Control architectures»

Sigurd Skogestad

Plantwide control (Control architecture)

- Objective: Put controllers on flow sheet (make P&ID)
- Two main objectives for control: Longer-term economics (CV1) and shorter-term stability (CV2)
- Regulatory (basic) control layer for CV2 and supervisory (advanced) control layer for CV1

How can we design a control system for a complete chemical plant?

Where do we start?

What should we control? And why?

Sigurd at Caltech (1984)

How we design a control system for a complete chemical plant?

- Where do we start?
- What should we control? and why?
- etc.
- etc.

Control system structure*

Alan Foss ("Critique of chemical process control theory", AIChE Journal,1973):

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?

*Current terminology: Control system architecture

Plantwide control =
Control structure (architecture) design

- *Not* the tuning and behavior of each control loop…

- But rather the *control philosophy* of the overall plant with emphasis on the ***structural decisions***:
  - Selection of controlled variables ("outputs")
  - Selection of manipulated variables ("inputs")
  - Selection of (extra) measurements
  - Selection of control **configuration** (structure of overall controller that interconnects the controlled, manipulated and measured variables)
  - Selection of controller type (LQG, H-infinity, PID, decoupler, MPC etc.)

Main objectives of a control system

1. Economics: Implementation of acceptable (near-optimal) operation
2. Regulation: Stable operation

ARE THESE OBJECTIVES CONFLICTING?

- Usually NOT
  - Different time scales
    - Stabilization → fast time scale
  - Stabilization doesn't "use up" any degrees of freedom
    - Reference value (setpoint) available for layer above
    - But it "uses up" part of the time window (frequency range)

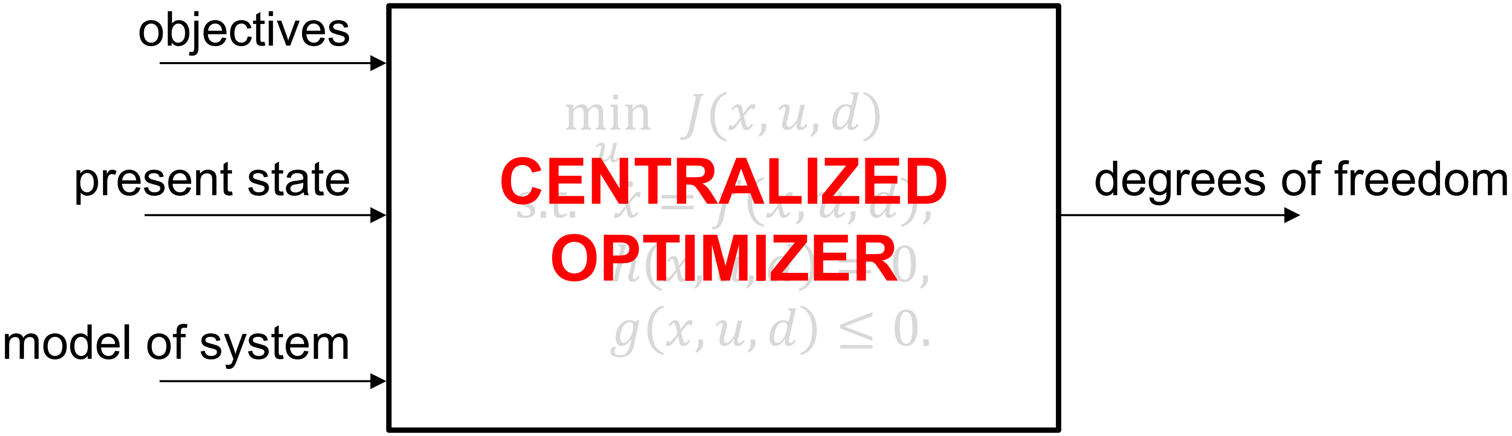
Optimal operation

General approach: minimize cost / maximize profit, subject to satisfying constraints (product quality, environment, resources)

Mathematically,

$$
\begin{aligned}
\min_{u} \quad & J(x,u,d) \\
\text{s.t.} \quad & \dot{x} = f(x,u,d), \\
& h(x,u,d) = 0, \\
& g(x,u,d) \leq 0.
\end{aligned}
$$

Optimal operation (in theory)



(example: Economic MPC)

Procedure:
- Obtain model of overall system
- Estimate present state
- Optimize all degrees of freedom

Problems:
- Model not available
- Optimization is complex
- Not robust (difficult to handle uncertainty)
- Slow response time

Engineering systems

- Most (all?) large-scale engineering systems are controlled using hierarchies of quite simple controllers
  - Large-scale chemical plant (refinery)
  - Commercial aircraft

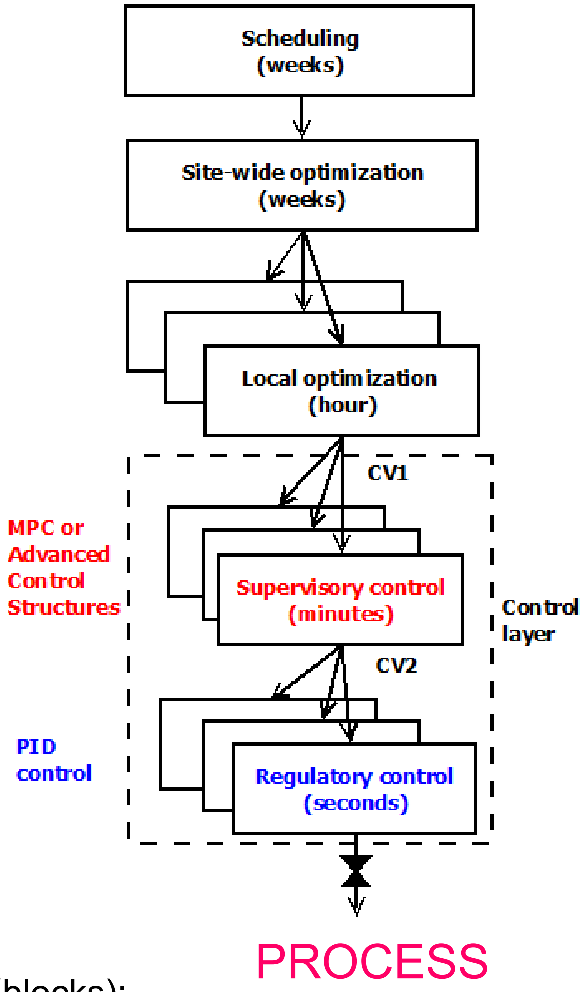
- 100’s of loops

- Simple components:

  on-off + PI-control + nonlinear fixes + some feedforward

Two fundamental ways of decomposing the controller

- Vertical (hierarchical; cascade)
- Based on time scale separation
- Decision: Selection of CVs that connect layers



- Horizontal (decentralized)
- Usually based on distance
- Decision: Pairing of MVs and CVs within layers

In addition: Decomposition of controller into smaller elements (blocks): Feedforward element, nonlinear element, estimators (soft sensors), switching elements

Time scale separation: Control* layers

Two objectives for control: Stabilization and economics

- **Supervisory (“advanced”) control layer**

  Tasks:

  – Follow set points for CV1 from economic optimization layer

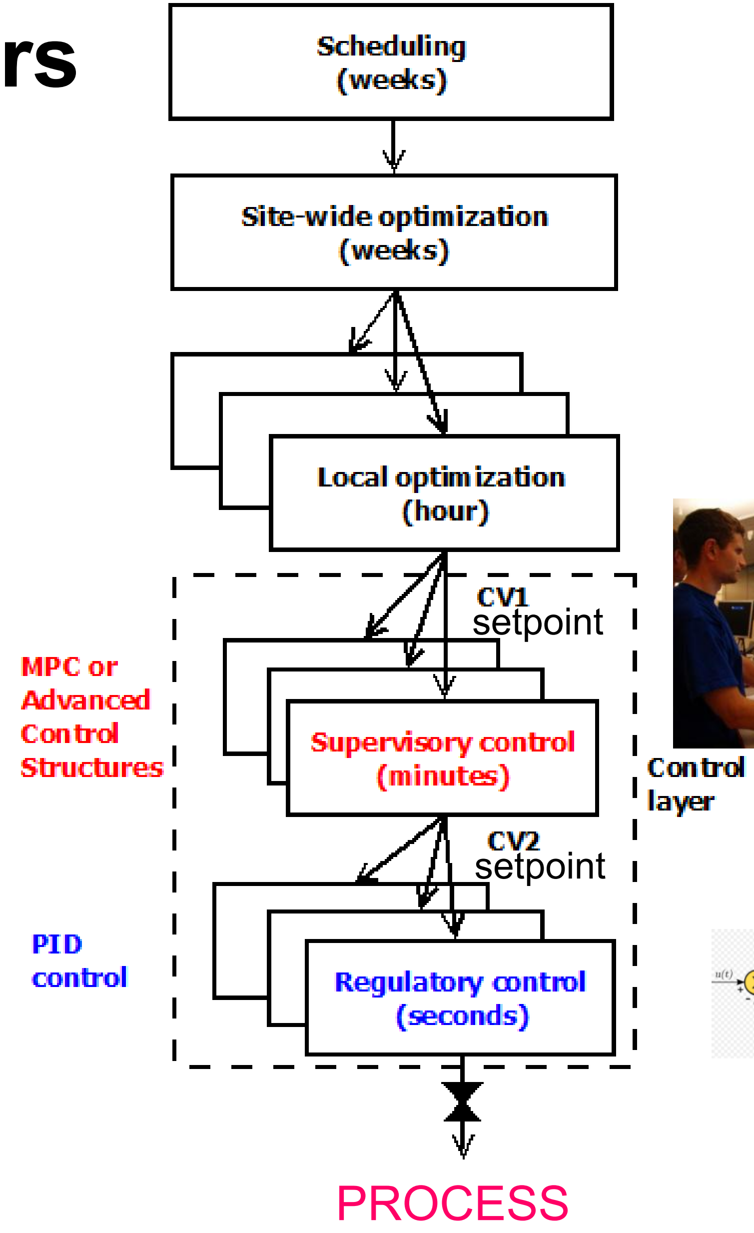
  – Switch between active constraints (change CV1)

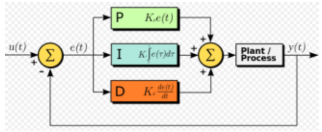
  – Look after regulatory layer (avoid that MVs saturate, etc.)

- **Regulatory control (PID layer):**

  – Stable operation (CV2)

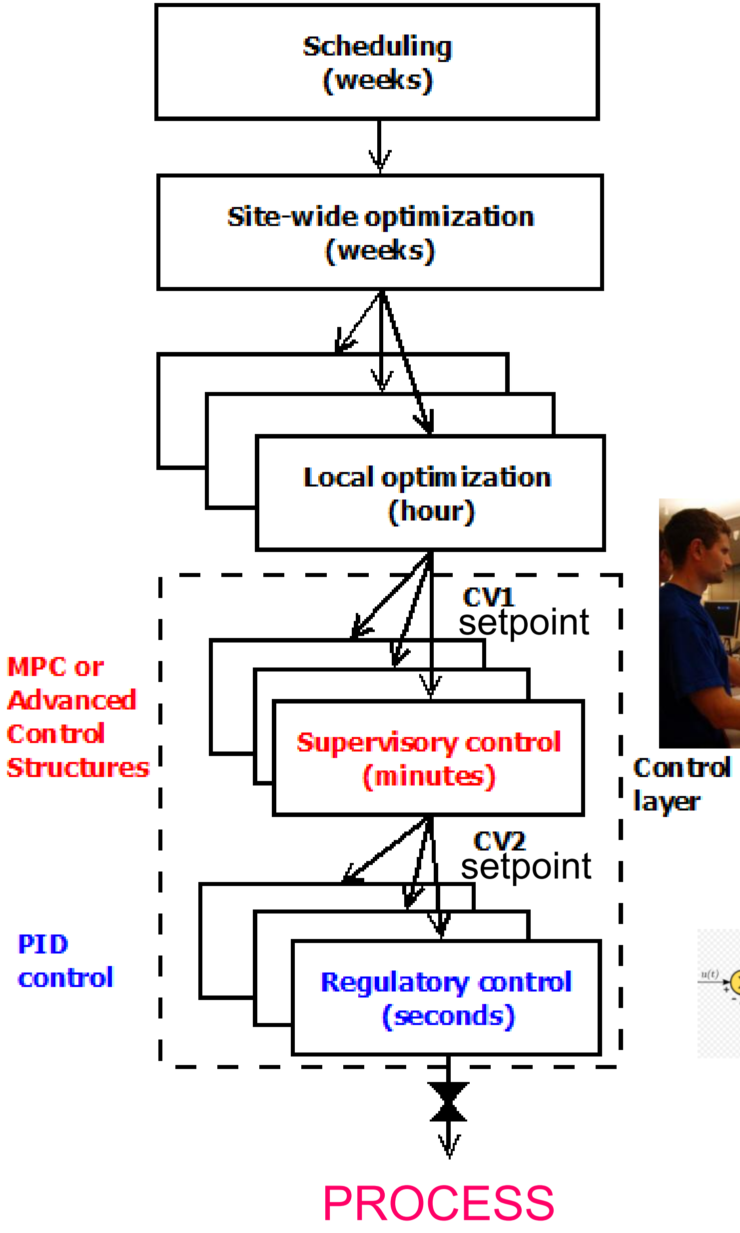
*My definition of «control» is that the objective is to track setpoints

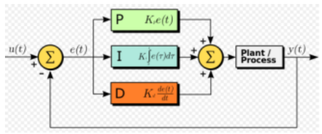




«Advanced» control

- Advanced: This is a relative term
- Usually used for anything than comes in addition to (or in top of) basic PID loops
- Mainly used in the «supervisory» control layer
- Two main options
  - **Standard «Advanced regulatory control» (ARC) elements**
    - Based on decomposing the control system
      - Cascade, feedforward, selectors, etc.
    - This option is preferred if it gives acceptable performance
  - **Model predictive control (MPC)**
    - Requires a lot more effort to implement and maintain
    - Use for interactive processes
    - Use with known information about future (use predictive capanulities)





Combine control and optimization into one layer?

EMPC: Economic model predictive “control”

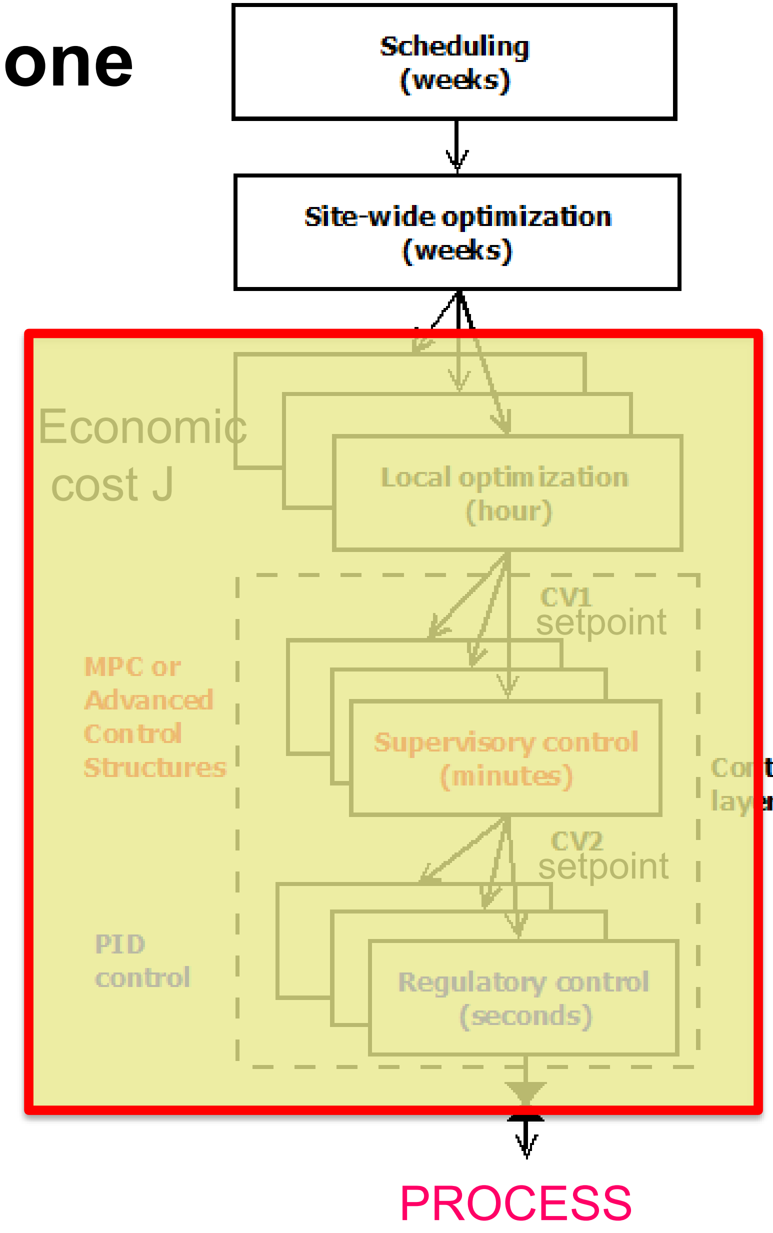
$J_{EMPC} = J + J_{control}$

- J [$/s] = $J_{economic}$ = cost feed + cost energy – value products
- $J_{control} = \Sigma \Delta u_i^2$ (typical) - Penalize input usage

NO, combining layers is generally not a good idea!
(the good idea is to separate them!)

One layer (EMPC) is optimal theoreretically, but

- Need detailed dynamic model of everything
- Tuning difficult and indirect
- Slow! (or at least difficult to speed up parts of the control)
- Robustness poor
- Implementation and maintainance costly and time consuming



EMPC

(no setpoints CV1, CV2)

What about «conventional» RTO and MPC?

- Yes, it’s OK
- Both has been around for more than 50 years (since 1970s)
  - but the expected growth never came
- MPC is still used mostly in large-scale plants (petrochemical and refineries).
- MPC is far from replacing PID as some expected in the 1990s.

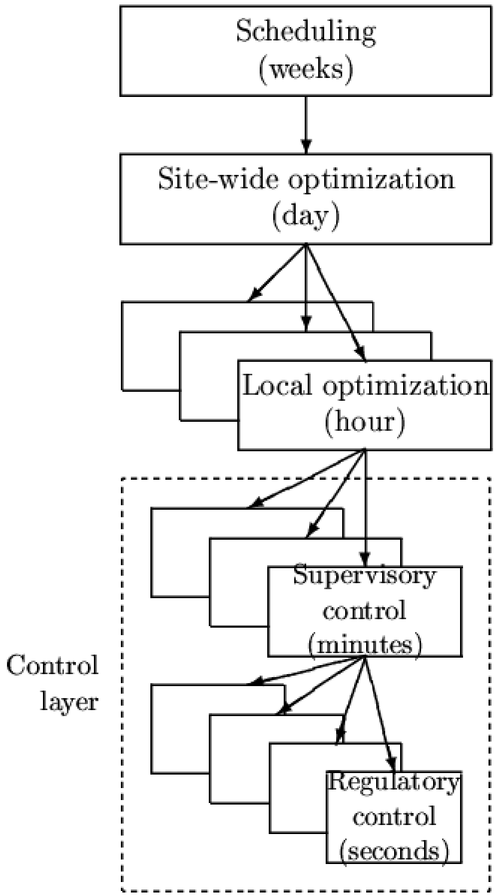
Alternative solutions for advanced control

- **Machine learning?**
  - Requires a lot of data, not realistic for process control
  - And: Can only be implemented after the process has been in operation

- **"Classical advanced regulatory control" (ARC) based on single-loop PIDs?**
  - YES!
  - 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)

ARC = Advanced regulatory control

Optimal operation and control objectives: What should we control?



CV_1 (economics)

CV_2 (stabilization)

Skogestad procedure for control structure design:

I. Top Down (analysis)
  - Step S1: Define operational objective (cost) and constraints
  - Step S2: Identify degrees of freedom and optimize operation for disturbances
  - Step S3: Implementation of optimal operation
    - What to control? (CV1) (self-optimizing control)
  - Step S4: Where set the production rate (TPM)? (Inventory control)

II. Bottom Up (design)
  - Step S5: Regulatory control: What more to control (CV2)?
  - Step S6: Supervisory control
  - Step S7: Real-time optimization

TPM = Throughput manipulator

S. Skogestad, ``Control structure design for complete chemical plants'', *Computers and Chemical Engineering*, **28** (1-2), 219-234 (2004)

ABB: Skip rest of these slides (read yourself)

Step S2b: Optimize for expected disturbances

- What are the optimal values for our degrees of freedom u (MVs)?

J = cost feed + cost energy – value products

- Minimize J with respect to u for given disturbance d (usually steady-state):

$$\min_u J(x,u,d)$$

subject to:

– Model equations : $\dot{x} = f(x,u,d) = 0$

– Operational constraints: $g(x,u,d) \leq 0$

OFTEN VERY TIME CONSUMING

– Commercial simulators (Aspen, Unisim/Hysys) are set up in “design mode” and often work poorly in “operation (rating) mode”.

– Optimization methods in commercial simulators often poor

  - We can use Matlab or even Excel “on top”

<u>Step S2b</u>: Optimize for expected disturbances

- Need good model, usually steady-state
- Optimization is time consuming! But it is offline
- **Main goal: Identify ACTIVE CONSTRAINTS (optimal to maintain)**
- A good engineer can often guess the active constraints:

Example:

Cost J = T [h]
Constraint: v ≤ 50 km/h
Control implementation: Cruise control with setpoint 50 km/h (active constraint)

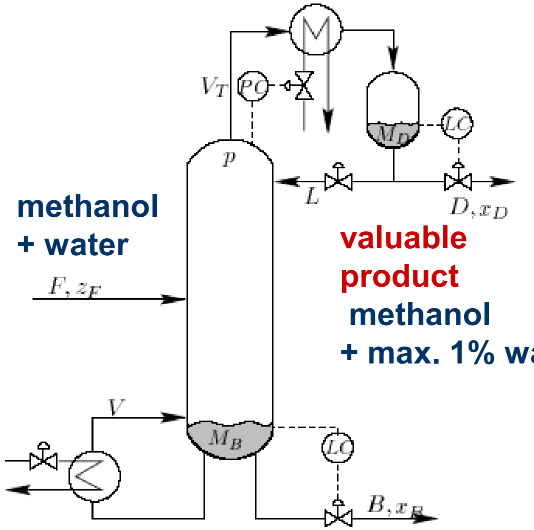
<u>Example Step S2b</u>: Active constraints for distillation

- **Both products (D, B) generally have purity specs**

- **Rule 1: Purity spec. always active for valuable product**
  - Reason: 1. Maximize amount of valuable product (D or B)
    - Avoid product "give-away" (So "sell water as methanol")
  - Reason 2: Save energy (because overpurification costs energy)

- **Rule 2: May overpurify (not control) cheap product**
  - Reason: Increase amount of valuable product ("reduce loss of methanol in bottom product")
  - This typically results in an unconstrained optimum because overpurification costs energy ("optimal purity of cheap product")

methanol
+ water



valuable product
methanol
+ max. 1% water

cheap product (byproduct)
water
+ max. 5% methanol

<u>Step S2b</u>: Optimize for expected disturbances

min J = cost feed + cost energy – value products

Generally: Two main cases (modes) depending on market conditions:

Mode 1 (low product price). Given throughput (feed rate)

Mode 2 (high product price). Maximum production (more constrained)

Comment: Depending on prices, Mode 1 may include many subcases (active constraints regions)

Mode 1. Given feedrate

Amount of products is then usually indirectly given and

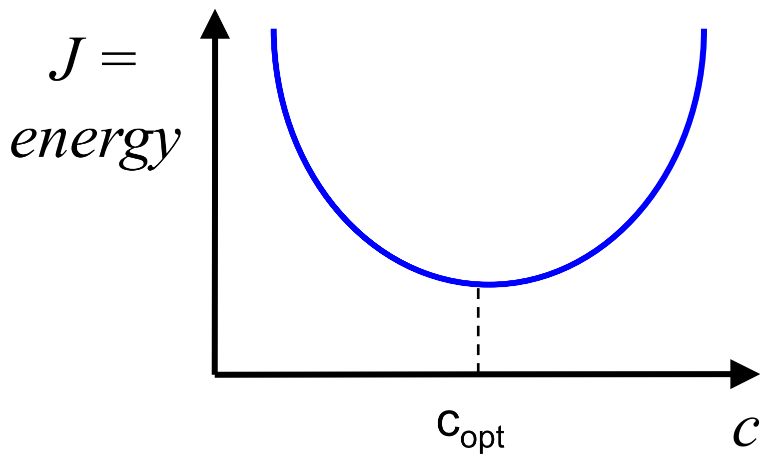
$$J = \underbrace{\text{cost feed} - \text{value products}}_{\text{Often constant}} + \textit{cost energy}$$

Optimal operation is then usually *unconstrained*

“maximize efficiency (energy)”

Control:

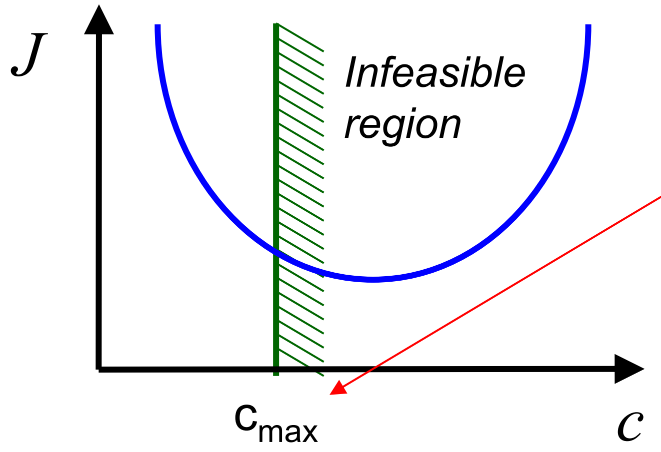
- Operate at optimal trade-off
- NOT obvious what to control
- CV = Self-optimizing variable



Mode 2. Maximum production

J = cost feed + cost energy – value products

- Assume feed rate is degree of freedom
- Assume products much more valuable than feed
- Optimal operation is then to maximize product rate
- **“max. constrained”, prices do not matter**



Control:
- Focus on tight control of bottleneck
- “Obvious what to control”
- CV = ACTIVE CONSTRAINT
- $CV_s = c_{max}$

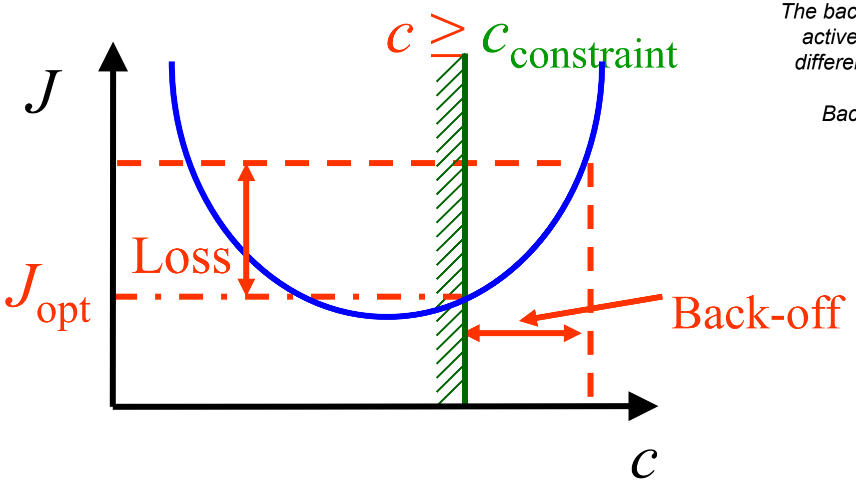
Step S3. Implementation of optimal operation

- Assume we have analyzed the optimal way of operation. How should it be implemented?

- **What to control?** (primary CV1's)
  1. Active constraints
  2. Self-optimizing variables (for unconstrained degrees of freedom)

1. Control of Active output constraints

Need back-off



The backoff is the "safety margin" from the active constraint and is defined as the difference between the constraint value and the chosen setpoint
Backoff = | Constraint – Setpoint |

a) If constraint can be violated dynamically (only average matters)
- Required Back-off = "measurement bias" (steady-state measurement error for *c*)

b) If constraint cannot be violated dynamically ("hard constraint")
- Required Back-off = "measurement bias" + maximum dynamic control error

Want tight control of hard output constraints to reduce the back-off. "Squeeze and shift"-rule

Motivation for better control of active constraints: Squeeze and shift rule

y

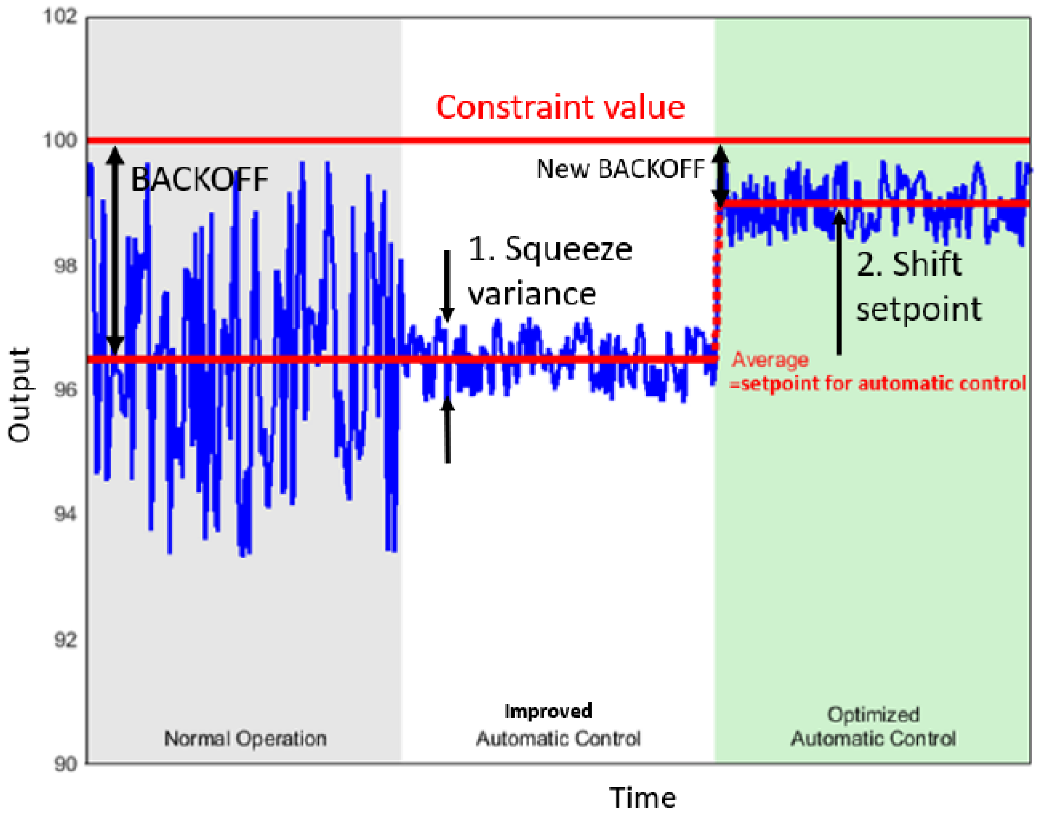
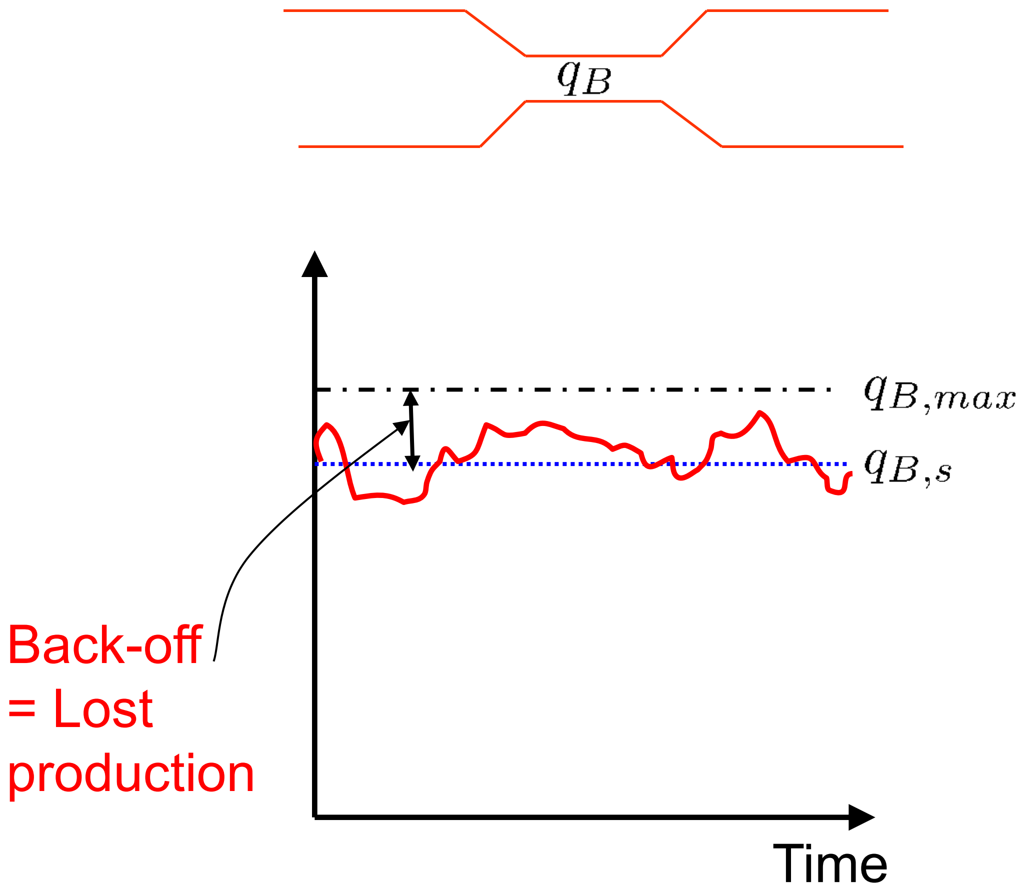


Figure 8: Squeeze and shift rule: Squeeze the variance by improving control and shift the setpoint closer to the constraint (i.e., reduce the backoff) to optimize the economics (Richalet et al., 1978).

Example: max. throughput.

Want tight bottleneck control to reduce backoff!



2. Unconstrained optimum

Control “self-optimizing” variable!

- Which variable is best?
- Often not obvious
  - Example: Control heart rate for marathon runner

What are good self-optimizing variables?

1. Optimal value of CV is constant
2. CV is “sensitive” to MV (large gain)

Note: Tight control of the self-optimizing variable is usually not important because optimum should be flat.

Conclusion optimal operation

ALWAYS:

1. Control active constraints and control them tightly!!
   - Good times: Maximize throughput → tight control of bottleneck

2. Identify “self-optimizing” CVs for remaining unconstrained degrees of freedom

- Use offline analysis to find expected operating regions and prepare control system for this!
  - One control policy when prices are low (nominal, unconstrained optimum)
  - Another when prices are high (constrained optimum = bottleneck)

ONLY if necessary: consider RTO <u>on top of this</u>