Systematic design of advanced control structures

Adriana Reyes-Lúa

February 28th, 2020

Norwegian University of Science and Technology

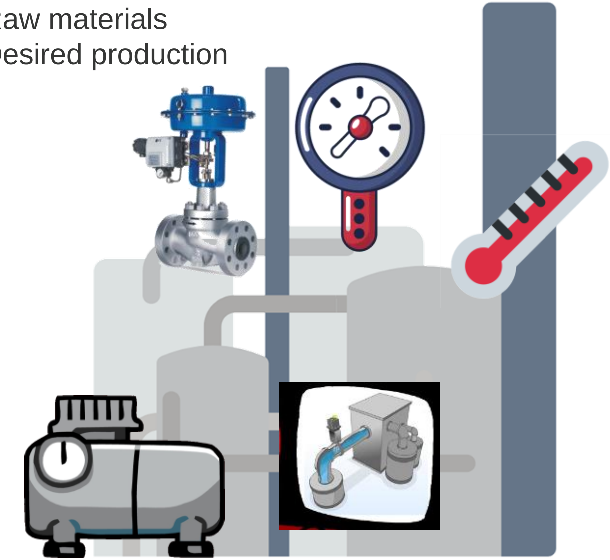
Content

- Motivation and scope
- Active constraint switching with advanced control structures (chapter 2)
  - Case study: mixing
  - Case study: distillation column
  - Case study: cooling cycle (chapter 3)
  - Case study: cooler (chapter 4)
- MV to MV constraint switching
  - Split range control
    - Design of standard split range controllers (chapter 5)
    - Generalized split range controller (chapter 6)
  - Multiple controllers with different setpoints (chapter 7)
- Improved PI control for tank level (chapter 8)
- Conclusions

Motivation and scope

DV: disturbance variable (*d*)

- Ambient temperature
- Raw materials
- Desired production



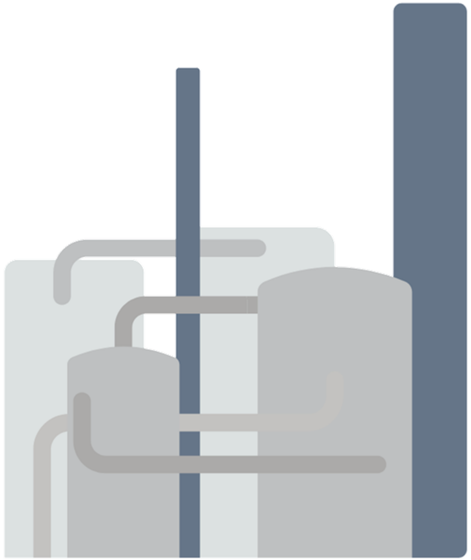
CV: controlled variable (output, *y*)

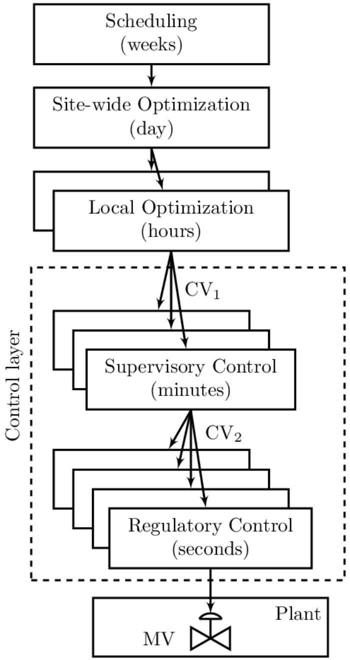
- Temperature
- Pressure
- Concentration

MV: manipulated variable (input, *u*)

- Valve opening
- Compressor rotational speed

Motivation and scope





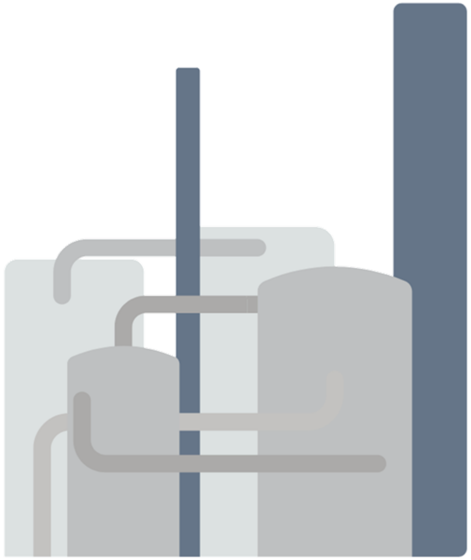
Top-down analysis:

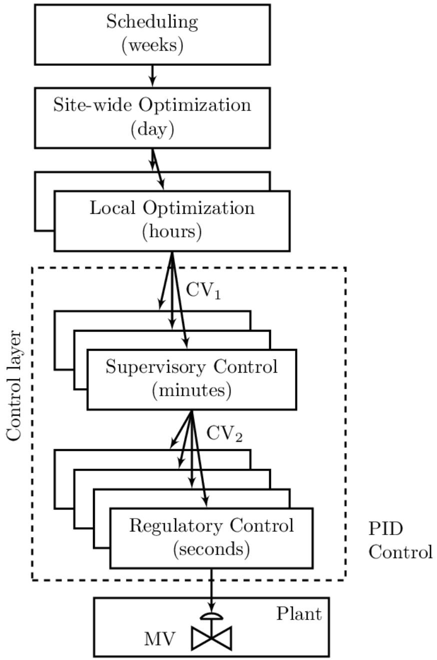
S1-S4: Identify steady-state optimal operation

Bottom-up analysis:

S5-S7: Design control structure

Motivation and scope



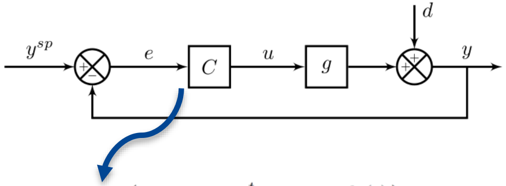


Bottom-up analysis:

S5: regulatory control layer

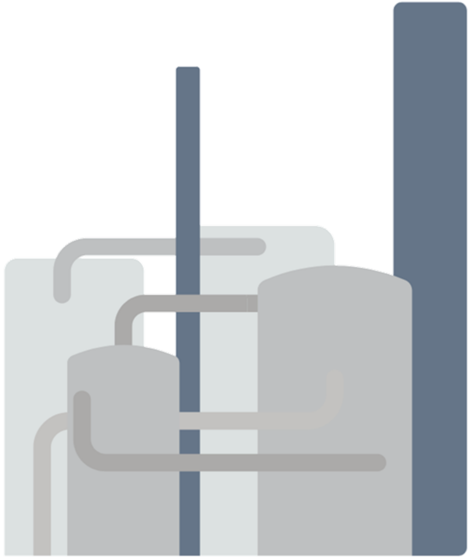
S6: supervisory control layer

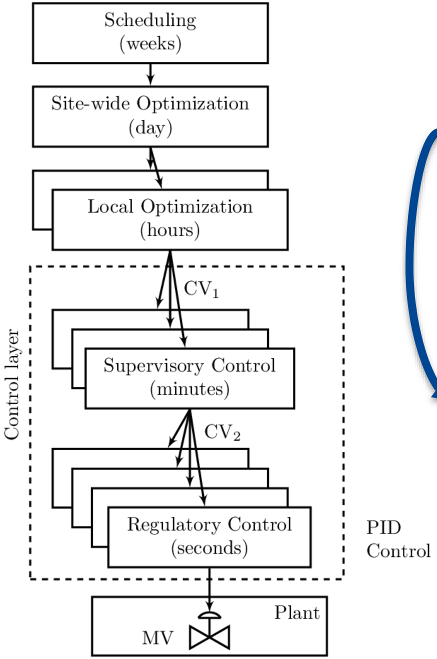
S7: online optimization layer



$$u(t) = u^0 + K_C \left(e(t) + \frac{1}{\tau_I} \int_0^t e(t) + \tau_d \frac{de(t)}{dt} \right)$$

Motivation and scope





Bottom-up analysis:

S5: regulatory control layer

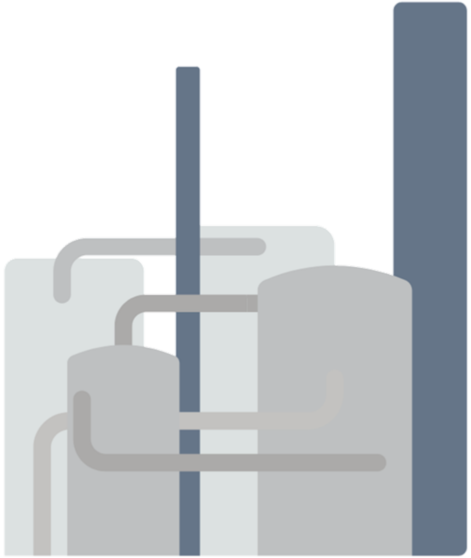
S6: supervisory control layer

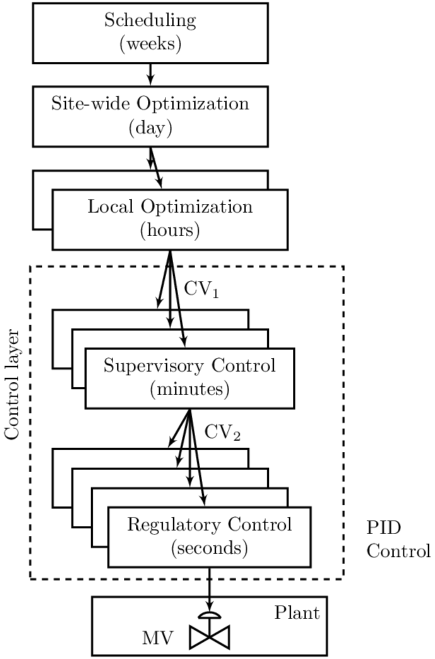
S7: online optimization layer

Keeps operation
in the right
active constraint region

Motivation and scope

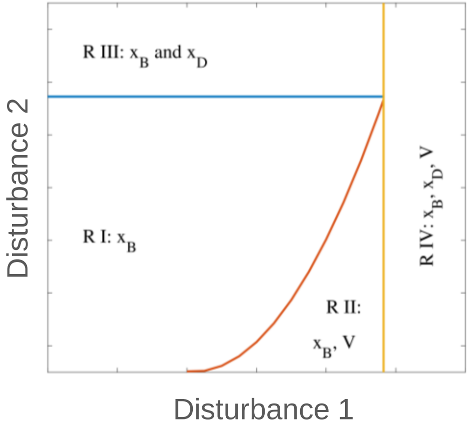
S6: supervisory control layer





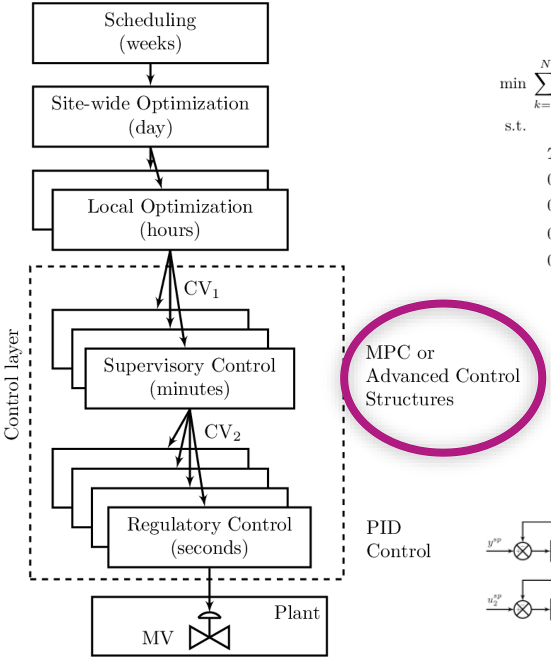
Constraint region

«region in the disturbance space defined by which constraints are active within it»



Jacobsen and Skogestad (2011) Active constraint regions for optimal operation of chemical processes. Industrial & Engineering Chemistry Research.

Motivation and scope



S6: supervisory control layer

Model predictive control

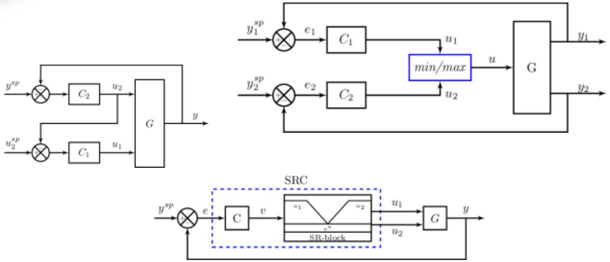
$$
\min \sum_{k=1}^{N} \left(\omega_1 \left\| (T_{H_k} - T_H^{sp}) \right\|^2 + \omega_2 \left\| (F_{H_k}^{max} - F_{H_k}) \right\|^2 \right)
$$

s.t.

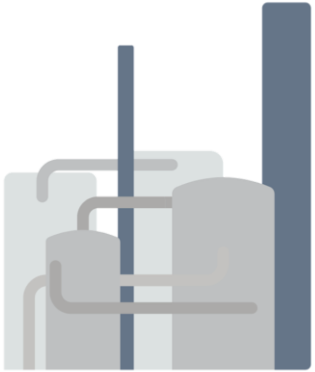
$$
\left.\begin{aligned}
&\boldsymbol{T}_{k,i} = f(T_{H_k,i}, T_{H_k,i-1}, T_{C_k,i}, T_{C_k,i+1}, F_{H_k}, F_{C_k}) \\
&0 \le F_{H_k} \le F_H^{max} \\
&0 \le F_{C_k} \le F_C^{max}
\end{aligned}\right\} \quad \forall k \in \{1, \ldots, N\}
$$

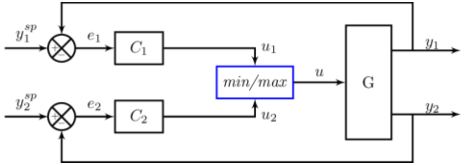
$$
\left.\begin{aligned}
&0 \le \Delta F_{H_k} \le 0.1 F_H^{max} \\
&0 \le \Delta F_{C_k} \le 0.1 F_C^{max}
\end{aligned}\right\} \quad \forall k \in \{1, \ldots, N-1\}
$$

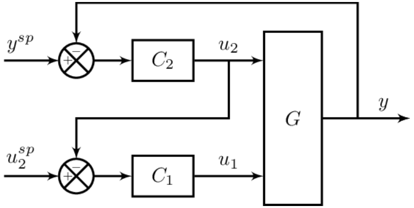
Advanced control structures

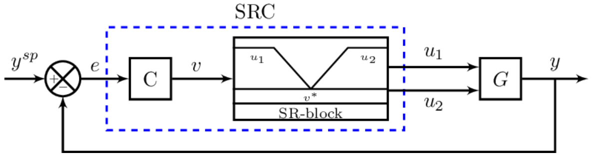


Active constraint switching with classical advanced control structures









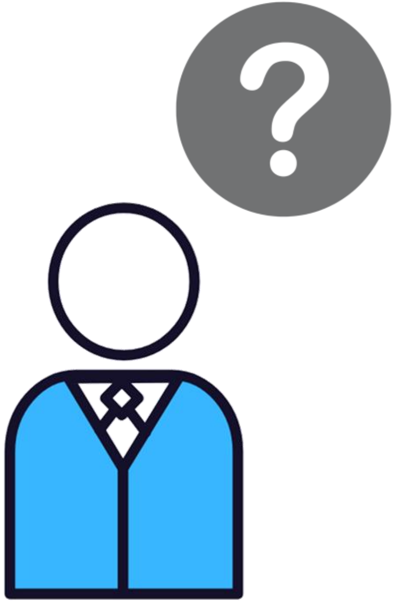
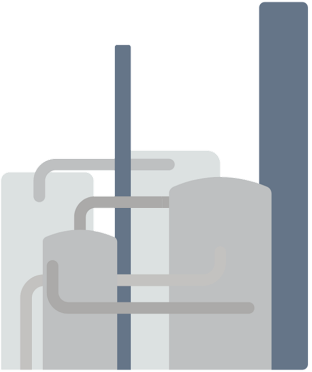
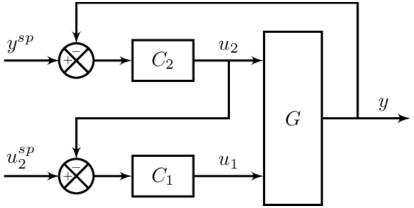
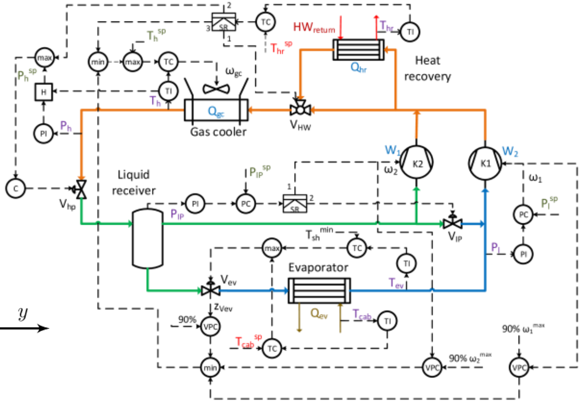


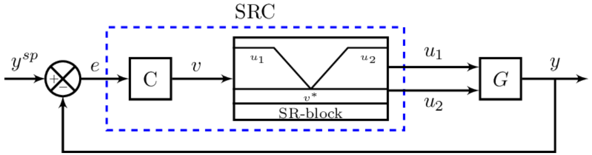
Figure taken from www.transmittershop.com/blog/causes-solutions-annoying-noise-control-valves

Active constraint switching with classical advanced control structures









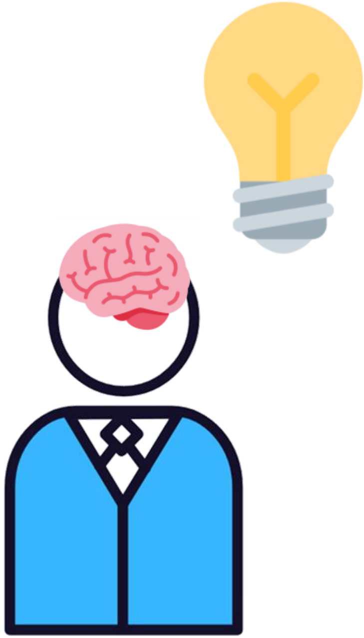
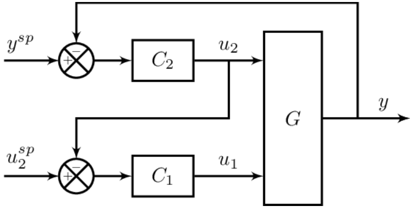
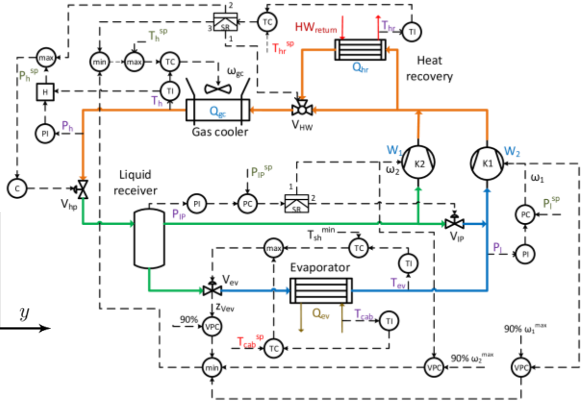
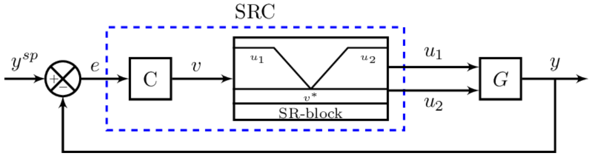


Figure taken from www.transmittershop.com/blog/causes-solutions-annoying-noise-control-valves

Active constraint switching with classical advanced control structures







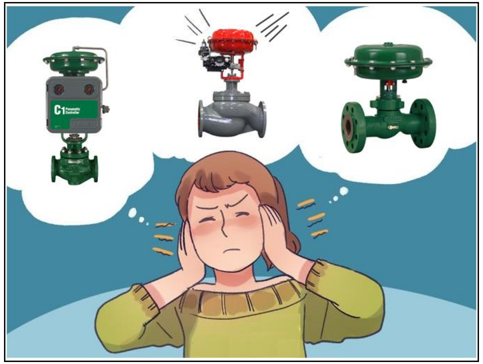


Figure taken from www.transmittershop.com/blog/causes-solutions-annoying-noise-control-valves

NTNU

Design procedure for active constraint switching with classical advanced control structures

A1
- Define control objectives, CV constraints and MV constraints

A2
- Organize constraints in priority list

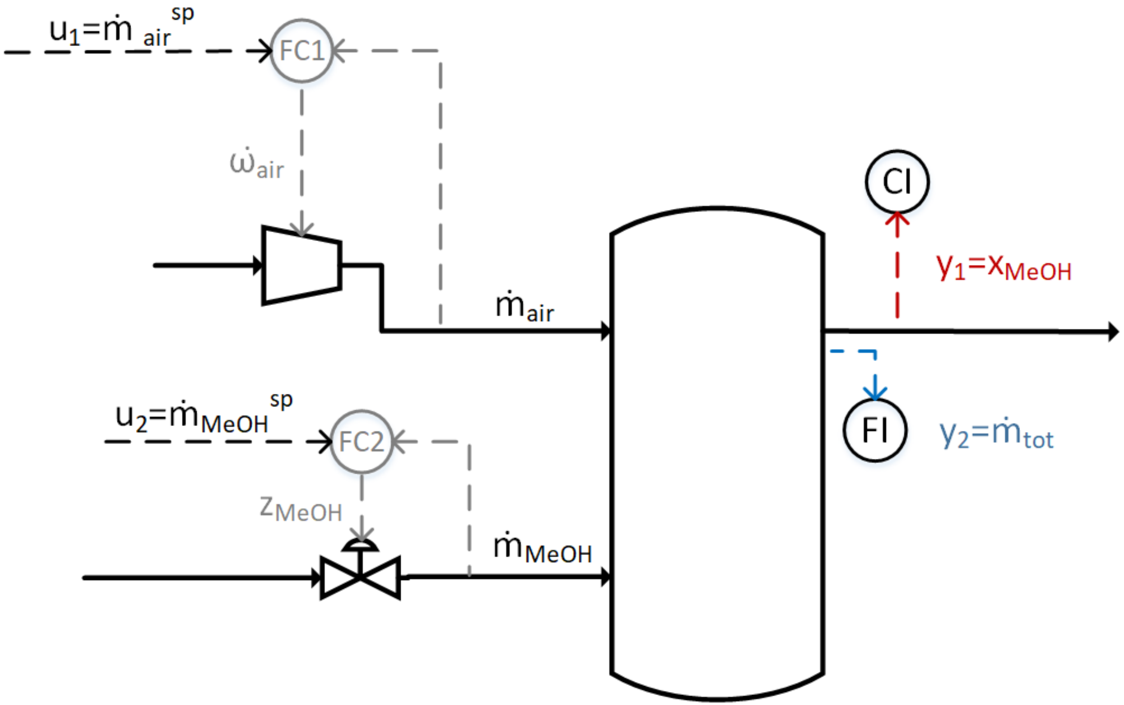
A3
- Identify possible and relevant active constraint switches

A4
- Design control structure for optimal operation

A5
- Design control structure to handle active constraint switches

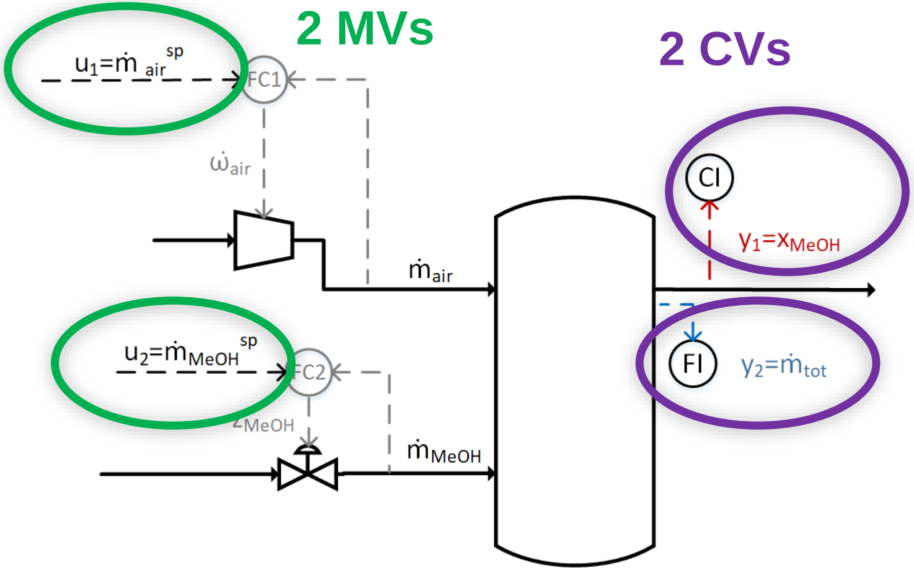
Design procedure for active constraint switching

Case study:
Mixing of
air and MeOH



Design procedure for active constraint switching

Step A1: Define control objectives, CV constraints and MV constraints

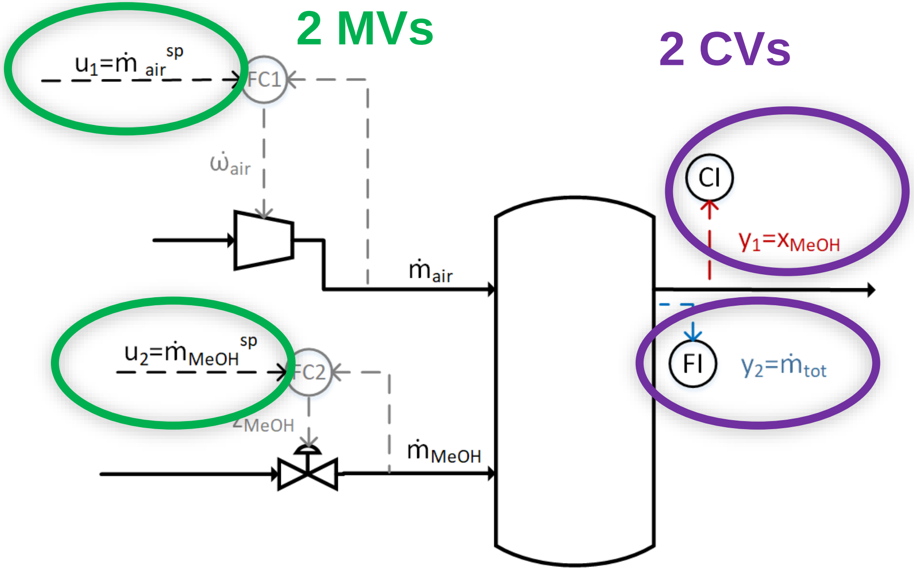


Control objectives:

- Keep $y_1 = x_{MeOH} = 0.10$ ← ideal
- Keep $y_1 = x_{MeOH} > 0.08$
- Control $y_2 = m_{tot}$ ← ideal

Design procedure for active constraint switching

Step A1: Define control objectives, CV constraints and MV constraints



Control objectives:

- Keep $y_1 = x_{MeOH} = 0.10$
- Keep $y_1 = x_{MeOH} > 0.08$
- Control $y_2 = m_{tot}$

Variable	Units	Maximum	Nominal
$y_1 = x_{MeOH}$	kmol/kmol	0.10	0.10
$y_2 = \dot{m}_{tot}$	kg/h	-	26860
$u_1 = \dot{m}_{air}$	kg/h	25800	23920
$u_2 = \dot{m}_{MeOH}$	kg/h	-	2940

u_1 is has a maximum value

Design procedure for active constraint switching

Step A2: Organize constraints in priority list

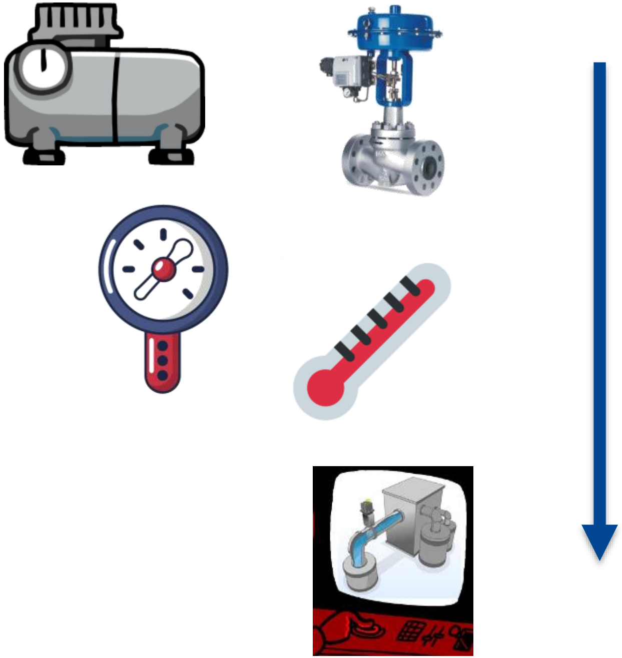


Figure from www.indelac.com/blog/control-valves-vs.-regulators-in-control-applications

Design procedure for active constraint switching

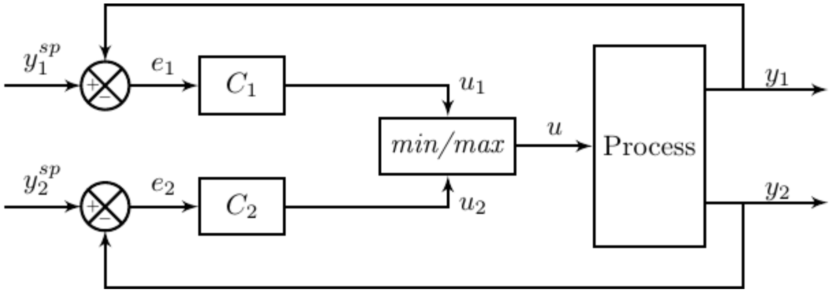
Step A2: Organize constraints in priority list

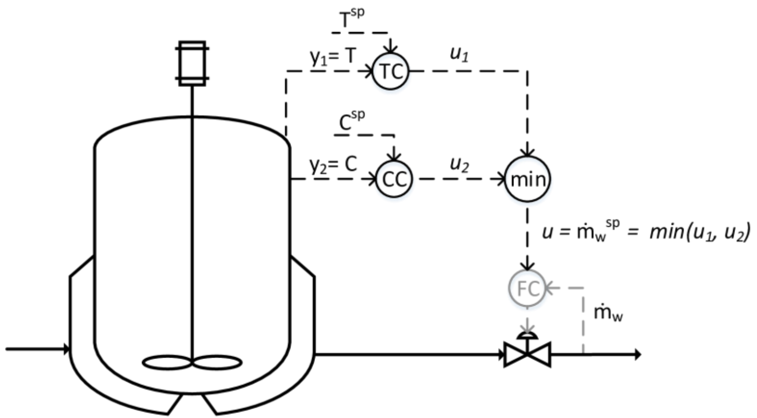
(P1) Physical MV inequality constraints	• Constraint on air flow (u_1) • Constraint on MeOH flow (u_2)	$\dot{m}_{air}^{min} \leq \dot{m}_{air} \leq \dot{m}_{air}^{max}$ $\dot{m}_{MeOH}^{min} \leq \dot{m}_{MeOH} \leq \dot{m}_{MeOH}^{max}$
(P2) Critical CV inequality constraints	• Constraint (max and min) on x_{MeOH} (y_1)	$x_{MeOH}^{min} \leq x_{MeOH} \leq x_{MeOH}^{max}$
(P3) Less critical CV and MV constraints	• Setpoint on x_{MeOH} (y_1)	$x_{MeOH} = x_{MeOH}^{sp}$
(P4) Desired throughput	• Setpoint on m_{tot} (y_2)	$\dot{m}_{tot} = \dot{m}_{tot}^{sp}$
(P5) Self-optimizing variables	• No unconstrained degrees of freedom	

Design procedure for active constraint switching

Step A3: Identify possible and relevant active constraint switches

- **Case 1: CV to CV constraint switching**
  One MV switching between two alternative CVs.



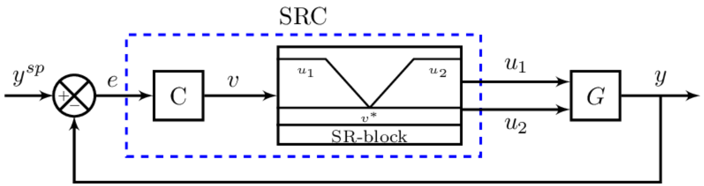


NTNU

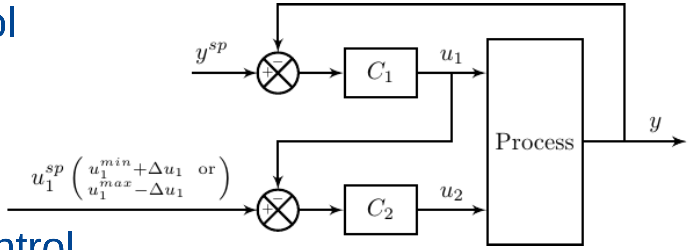
Design procedure for active constraint switching

Step A3: Identify possible and relevant active constraint switches

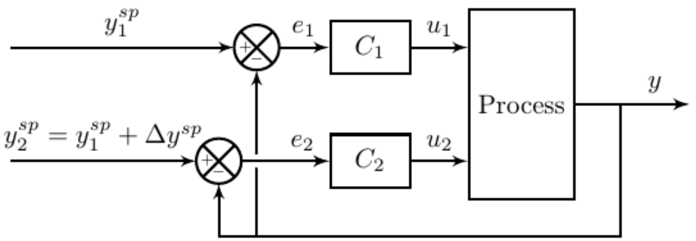
- **Case 2: MV to MV constraint switching**
  More than one MV for one CV.



Split range control



Valve position control

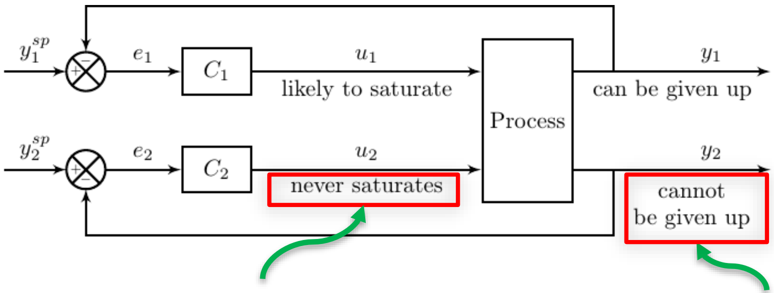


Different controllers with different setpoints

Design procedure for active constraint switching

Step A3: Identify possible and relevant active constraint switches

- **Case 3: MV to CV constraint switching**
  MV controlling a CV that may saturate; no extra MVs



MV that does not saturate

High priority CV: always controlled

Input saturation pairing rule

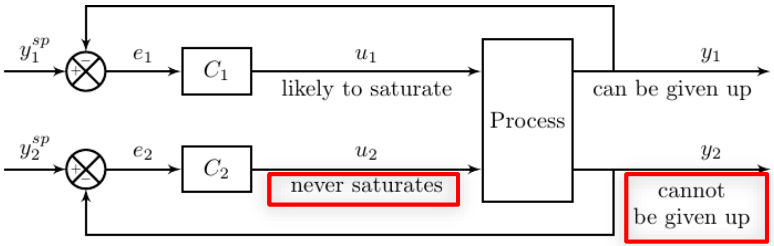
«an MV that is likely to saturate at steady-state should be paired with a CV that can be given up»

Low priority CV

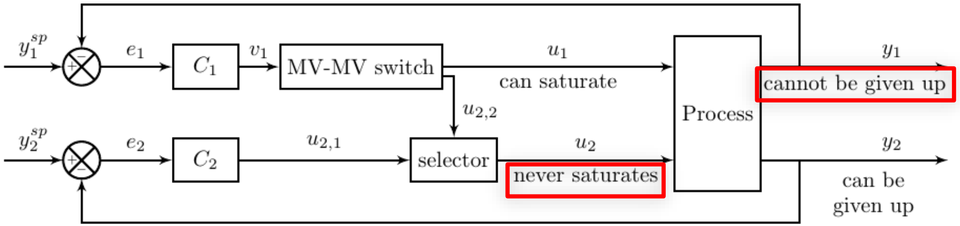
Design procedure for active constraint switching

Step A3: Identify possible and relevant active constraint switches

- **Case 3: MV to CV constraint switching**
  MV controlling a CV that may saturate; no extra MVs



Following input saturation pairing rule



NOT following input saturation pairing rule

Design procedure for active constraint switching

Step A3: Identify possible and relevant active constraint switches

- At nominal operation point all constraints are satisfied

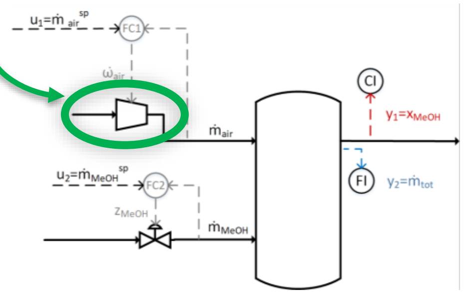
- **Constraint switch:**
  - Reach maximum air flow (u_1)

    ⬇

  - Lose a degree of freedom (**case 3**)
    - Must give up controlling the constraint with the lowest priority (desired throughput)

Variable	Units	Maximum	Nominal
$y_1 = x_{MeOH}$	kmol/kmol	0.10	0.10
$y_2 = \dot{m}_{tot}$	kg/h		26860
$u_1 = \dot{m}_{air}$	kg/h	25800	23920
$u_2 = \dot{m}_{MeOH}$	kg/h		2940

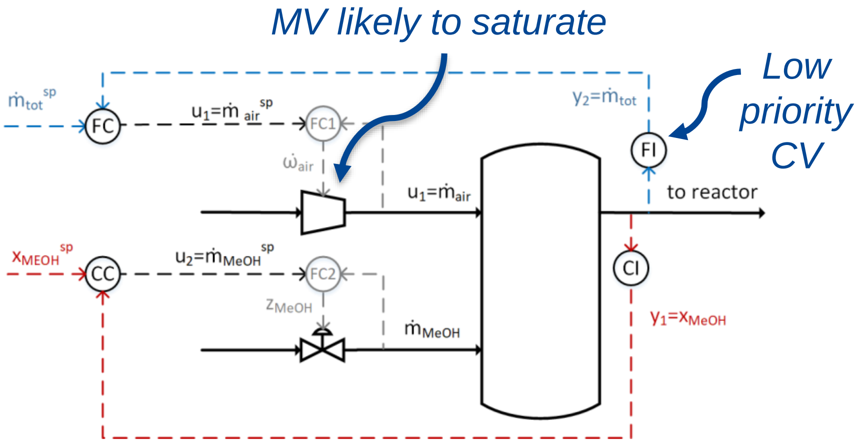
$u_1 = \dot{m}_{air}^{max}$



Design procedure for active constraint switching

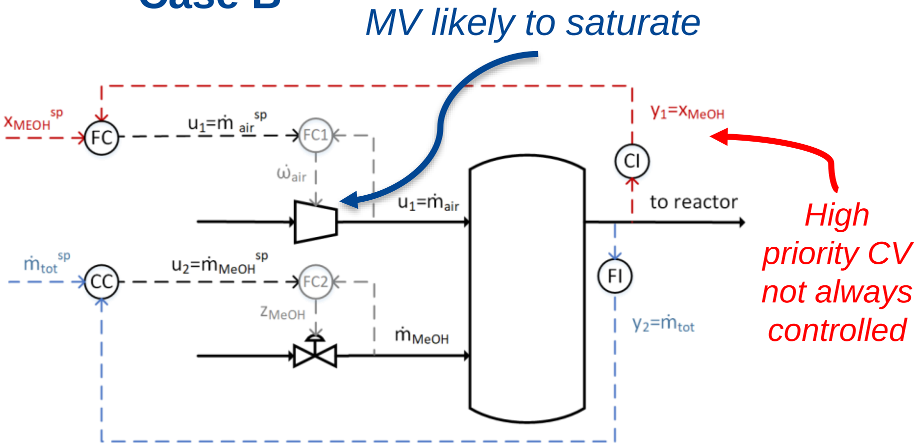
Step A4: Design control structure for optimal operation

Case A



Following input saturation pairing rule

Case B



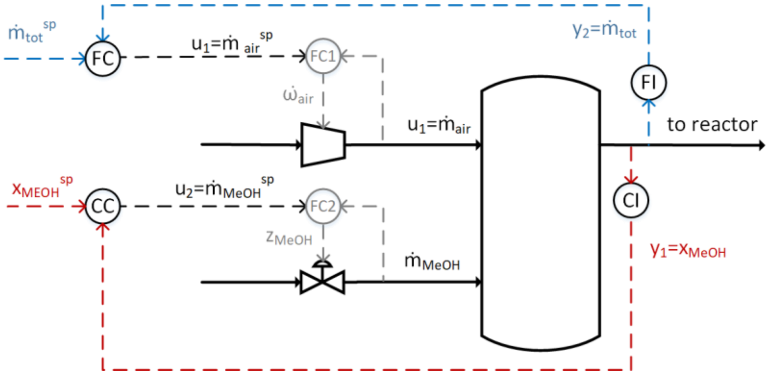
NOT following input saturation pairing rule

NTNU

Design procedure for active constraint switching

Step A4: Design control structure for optimal operation

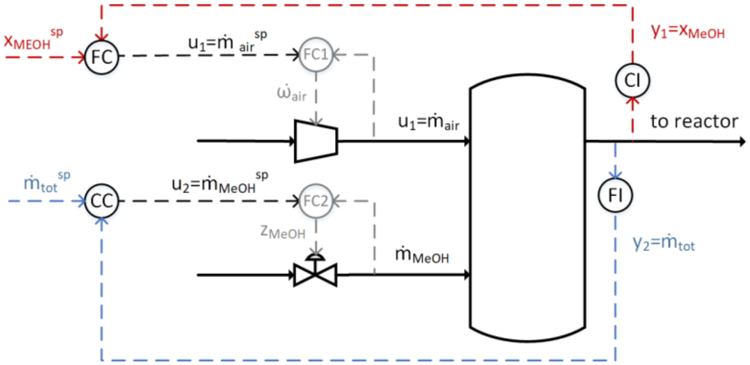
Case A



Following input saturation pairing rule

Case B

Needs MV to CV switching



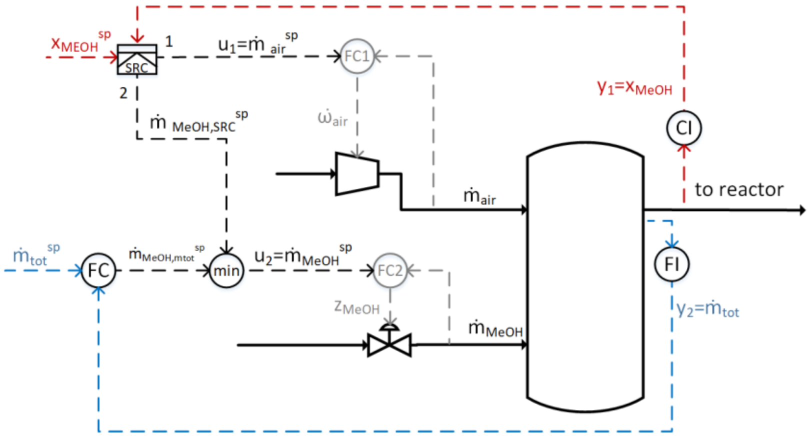
NOT following input saturation pairing rule

NTNU

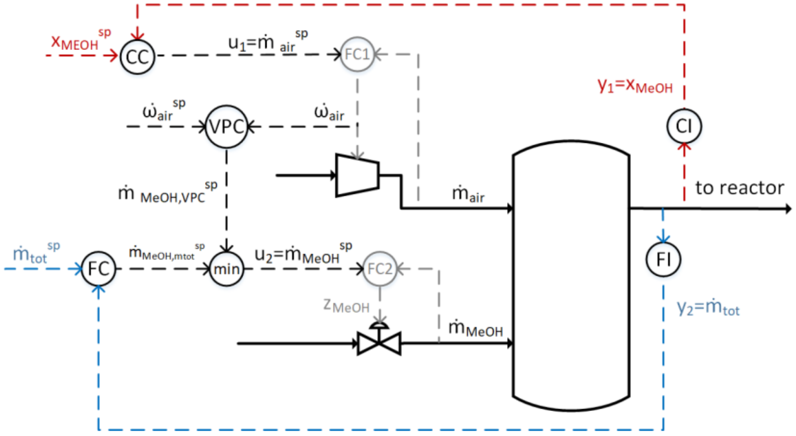
Design procedure for active constraint switching

Step A5: Design control structure to handle active constraint switches

Case B-SRC
Split range control+selector



Case B-VPC
Valve position control + selector



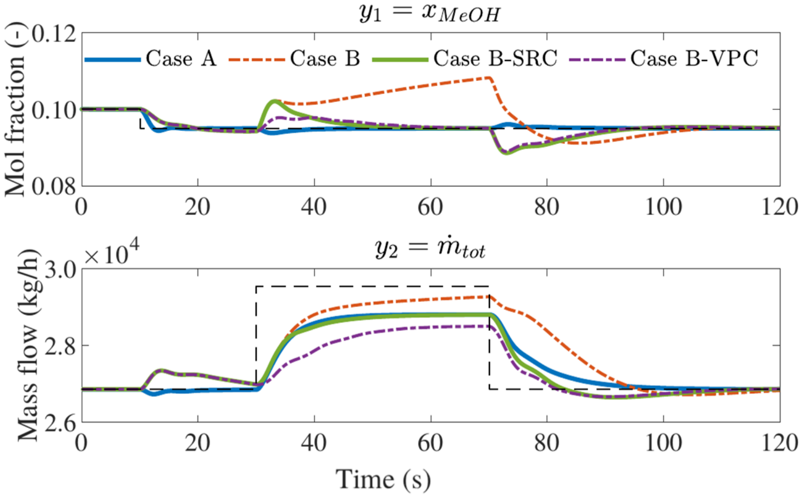
NTNU

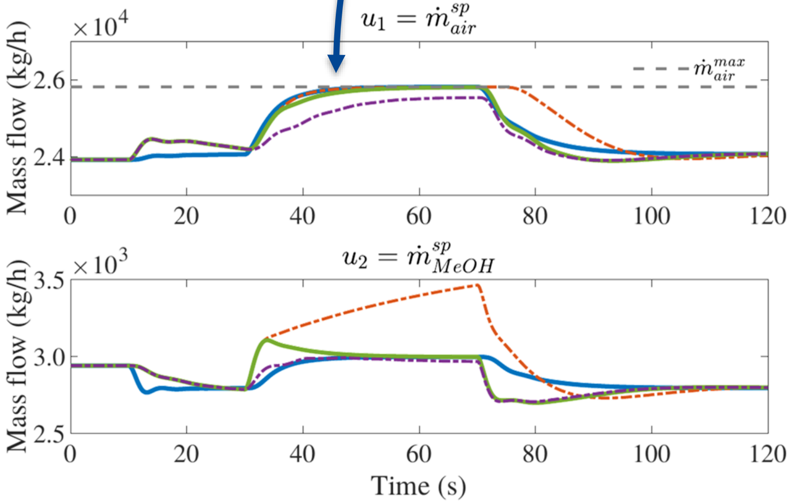
Design procedure for active constraint switching

Case study: Mixing of air and MeOH

MV1 is saturated:
lost degree of freedom

High priority CV: concentration





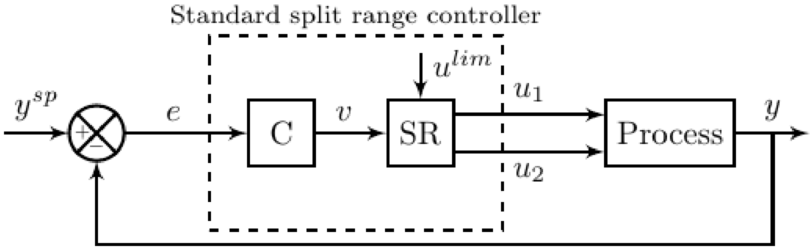
Low priority CV (throughput)

MV2 is not saturated:
It should be used to control the high priority CV

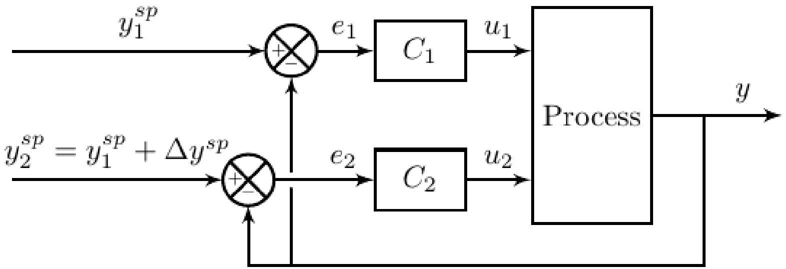
NTNU

MV to MV constraint switching

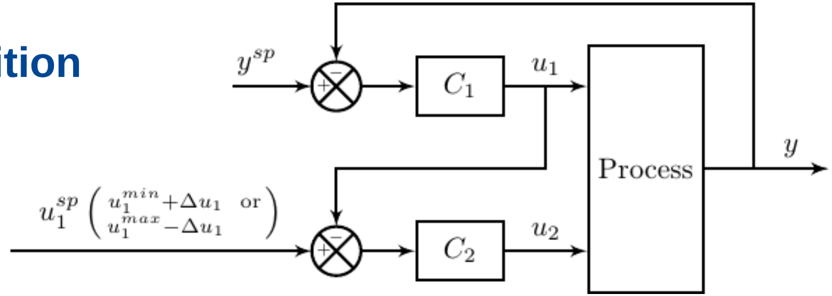
Split range control



Different controllers with different setpoints



Valve position control



Classical split range control

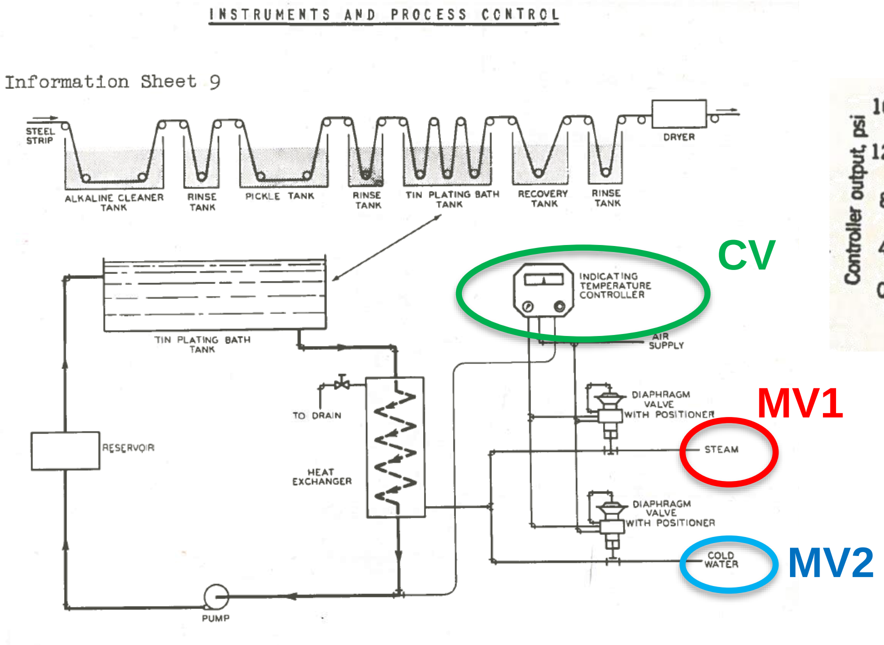
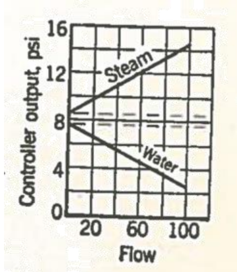
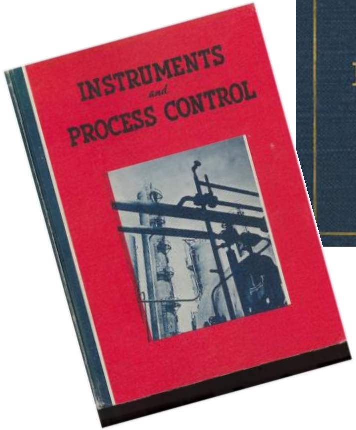
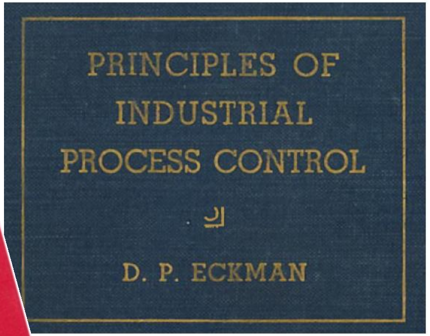


Fig. 125 - Temperature Control for a Tin Plating Bath
Courtesy of Taylor Instrument Companies



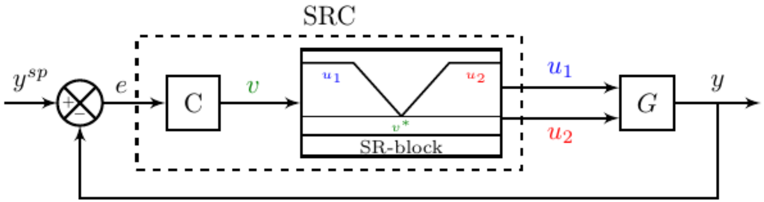


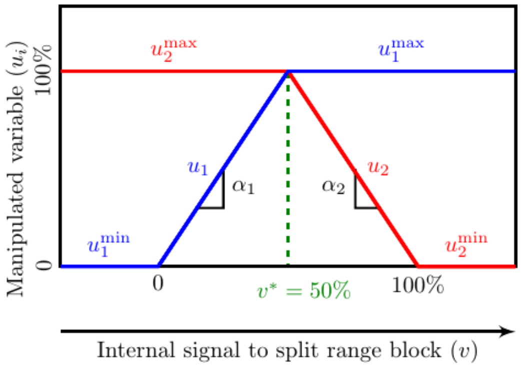


Eckman, D.P. (1945). Principles of industrial control, New York.

Monogram of Instruments and Process Control prepared at Cornell, NY, in 1945

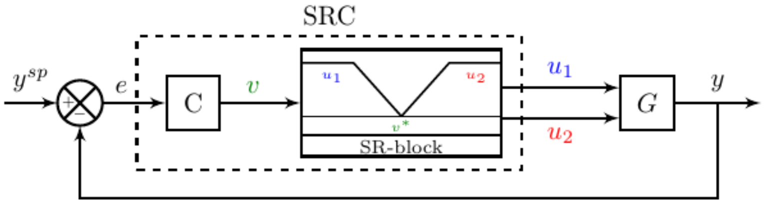
Classical split range control

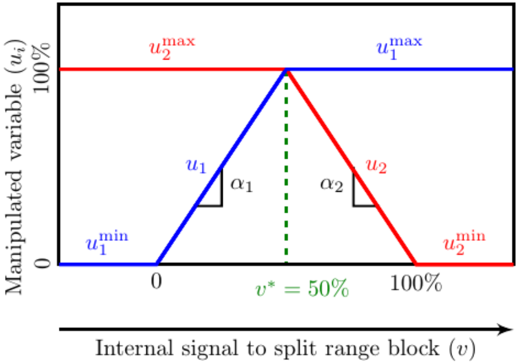




v	internal signal to split range block→ limited physical meaning
v^*	split value
u_i	controller output → physical meaning
α_i	gain from v to ui → slope

Classical split range control



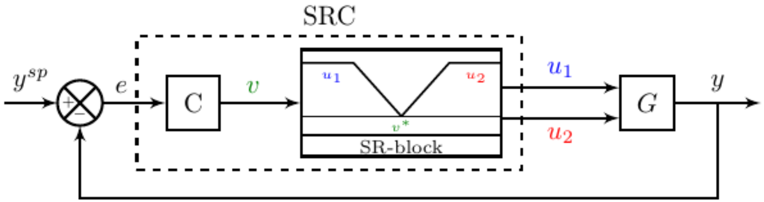


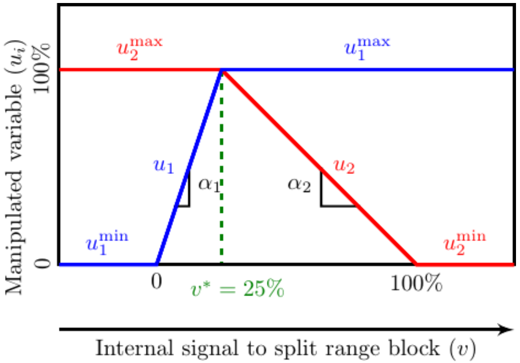
v	internal signal to split range block→ limited physical meaning
v^*	split value → degree of freedom
u_i	controller output → physical meaning
α_i	gain from v to ui → slope

$$u_i = u_{i,0} + \alpha_i \; v \quad \forall i \in \{1, \ldots, N\}$$

NTNU

Classical split range control





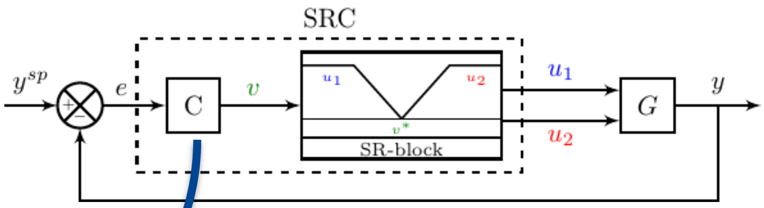
v	internal signal to split range block → limited physical meaning
v^*	split value → degree of freedom
u_i	controller output → physical meaning
α_i	gain from v to ui → slope

$$u_i = u_{i,0} + \alpha_i \; v \quad \forall i \in \{1, \ldots, N\}$$

Design of split range control: select slopes

Goal: get desired loop gain $|g\,C|$

at crossover frequency $\omega_c = \frac{1}{\tau_C}$



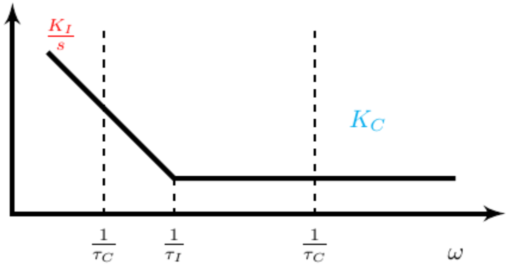
$$C(s) = K_C\left(1 + \frac{1}{\tau_I s}\right)$$

Fast process

$$K_{I,i} = \alpha_i K_I$$

Desired gain for u_i — DOF — Common gain in C

$K_I = \frac{K_C}{\tau_I}$



Slow process

$$K_{C,i} = \alpha_i K_C$$

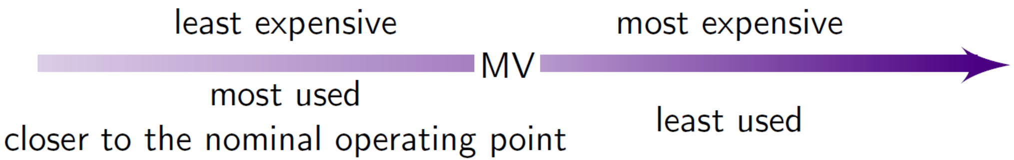
Desired gain for u_i — DOF — Common gain in C

Design of split range control: order of MVs

Define the desired operating point for every MV

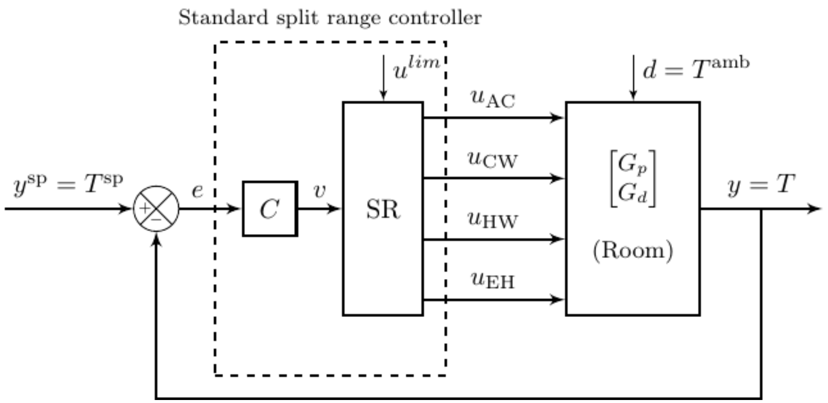
Group the MVs according to the effect on the CV

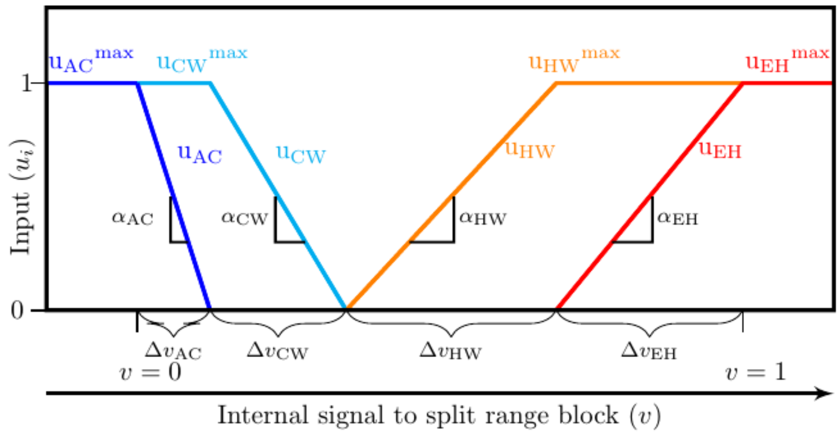
Within each group, define order of use



NTNU

Design of split range control





$u_1 = u_{AC}$: air conditioning (AC)

$u_2 = u_{CW}$: cooling water (CW)

$u_3 = u_{HW}$: heating water (HW)

$u_4 = u_{EH}$: electrical heating (EH)

NTNU

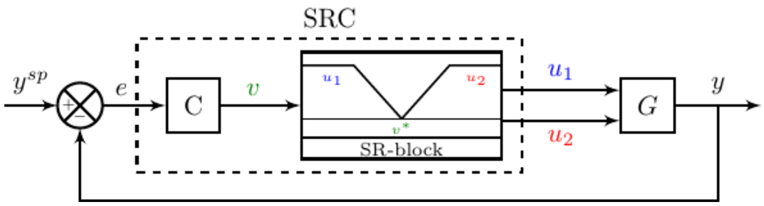
Classical split range control: a compromise

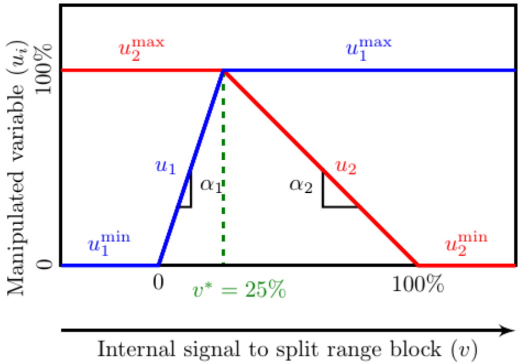
$$C(s) = K_C\left(1 + \frac{1}{\tau_I s}\right)$$

2 tuning parameters

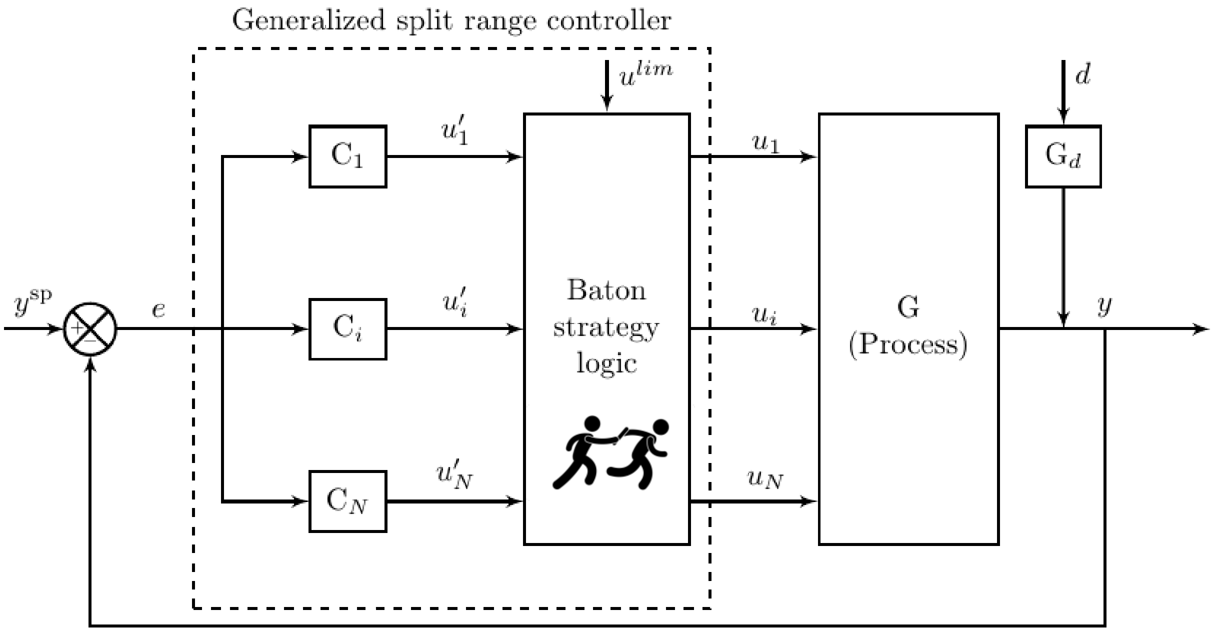
$$K_{C,i} = \alpha_i K_C$$

1 DOF



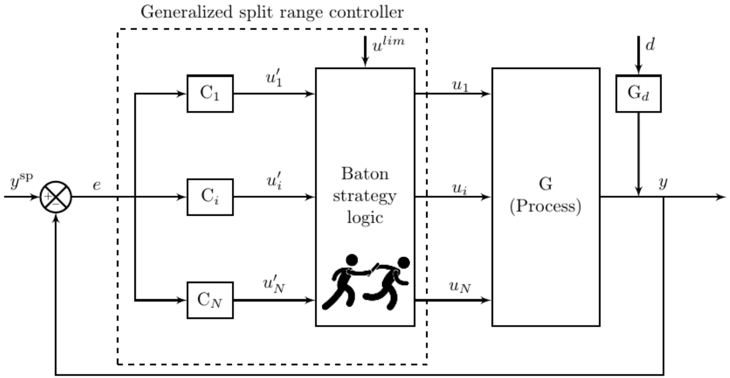


Generalized split range controller



NTNU

Generalized split range controller



Preliminary step:

- Define order of use of MVs (j=1,…,N)
- Tune controllers

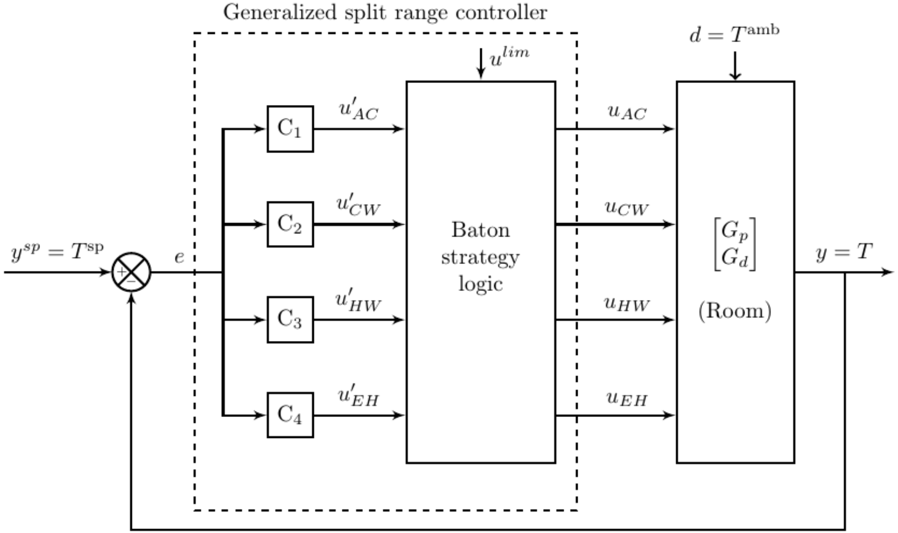
«Baton strategy» logic

k is the active input

- C_k computes u_k' (suggested value for u_k)
- If $u_k^{min} < u_k' < u_k^{max}$
  - Keep u_k active and $u_k \leftarrow u_k'$
  - Keep remaining u_i at limiting value
- else
  - Set $u_k = u_k^{min}$ or $u_k < u_k^{max}$, depending on the reached limit
  - New active input selected according to predefined sequence ($j = k-1$ or $j = k+1$)

The active input will *decide* when to switch and will remain active as long as it is not saturated.

Generalized split range controller



	Active input (input with *baton*, u_k)			
Value of u'_k	$u_1 = u_{\text{AC}}$	$u_2 = u_{\text{CW}}$	$u_3 = u_{\text{HW}}$	$u_4 = u_{\text{EH}}$
$u_k^{min} < u'_k < u_k^{max}$	keep u_1 active $u_1 \leftarrow u'_1$ $u_2 \leftarrow u_2^{max}$ $u_3 \leftarrow u_3^{min}$ $u_4 \leftarrow u_4^{min}$	keep u_2 active $u_1 \leftarrow u_1^{min}$ $u_2 \leftarrow u'_2$ $u_3 \leftarrow u_3^{min}$ $u_4 \leftarrow u_4^{min}$	keep u_3 active $u_1 \leftarrow u_1^{min}$ $u_2 \leftarrow u_2^{min}$ $u_3 \leftarrow u'_3$ $u_4 \leftarrow u_4^{min}$	keep u_4 active $u_1 \leftarrow u_2^{min}$ $u_2 \leftarrow u_1^{min}$ $u_3 \leftarrow u_3^{max}$ $u_4 \leftarrow u'_4$
$u'_k \geq u_k^{max}$	keep u_1 active (max. cooling)	baton to u_1 $u_1^0 = u_1^{min}$	baton to u_4 $u_4^0 = u_4^{min}$	keep u_4 active (max. heating)
$u'_k \leq u_k^{min}$	baton to u_2 $u_2^0 = u_2^{max}$	baton to u_3 $u_3^0 = u_3^{min}$	baton to u_2 $u_2^0 = u_2^{min}$	baton to u_3 $u_3^0 = u_3^{max}$

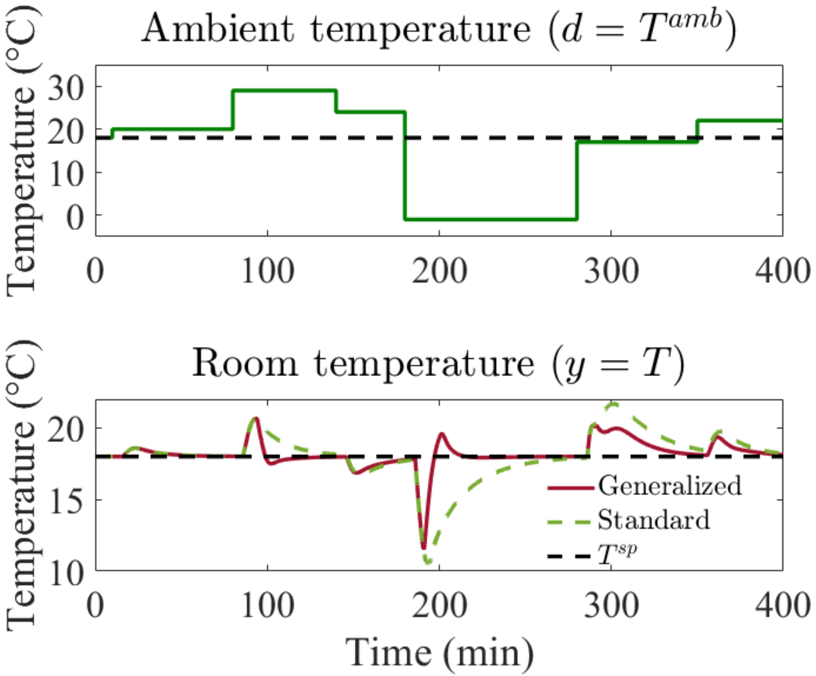
$u_1 = u_{AC}$: air conditioning (AC)

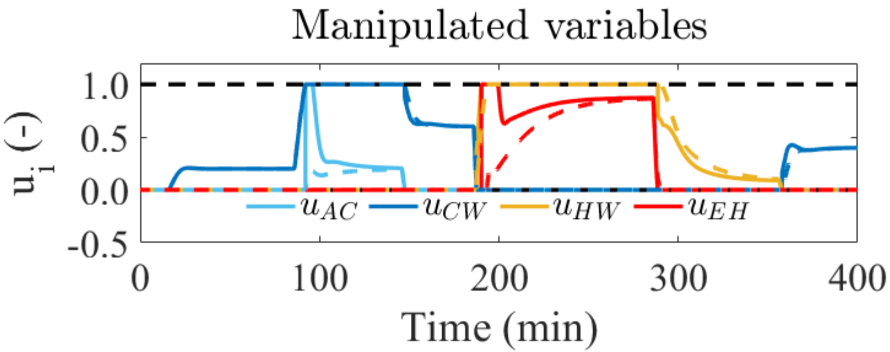
$u_2 = u_{CW}$: cooling water (CW)

$u_3 = u_{HW}$: heating water (HW)

$u_4 = u_{EH}$: electrical heating (EH)

Generalized vs standard split range controller





NTNU

Generalized split range controller: initialization

k

$$u'_k(t) = u_k^0 + K_{C,k}\left(e(t) + \frac{1}{\tau_{I,k}}\int_{t_b}^{t} e(t)\right)$$

This suggested input was not being applied while input k was not in use

This accumulated error is not due to the previous actions of input k

Generalized split range controller: initialization

k

Resetting:

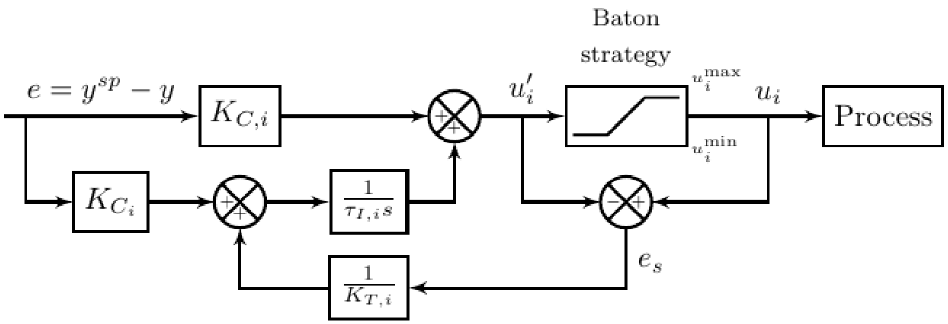
$$u'_k(t) = u_k^0 + K_{C,k}\left(e(t) + \frac{1}{\tau_{I,k}}\int_{t_b}^{t} e(t)\right)$$

$$u_k(t_b) = u_k^0 + K_{C,k}e(t_b)$$

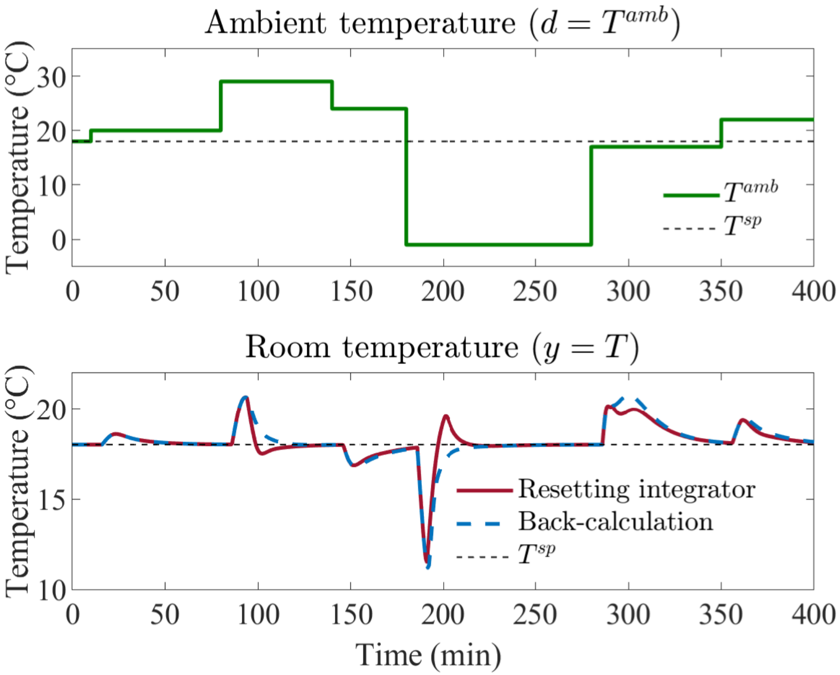
Initial action proportional to error at t_b

Generalized split range controller: initialization

Back-calculation:

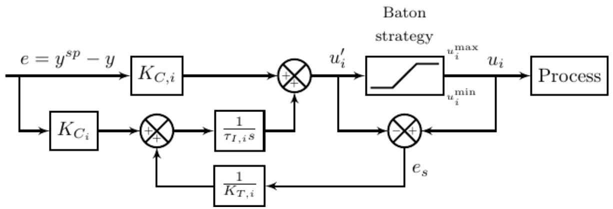


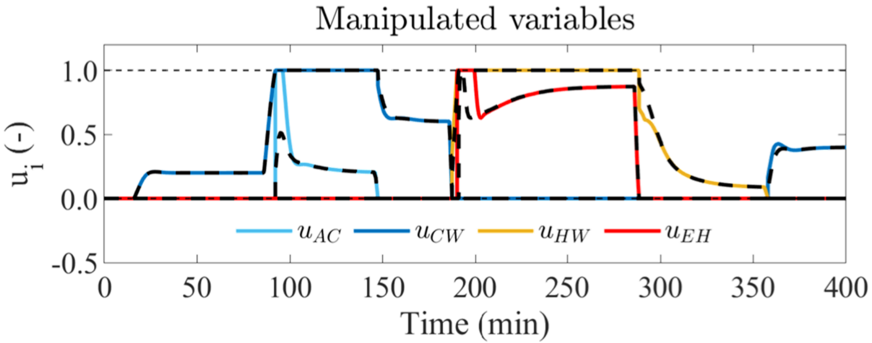
Generalized split range controller: initialization



Resetting: $u_k(t_b) = u_k^0 + K_{C,k} e(t_b)$

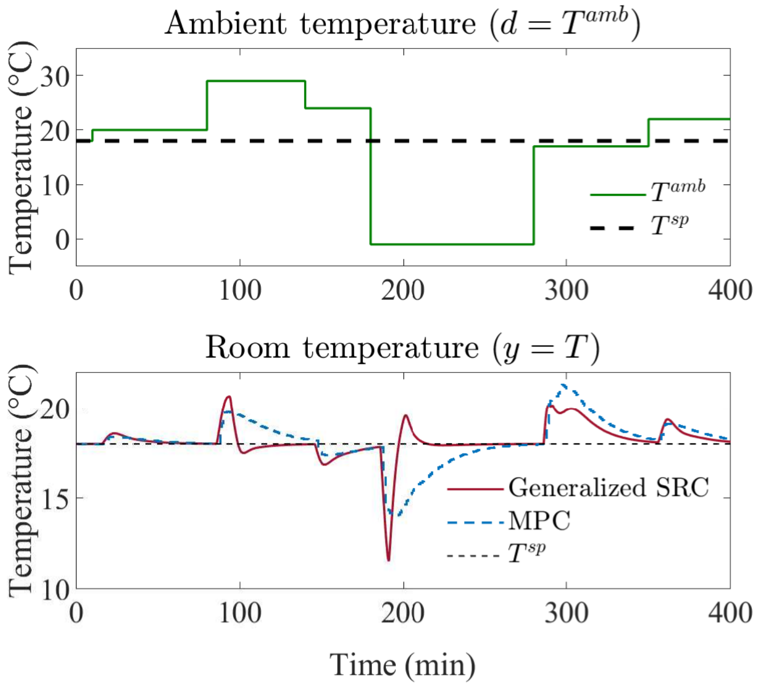
Back-calculation:

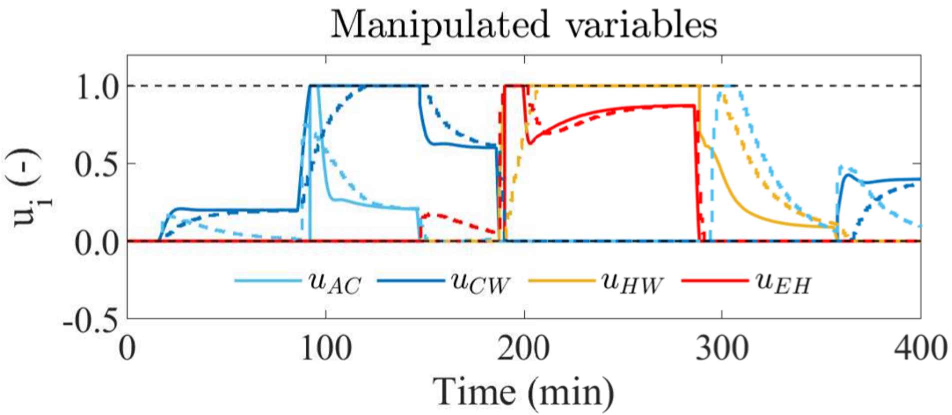




NTNU

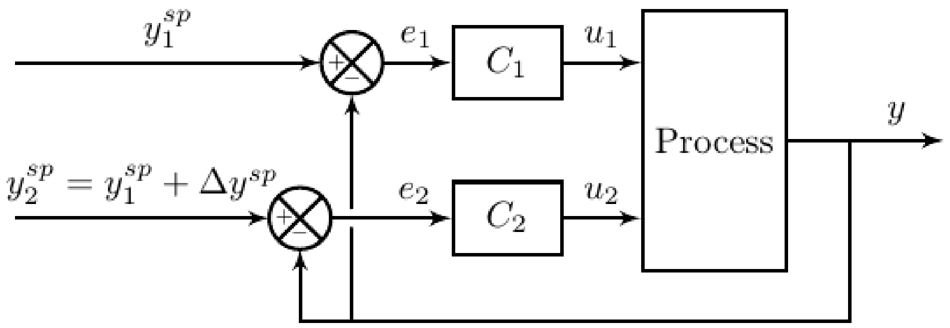
Generalized split range controller vs MPC





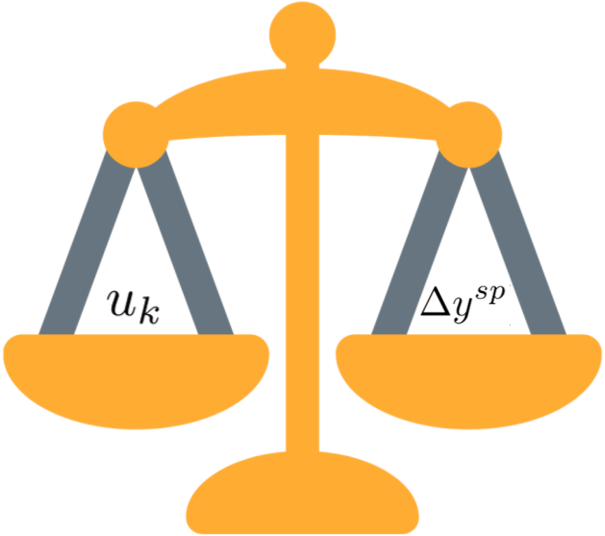
Multiple controllers with different setpoints

Does this make sense at any point?



Multiple controllers with different setpoints

Input usage



Setpoint deviation

$$J(u_k, \Delta y^{sp})$$

Multiple controllers with different setpoints: Optimal setpoint deviation

Linear for *u* and quadratic for Δ*y*

$$J = p_{u_k} u_k + p_y (y - y^{sp})^2 + c$$

Inputs* are a linear function of *output

$$u_i = k_i \; y + u_{i,0}$$

Multiple controllers with different setpoints: Optimal setpoint deviation

Linear for *u* and quadratic for Δ*y*

$$J = p_{u_k} u_k + p_y (y - y^{sp})^2 + c$$

Inputs* are a linear function of *output

$$u_i = k_i \; y + u_{i,0}$$

Cost when using u_k as input

$$J = p_{u_k} k_k \; y + p_y (y - y^{sp})^2 + c_k + p_{u_k} u_{k,0}$$

Multiple controllers with different setpoints: Optimal setpoint deviation

Linear for *u* and quadratic for Δ*y*

$$J = p_{u_k} u_k + p_y (y - y^{sp})^2 + c$$

Inputs* are a linear function of *output

$$u_i = k_i \; y + u_{i,0}$$

Cost when using u_k as input

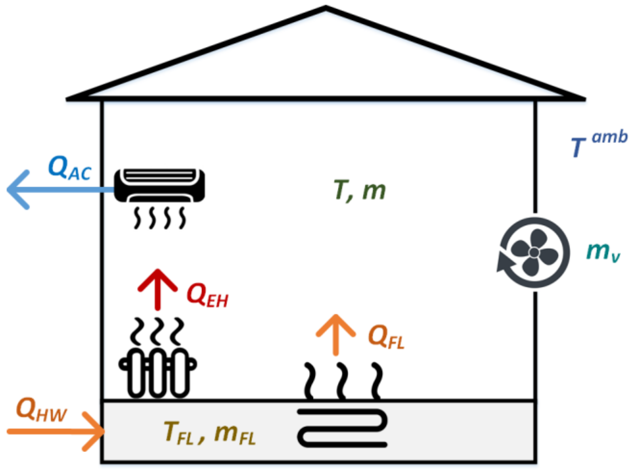
$$J = p_{u_k} k_k \; y + p_y (y - y^{sp})^2 + c_k + p_{u_k} u_{k,0}$$

$$\frac{dJ}{dy} = 0$$

Optimal setpoint deviation minimizing cost

$$\Delta y^{sp^*} = y^* - y^{sp} = -\frac{p_{u_k} k_k}{2 p_y}$$

Multiple controllers with different setpoints: Case study

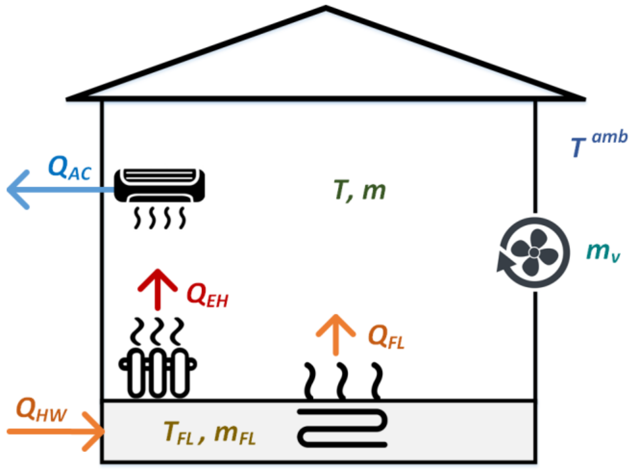


Q_{AC} : air conditioning

Q_{HW} : heating water

Q_{EH} : electrical heating

Multiple controllers with different setpoints: Room T



Q_{AC} : air conditioning

Q_{HW} : heating water

Q_{EH} : electrical heating

Cost: linear for *u* and quadratic for Δ*y*

$$J = \overbrace{\underbrace{p_{AC}Q_{AC}}_{p_1u_1} + \underbrace{p_{HW}Q_{HW}}_{p_2u_2} + \underbrace{p_{EH}Q_{EH}}_{p_3u_3}}^{J_{\text{energy}}} + \underbrace{p_T(T - T^{sp})^2}_{p_y(y-y^{sp})^2} \quad [\$/s]$$

Inputs **(Q_i) are a linear function of *output* (T)**

$$0 = \alpha(T^{amb} - T) + Q_{HW} + Q_{EH} - Q_{AC} \; [W]$$

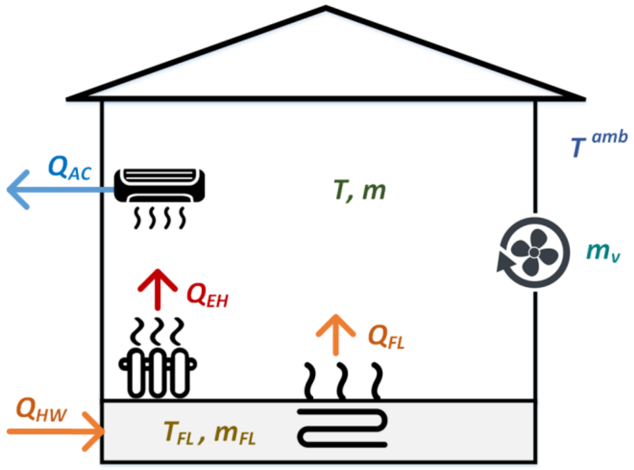
Optimal setpoint deviation minimizing cost

$$\Delta y^{sp,1} = T^{sp}_{AC} - T^{sp} = +\frac{\alpha p_{ac}}{2p_T}$$

$$\Delta y^{sp,2} = T^{sp}_{HW} - T^{sp} = -\frac{\alpha p_{hw}}{2p_T}$$

$$\Delta y^{sp,3} = T^{sp}_{EH} - T^{sp} = -\frac{\alpha p_{el}}{2p_T}$$

Multiple controllers with different setpoints: Room T



Q_{AC} : air conditioning

Q_{HW} : heating water

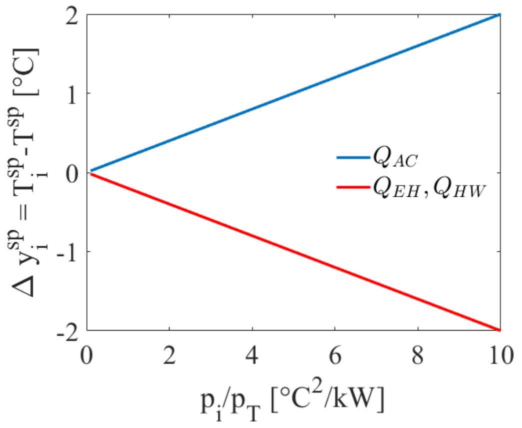
Q_{EH} : electrical heating

Optimal setpoint deviation minimizing cost

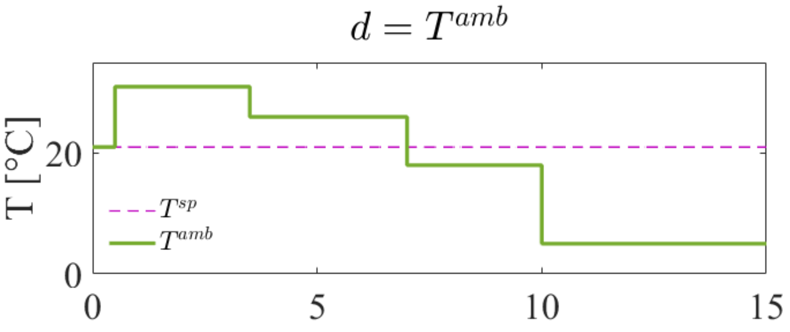
$$\Delta y^{sp,1} = T^{sp}_{AC} - T^{sp} = +\frac{\alpha p_{ac}}{2p_T}$$

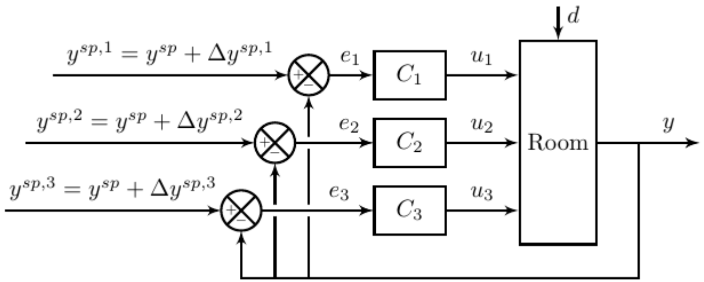
$$\Delta y^{sp,2} = T^{sp}_{HW} - T^{sp} = -\frac{\alpha p_{hw}}{2p_T}$$

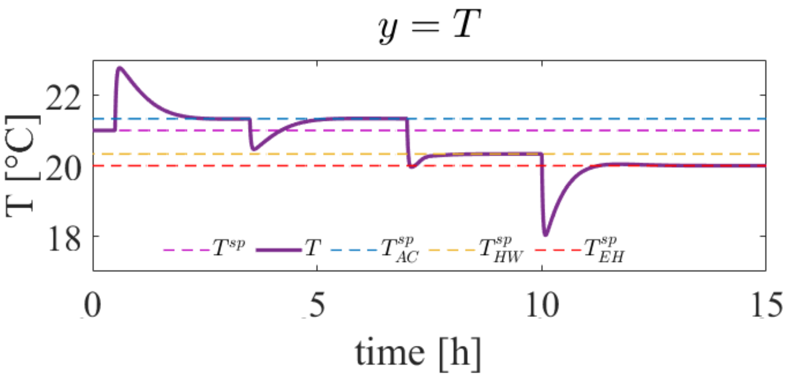
$$\Delta y^{sp,3} = T^{sp}_{EH} - T^{sp} = -\frac{\alpha p_{el}}{2p_T}$$

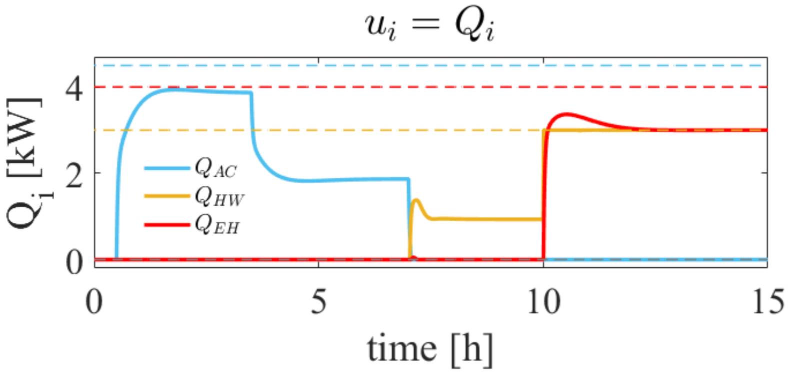


Multiple controllers with different setpoints: Room T



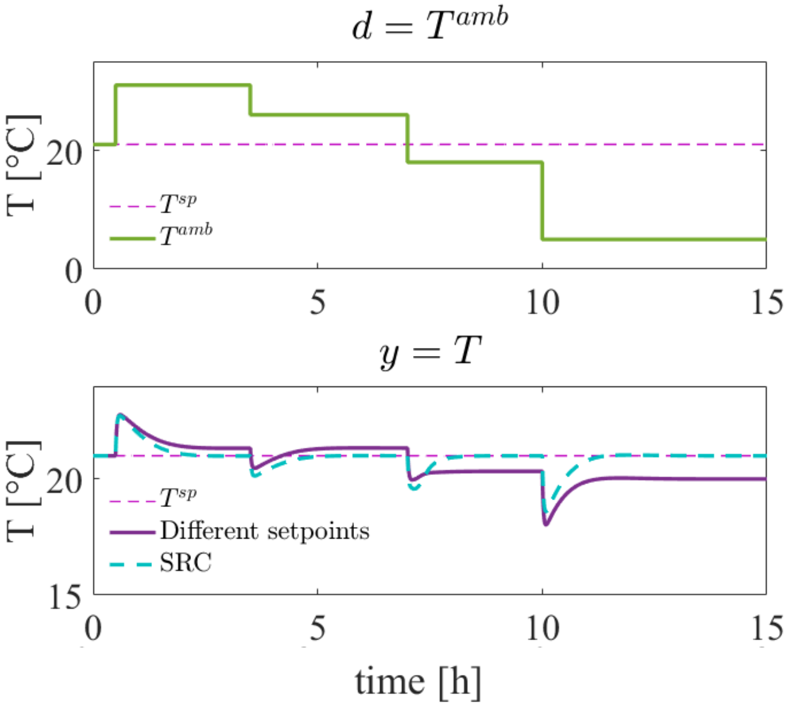


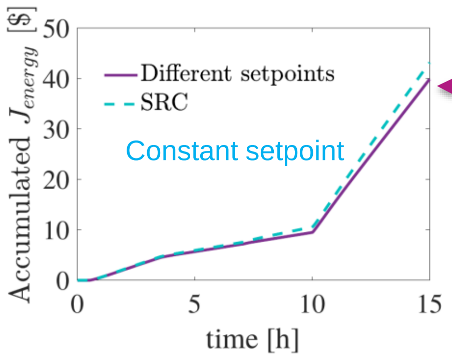




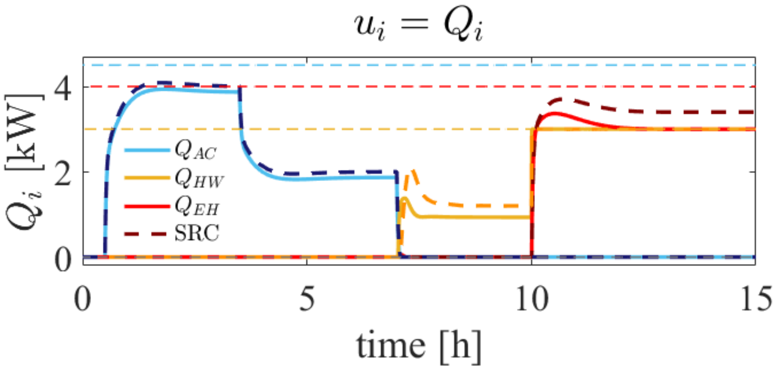
NTNU

Multiple controllers with different setpoints: Room T





Lower accumulated cost with minimum setpoint deviation



NTNU

Final comments

- Steady-state optimal operation may be easily achieved using PID-based control structures
  - Chapters 2,3,4: active constraint switching
  - Chapter 7: optimal setpoints

- Useful to systematically define control objectives, feasibility and tools
  - Priority list of constraints
  - Control structures available for each type of switch (CV-CV, MV-MV, MV-CV)

- Possible to improve performance of PID-based advanced control
  - Chapters 5, 6: design of split range controllers
  - Chapter 8: improved level control

One final comment

- The “gap” between theory and practice can be in both directions

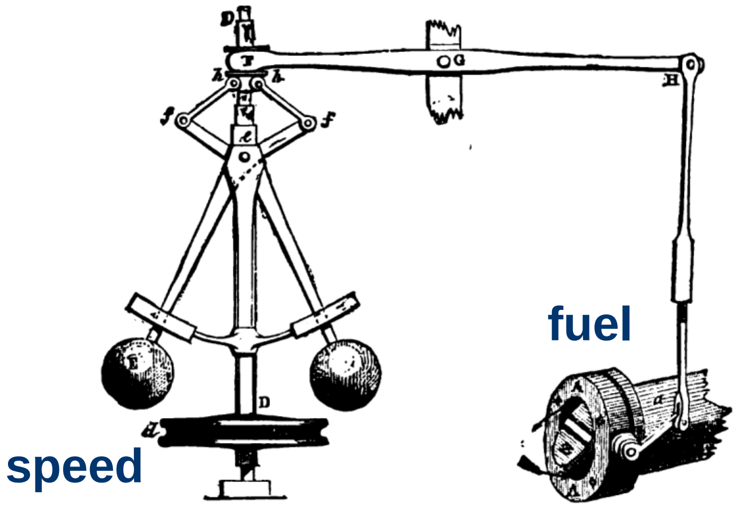


FIG. 4.—*Governor and Throttle-Valve.*

Centrifugal governor used in steam engines in the 1780’s:
Proportionally controls fuel flow to maintain engine speed.

Theoretical investigation started about a century later.

Åström, K. J., & Kumar, P. R. (2014). Control: A perspective. *Automatica*, *50*(1), 3–43.

NTNU

Systematic design of advanced control structures

Thank you for your attention!

Norwegian University of Science and Technology