



**NTNU – Trondheim**  
Norwegian University of  
Science and Technology

# Systematic design of advanced control structures

**Adriana Reyes Lúa**

Doctoral thesis for the degree of philosophiae doctor  
Trondheim, February 2020

Norwegian University of Science and Technology  
Faculty of Natural Sciences and Technology  
Department of Chemical Engineering

**NTNU**

Norwegian University of Science and Technology  
Thesis for the degree of philosophae doctor

Faculty of Natural Sciences and Technology  
Department of Chemical Engineering

2020

Adriana Reyes Lúa

ISBN 978-82-326-4494-0 (printed version)  
ISBN 978-82-326-4495-7 (electronic version)  
ISSN 1503-8181

Doctoral theses at NTNU, 2020:71

Printed by Skipnes

## Summary

PID-based advanced control structures are commonly used in process industries. However, their design mostly relies on the experience of the control and process engineers, and not always considers optimal operation. This thesis introduces a framework for the design of PID-based control structures, also considering steady-state optimal operation. In the proposed procedure, we use a *priority list of constraints* to make operation feasible. We also identify the relevant active constraint switches in the supervisory layer and how to handle each case. The active constraints can be on manipulated variables (MV, input) or on controlled variables (CV, output), and the switching cases are:

- *CV to CV constraint switching*: use selectors
- *MV to MV constraint switching*: use split range control or alternatively controllers with different setpoints or input (valve) position control.
- *MV to CV constraint switching*: use nothing if the *input saturation pairing rule* is followed; otherwise, use an MV to MV scheme and a selector to take over control when the main MV saturates.

Control structures that extend the operating range for the controlled variable by using more than one manipulated variable are of interest because they can handle *MV to MV constraint switching* and *MV to CV constraint switching*. This thesis gives a closer look to standard split range control and proposes a systematic design procedure for this control structure, considering the different dynamic effects of each MV on the CV. Standard split range control has intrinsic limitations in terms of tuning because it only has one common controller and one design parameter for each MV. To overcome this, we propose a generalized split range controller that uses a *baton strategy* to allow using independent controllers for each MV and the same setpoint. However, sometimes it is optimal to have different setpoints, and in this thesis we identify in which cases it is optimal to use different controllers with different setpoints and describe a procedure to find the optimal setpoint deviations. We illustrate the use of the analyzed and proposed control structures through several case studies. We also give some insight on when to use classical control schemes and when to use Model Predictive Control (MPC).



## Acknowledgements

First of all, I would like to thank Sigurd Skogestad, not only for giving me the opportunity to pursue a Ph.D. under his supervision but also for his support, feedback and encouragement for finishing this thesis. His dedication, understanding and passion for process control are admirable. I also want to thank the other professors of the PSE Group, Johannes Jäschke, Heinz Presig, Krister Forsman, Tore Haug-Warberg and Nadav Bar for encouraging open discussions about very diverse topics.

I would like to acknowledge the support of the NT Faculty at NTNU for financing my studies. I also appreciate the opportunity from SINTEF Energy to have the time to pursue my Ph.D., including a maternity leave. Thank you to HighEFF for including me as an associated Ph.D. and giving me the opportunity to meet other researchers and to frame my work within a larger project. One of the conference papers in this thesis was a product of a collaboration that started after one of the HighEFF meetings. I would also like to thank my opponents, Tore Hägglund, Bjørn Glemmestad and Heinz Preisig, for taking the time to evaluate this thesis.

The PSE Group is full of not only brilliant, but also very human and warm people. When I started, it was quite small, with Pablo, Vinicius, Vlad and Chriss. Then, Julian, Cansu, Sigve, Tobias and I started our Ph.D. more or less at the same time. The group grew a lot while I was there, with Adriaen, Eka, Christoph, Tamal, Dinesh, Timur, Bahareh, Pedro, Cristina, Mandar, Håkon, José, Allyne, Fabienne, Andrea, Robert and Fernando. Lunchtimes would not have been the same without the visiting researchers, Daniela, Prathak, Xing, Thomas, Simon, Ana, and Lucas. Special thanks to Mandar and Cristina for proof-reading several chapters of this thesis. Cristina, also thank you for co-authoring several of the papers, always giving thoughtful and valuable input and drawing many of the figures in the thesis.

I would also like to mention my parents, who always supported me for studying and doing what I wished. Also, thank you to my friends in Trondheim, who help me feel at home here.

Finally, the most important to me. Ernesto, you will not be able to read this soon, but thank you for understanding when I had to be working on this thesis instead of playing with you. Mario, thank you for supporting me, taking care of Ernesto, and trying to make it easier for me to finish this thesis. I love you both.



# Contents

<b>List of Figures</b>	<b>xi</b>
<b>List of Tables</b>	<b>xvii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation and scope . . . . .	1
1.2 Classical advanced control structures . . . . .	3
1.2.1 Cascade control . . . . .	3
1.2.2 Feedforward control . . . . .	4
1.2.3 Selectors . . . . .	6
1.2.4 Input (valve) position control (VPC) . . . . .	9
1.2.5 Input sequencing . . . . .	11
1.2.6 Anti-windup . . . . .	12
1.2.7 Other blocks . . . . .	12
1.3 Main contributions of this thesis . . . . .	13
1.4 Organization of the thesis . . . . .	14
1.5 Publications . . . . .	15
1.5.1 Journal papers . . . . .	15
1.5.2 Peer-reviewed conference papers . . . . .	16
<b>2 Active constraint switching with advanced control structures</b>	<b>19</b>
2.1 Introduction . . . . .	20
2.2 Design procedure for constraint switching using classical advanced control structures . . . . .	24
2.2.1 Step A1: Define the control objective, MVs and constraints . . . . .	24
2.2.2 Step A2: Organize the constraints in a list of priorities . . . . .	25
2.2.3 Step A3: Identify active constraint switches . . . . .	26
2.2.4 Step A4: Design control structure for base case operation . . . . .	27
2.2.5 Step A5: Design control structures for active constraint switching . . . . .	28
2.3 Case study I: Mixing of air and methanol . . . . .	34
2.3.1 Design of the supervisory control layer . . . . .	34

2.3.2	Simulations . . . . .	38
2.4	Case study II: Control structure for a distillation column . . . . .	40
2.4.1	Design of the supervisory control layer . . . . .	41
2.4.2	Simulations . . . . .	47
2.5	Discussion . . . . .	48
2.5.1	Optimal operation without a model . . . . .	48
2.5.2	"Opposing <i>pairing rules</i> " . . . . .	48
2.5.3	Alternative control structures . . . . .	50
2.6	Concluding remarks . . . . .	51
<b>3</b>	<b>Control structure for CO<sub>2</sub>-refrigeration cycle</b>	<b>53</b>
3.1	Introduction . . . . .	53
3.2	Description of the CO <sub>2</sub> -refrigeration system with heat recovery . . . . .	54
3.2.1	High-pressure side . . . . .	54
3.2.2	Heat recovery section . . . . .	56
3.3	Design of the supervisory control layer . . . . .	57
3.4	Concluding remarks . . . . .	61
<b>4</b>	<b>Classical advanced control vs MPC for MV saturation</b>	<b>63</b>
4.1	Introduction . . . . .	64
4.2	Optimal control in the presence of active constraint changes . . . . .	65
4.2.1	Advanced control structures in the supervisory layer . . . . .	66
4.2.2	Model predictive control in the supervisory layer . . . . .	66
4.3	Case Study: Optimal Control of a Cooler . . . . .	67
4.3.1	List of Priorities . . . . .	68
4.3.2	Active Constraint Regions . . . . .	69
4.4	Evaluation of classical advanced control structures for input saturation . . . . .	69
4.4.1	Split range control (SRC) . . . . .	70
4.4.2	Valve position control (VPC) . . . . .	70
4.4.3	Different controllers with different setpoints . . . . .	72
4.5	Model predictive control (MPC) for cooler case study . . . . .	73
4.6	Comparison of split range control with MPC . . . . .	75
4.7	Concluding remarks . . . . .	75
<b>5</b>	<b>Design of split range controllers</b>	<b>77</b>
5.1	Introduction . . . . .	77
5.2	Classical split range control . . . . .	78
5.3	Selection of slopes . . . . .	80
5.3.1	Controller tunings . . . . .	82
5.3.2	Bias . . . . .	83
5.4	A new procedure for designing the Split Range Block . . . . .	83
5.4.1	Ordering the use of MVs (Step S5) . . . . .	84



5.4.2	Systems with "fast" and "slow" MVs . . . . .	85
5.5	Case study . . . . .	86
5.5.1	Description of the system . . . . .	86
5.5.2	Design of the split range controller . . . . .	88
5.5.3	Simulations . . . . .	89
5.6	Discussion . . . . .	91
5.7	Concluding remarks . . . . .	92
<b>6</b>	<b>Generalized split range control</b>	<b>93</b>
6.1	Introduction . . . . .	93
6.2	Standard split range controller . . . . .	96
6.2.1	The split range block . . . . .	96
6.2.2	Slopes ( $\alpha_i$ ) in split range block . . . . .	97
6.3	Generalized split range control structure . . . . .	98
6.3.1	Proposed baton strategy . . . . .	98
6.3.2	Sequencing of inputs . . . . .	99
6.3.3	Baton strategy logic . . . . .	100
6.3.4	Anti-windup strategy . . . . .	100
6.4	Case Study: Control of room temperature . . . . .	101
6.4.1	Model . . . . .	101
6.4.2	Standard implementation of split range control . . . . .	102
6.4.3	Generalized implementation of split range control . . . . .	103
6.5	Discussion . . . . .	107
6.5.1	Alternative implementations of generalized split range control	107
6.5.2	Comparison with multiple controllers with different setpoints	108
6.5.3	Comparison of split range control with model predictive control	108
6.5.4	Anti-windup for generalized split range control . . . . .	110
6.5.5	Stability for controllers extending the operating range . . . . .	111
6.6	Conclusions . . . . .	113
<b>7</b>	<b>Multiple controllers with different setpoints</b>	<b>115</b>
7.1	Introduction . . . . .	115
7.2	Classical advanced control structures for more than one input for one output . . . . .	116
7.3	Optimal setpoint for each input . . . . .	119
7.4	Case study . . . . .	120
7.4.1	Optimal operation for temperature control . . . . .	121
7.4.2	Optimal setpoints for room temperature . . . . .	122
7.4.3	Three controllers with different setpoints . . . . .	123
7.4.4	Comparison with split range control . . . . .	125
7.5	Discussion . . . . .	126
7.6	Concluding remarks . . . . .	128

<b>8</b>	<b>Improved PI control for tank level</b>	<b>131</b>
8.1	Introduction . . . . .	131
8.2	Problem formulation . . . . .	132
8.3	Simple controller schemes for surge tanks . . . . .	133
8.4	Proposed control structure for improved liquid level control . . . . .	136
8.4.1	Tuning . . . . .	137
8.4.2	Simulation . . . . .	138
8.5	MPC Implementation . . . . .	139
8.6	Comparison of simple PIPP scheme with MPC . . . . .	140
8.6.1	Case 1: steps in the inflow . . . . .	141
8.6.2	Case 2: steps in the inflow plus noisy measurement . . . . .	142
8.6.3	Case 3: sinusoidal inflow . . . . .	143
8.6.4	Case 4: higher frequency sinusoidal inflow . . . . .	143
8.7	Comparison of performance of proposed PIPP structure and MPC . . . . .	144
8.8	Concluding remarks . . . . .	146
<b>9</b>	<b>Conclusions and future work</b>	<b>147</b>
9.1	Conclusions . . . . .	147
9.2	Future work . . . . .	149
	<b>Bibliography</b>	<b>151</b>
	<b>Appendices</b>	
<b>A</b>	<b>Model for mixing of air and MeOH in Chapter 2</b>	<b>163</b>
<b>B</b>	<b>Optimization of distillation column in Chapter 2</b>	<b>165</b>
<b>C</b>	<b>Supporting information for Chapter 4</b>	<b>167</b>
C.1	Cooler model . . . . .	167
C.2	Controller tuning . . . . .	167
<b>D</b>	<b>Supporting information for Chapter 6</b>	<b>171</b>
D.1	Parameters for standard split range controller for case study . . . . .	171
D.2	MPC implementation . . . . .	171
<b>E</b>	<b>Supporting information for Chapter 7 - Controllers with different setpoints</b>	<b>173</b>
E.1	DAE model for room temperature . . . . .	173
E.2	Tuning parameters for each input . . . . .	173
E.3	Design of the split range controller . . . . .	175
<b>F</b>	<b>Active constraint switching with baton strategy</b>	<b>177</b>





# List of Figures

1.1	Feedback control; $y^{sp}$ is the setpoint for the output ( $y$ ), $d$ is the disturbance, $u$ is the input, and $g$ is the process. . . . .	1
1.2	Cascade control structure. . . . .	3
1.3	Feedforward control structure. . . . .	4
1.4	Simple ratio control structure. . . . .	5
1.5	Two-way decoupling using the "inverted" implementation. . . . .	6
1.6	Use of a selector for the case with two CVs ( $y_1$ and $y_2$ ) and one MV ( $u$ ). . . . .	7
1.7	<i>mid</i> -selector with three inputs for steady-state range control. . . . .	7
1.8	Cascade controller with override <i>min</i> and <i>max</i> selector for dynamic operation. . . . .	8
1.9	Override <i>mid</i> -selector with three controllers. . . . .	9
1.10	Input (valve) position control to improve dynamic performance. . . . .	9
1.11	Input (valve) position control for floating control (of $y$ ). . . . .	10
1.12	Input (valve) position control for range extending (alternative 3 in Section 1.2.5). . . . .	11
1.13	Alternative 1 for range extending: (standard) split range control. . . . .	11
1.14	Alternative 2 for range extending: different controllers with different setpoints. . . . .	12
1.15	Antiwindup with bumpless transfer: input tracking with back-calculation for input $u_i$ . . . . .	12
2.1	Typical control hierarchy in a process plant. . . . .	20
2.2	CV to CV switching using a selector for the case with two CVs ( $y_1$ and $y_2$ ). . . . .	29
2.3	Typical example of CV to CV switching based on controller output signals. The regulatory layer is dimmed in gray. . . . .	29
2.4	MV to MV constraint switching using split range control (SRC) for a case with two MVs ( $u_1$ and $u_2$ ) and one CV ( $y$ ). . . . .	30
2.5	MV to MV constraint switching using two controllers with different setpoints. . . . .	31
2.6	MV to MV constraint switching using input (valve) positioning control. . . . .	32
2.7	MV to CV switching for the case when the input saturation rule is followed, so control of $y_1$ can be given up. . . . .	32

2.8	MV to CV switching for the case when the input saturation rule is <i>not</i> followed; so control of $y_1$ cannot be given up. . . . .	33
2.9	Alternative scheme proposed by Shinkskey (1978) for MV to CV switching when the input saturation rule is <i>not</i> followed. . . . .	33
2.10	Case A: control structure for mixing of MeOH and air following the <i>input saturation pairing rule</i> . The (lower) regulatory control layer is dimmed in gray. . . . .	37
2.11	Case B-SRC. Control structure for mixing of MeOH and air when <i>not</i> following the <i>input saturation pairing rule</i> using split range control with a <i>min</i> selector. . . . .	37
2.12	Case B-VPC. Alternative control structure for mixing of MeOH and air in Case B, using input (valve) positioning control (VPC). . . . .	38
2.13	Comparison of control structures for mixing of MeOH and air. The best results are achieved with Case A and case B-SRC. . . . .	39
2.14	Distillation column with regulatory control layer in gray. $u_1 = V$ and $u_2 = L$ are MVs for the supervisory control layer. . . . .	42
2.15	Active constraint regions for binary distillation column with the bottom as valuable product. . . . .	44
2.16	Base case control structure for distillation column (region I). . . . .	45
2.17	Control structure for distillation column for regions I, II and III. . . . .	46
2.18	Control structure for distillation column for all regions (I, II, III, IV). . . . .	47
2.19	Simulation for structure in Fig. 2.18 for case study II. . . . .	49
2.20	Cost for distillation column case study (which should be minimized). . . . .	50
2.21	Alternative control structure for distillation column, all regions. This structure behaves differently from Fig. 2.18 in region IV. . . . .	51
3.1	$CO_2$ -refrigeration system with parallel compression and heat recovery. There are seven available manipulated variables (MV). . . . .	55
3.2	Pressure-enthalpy diagram of $CO_2$ -refrigeration system with parallel compression. . . . .	55
3.3	Basic control structure for the $CO_2$ -refrigeration system with heat recovery. . . . .	60
3.4	Control structure for the $CO_2$ -refrigeration system with heat recovery and parallel compression. This control structure handles the three operating regions. . . . .	61
4.1	Cooler with temperature and flow control <sup>1</sup> . . . . .	68
4.2	Active constraint regions for the studied cooler. . . . .	69
4.3	Split range control structure for cooler. . . . .	70
4.4	Disturbance rejection simulation results with split range control for cooler (Fig. 4.3). . . . .	71
4.5	Valve positioning control (VPC) structure for cooler. . . . .	71
4.6	Disturbance rejection simulation results with VPC for cooler (Fig. 4.5). . . . .	72

4.7	Control structure using two controllers for cooler. . . . .	73
4.8	Disturbance rejection simulation results using two controllers for cooler (Fig. 4.7). . . . .	73
4.9	Simulation results for MPC and SRC. . . . .	75
5.1	Block diagram for standard implementation of split range control with two MVs and one CV. . . . .	78
5.2	Split range block with extended $v$ , giving the relationship between $v$ and two MVs ( $u_1$ and $u_2$ ). In general, the split value ( $v^*$ ) should not be fixed at 50%. . . . .	79
5.3	Significant controller parameters according to frequency. . . . .	82
5.4	Split range block for room temperature control with heating ( $u_4$ ), cooling ( $u_2$ ) and ventilation flow ( $u_1, u_4$ ) as MVs. . . . .	86
5.5	Block diagram of split range control for controlling room temperature. . . . .	87
5.6	Standard split range block for room temperature control with air conditioning (AC), cooling water (CW), hot water (HW), and electric heating (EH). . . . .	90
5.7	Closed-loop response for changes in temperature set-point ( $T^{ref}$ ). . . . .	91
5.8	Closed-loop response for changes in ambient temperature ( $T^{amb}$ ). . . . .	92
6.1	Classical structure 1: Standard implementation of split range control with two inputs ( $u_i$ ) and one output ( $y$ ). A typical SR-block is shown in Fig. 6.4. Note that $v$ is a non-physical internal signal, whereas $u_i$ is the physical input. $u^{lim}$ contains information about the maximum and minimum input values, which the SR-block uses to decide on the input switching. . . . .	94
6.2	Classical structure 2: Two controllers and two inputs for the same output ( $y$ ), each controller with a different setpoint ( $y^{sp,1}$ and $y^{sp,2}$ ). . . . .	94
6.3	Classical structure 3: Input (valve) position control to control one output ( $y$ ) with two inputs ( $u_1$ and $u_2$ ). . . . .	94
6.4	Typical split range (SR) block for Fig. 6.1. . . . .	97
6.5	Proposed generalized split range control using the baton strategy. Each controller computes a suggested input $u'_i$ and the baton logic decides on the actual input $u_i$ . . . . .	98
6.6	Baton strategy for relay. . . . .	99
6.7	Block diagram for standard split range control for room temperature control. The SR block is shown in Fig. 6.8. . . . .	102
6.8	Standard split range block for room temperature control with air conditioning (AC), cooling water (CW), hot water (HW), and electric heating (EH); SR block in Fig. 6.7. . . . .	103
6.9	Generalized split range control solution for room temperature control. . . . .	104

6.10	Comparison of standard and generalized split range controller (SRC) for case study. The standard SRC structure is shown in Fig. 6.7 and the generalized structure is shown in Fig. 6.9. . . . . .	106
6.11	Comparison of standard MPC with generalized split range control (SRC) for room temperature. The dashed lines correspond to MPC and the solid lines correspond to the strategy proposed in this chapter, which is also depicted in Fig. 6.10. . . . .	109
6.12	Antiwindup with bumpless transfer: input tracking with back-calculation for input $u_i$ . . . . .	111
6.13	Comparison of two anti-windup strategies for the generalized split range control structure in Fig. 6.9. The dashed lines correspond to back-calculation (Fig. 6.12) and the solid lines correspond to the strategy of resetting the integrator (Eq. (6.1)), which is also depicted in Fig. 6.10. . . . .	112
7.1	Classical split range control (SRC) with two inputs and one output. A typical split range block (SR-block) is shown in Fig. 7.2. . . . .	117
7.2	Split range block (SR-block) in Fig. 7.1 for a case in which the two inputs have different steady-state gain. . . . .	117
7.3	Block diagram for input (valve) positioning control for extending the steady-state range. . . . .	118
7.4	Block diagram for two controllers with different setpoints. This is the control structure studied in this chapter. . . . .	118
7.5	Room with three available (independent) inputs ( $u_i = Q_i$ ) for temperature control ( $y = T$ ). . . . .	121
7.6	Effect of comfort penalty ( $p_T$ ) and input usage ( $p_i$ ) on optimal setpoint deviation ( $\Delta y^{sp,i}$ ). . . . .	124
7.7	Block diagram for three controllers with different setpoints, one for each input; $y = T$ , $d = T^{amb}$ and $u_i = Q_i$ , where 1 = AC, 2 = HW, 3 = EH. . . . .	124
7.8	Simulation results using three different controllers (Fig.7.7), with the optimal setpoint deviations in Table 7.3. . . . .	125
7.9	Block diagram for split range control (SRC) with three inputs and one output. The SR block is shown in Fig. E.1; $y = T$ , $d = T^{amb}$ and $u_i = Q_i$ , where 1 = AC, 2 = HW, 3 = EH. . . . .	126
7.10	Comparison of simulation results with three controllers with different setpoints (Fig. 7.7) and split range control (SRC, Fig. 7.9). The simulation with three controllers is shown with solid lines and the simulation with SRC is shown with dashed lines. . . . .	127
7.11	Comparison of accumulated cost of energy ( $J_{energy}$ ) using three controllers with different setpoints (Fig. 7.7) and split range control (SRC, Fig. 7.9). . . . .	128
8.1	Nonlinear relationship (solid lines) between level and outflow for case with three P-controllers. . . . .	134



8.2	Level control with three P-controllers. $K_c = 0.33$ , $K_{c,max} = K_{c,min} = 6.67$ .	134
8.3	Level control with PI-controller. . . . .	136
8.4	Proposed PIPP control structure with one PI-controller and two P-controllers to track safety limits. . . . .	137
8.5	Simulation of proposed PIPP control structure. . . . .	139
8.6	Case 1: Response of PIPP structure and MPC with step changes in inflow ( $q_{in}$ ). . . . .	141
8.7	Case 2: Response of PIPP structure and MPC with step changes in inflow ( $q_{in}$ ) and measurement noise. . . . .	142
8.8	Case 3: Response of proposed PIPP control structure and MPC with sinusoidal inflow ( $q_{in}$ ). . . . .	143
8.9	Case 4: Response of proposed PIPP control structure and MPC with higher frequency sinusoidal inflow ( $q_{in}$ ). . . . .	144
B.1	Contour plot for objective function $J^{opt}$ for distillation column. . . . .	165
B.2	Contour plot for $x_D^{opt}$ for distillation column, with active constraint regions indicated. . . . .	166
B.3	Linear approximation of $x_D^{opt}$ as a function of $p_V$ in Region I. . . . .	166
C.1	Lumped model for cooler. . . . .	167
C.2	Open-loop response in $T_H$ for a step in $F_C$ and $F_H$ . . . . .	168
C.3	Response in $F_C$ for a step in $F_H$ ; used to tune the valve position controller. . . . .	169
E.1	Split range block for controlling room temperature with one source of cooling (AC) and two sources of heating (HW and EH). This is the SR block in Fig. 7.9. . . . .	175



# List of Tables

2.1	Maximum and nominal values for case study I. . . . .	35
2.2	Data for distillation case study. . . . .	41
2.3	Pairings in Fig. 2.18 for each of the active constraint regions. . . . .	47
4.1	Constraints for the cooler system. . . . .	68
5.1	Parameters for $G_{p,i}(s)$ from $u_i$ to $T$ . . . . .	87
5.2	Tuning parameters for each MV. . . . .	88
5.3	Values for $\alpha_i$ , $\Delta v_i$ and $u_{i,0}$ . . . . .	89
6.1	Parameters for $G_{p,i}(s)$ from $u_i$ to $y = T$ and $G_d(s)$ from $d = T^{amb}$ to $y = T$ . . . . .	102
6.2	Tuning parameters in room temperature control. . . . .	103
6.3	Baton strategy logic for case study. . . . .	105
6.4	Integral absolute error (IAE) for the case study with alternative controllers. . . . .	110
7.1	Ranges for available inputs ( $u_i$ ). . . . .	121
7.2	Parameters for cost function for optimization. . . . .	122
7.3	Optimal deviations for the three available inputs. . . . .	123
8.1	Deviation of the level from its setpoint. . . . .	145
8.2	Deviation of the outflow from the steady inflow setpoint. . . . .	145
8.3	Total outflow variation. . . . .	146
A.1	First order transfer functions for mixing of MeOH and air. . . . .	163
C.1	Tuning parameters for the cooler case study. . . . .	168
D.1	Values for $\alpha_i$ , $\Delta v_i$ and $u_{i,0}$ . . . . .	171
E.1	Parameters for room model. . . . .	174
E.2	Parameters for first-order transfer functions for the available inputs. . . . .	174
E.3	PI tuning parameters for the available inputs. . . . .	175

E.4	Values for the slopes $\alpha_i$ , $\Delta v_i$ and $u_{i,0}$ in the split range block. . . . .	175
-----	---	-----

# Chapter 1

## Introduction

In this chapter we describe the motivation and define the scope of the thesis. We also give an overview of the thesis and the list of publications on which it is based.

### 1.1 Motivation and scope

Process control is used to stabilize a process and achieve optimal operation. Closed-loop feedback control<sup>1</sup>, shown Fig. 1.1, has been extensively used since the early development of control and the controller ( $C$ ) is usually a PID-controller (Eq. (1.1)) (Åström and Hägglund, 2006).

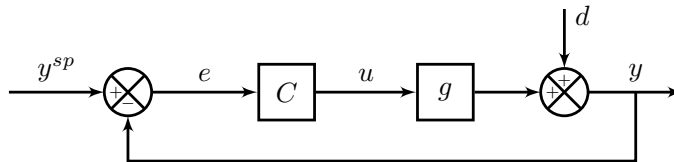


Figure 1.1: Feedback control;  $y^{sp}$  is the setpoint for the output ( $y$ ),  $d$  is the disturbance,  $u$  is the input, and  $g$  is the process.

$$u(t) = u^0 + K_C \left( e(t) + \frac{1}{\tau_I} \int_0^t e(t) + \tau_d \frac{de(t)}{dt} \right) \quad (1.1)$$

Early development of control was from the 1870s through the 1920's (Åström and Hägglund, 2006) and still in the late 1930s, process control was mostly practical, mainly due to hardware limitations (Åström, 1999). Automatic control emerged as a discipline in the mid 1940s and one of the first books on process control was written by Eckman (1945). Since then, PID-control has been applied in all process industries.

---

<sup>1</sup>Feedback control refers to the case when we (generally) measure the controlled variable (CV, output  $y$ ) and keep adjusting the manipulated variable (MV, input  $u$ ) to maintain  $y = y^{sp}$ , based on the error  $e = y^{sp} - y$ .

In the last decades, advanced control based on model-based optimization routines such as model predictive control (MPC) has also been applied in industry with good results (Qin and Badgwell, 2003). However, MPC is based on a dynamic model of the process, which is not always available or accurate (Seborg, 1999). Moreover, when MPC is implemented in industry, PID controllers are normally required in the regulatory control layer for stabilization<sup>2</sup>. Therefore, the majority ( $\approx 90\%$ ) of automatic control loops in process industries still rely on various forms of PID-controllers due to its simplicity and robustness and because classical advanced control techniques can provide cost-effective solutions for important classes of problems (Seborg, 1999; Åström and Hägglund, 2001).

*Classical advanced control* uses different control structures to define the setpoints for the regulatory layer when a single-loop PID-controller is not sufficient. *Classical advanced control structures* have been in use for at least 75 years (Eckman, 1945; Fink, 1945), and are still commonly used in industry for multivariable control.

Foss (1973) pointed out that *the central issue to be resolved by the new theories of chemical process control is the determination of control system structure*. He then goes further by stating that *an acceptable, broadly applicable solution to the control structure problem cannot be achieved by the dreaming up of a number of candidate configurations for a given process and then testing them... the method must have its basis in a broadly applicable representation of the process dynamics and control objectives*. He concludes by saying that to close the gap between theory and practice, we need to have knowledge of the process as well as the theory.

However, almost fifty years after Foss (1973) identified this gap between theory and practice, some aspects of the implementation of PID-based control structures still have not received much attention and are not covered in textbooks, despite being widely used in practice (Åström, 1999; Hägglund and Guzmán, 2018). For example, actual implementation of control structures conventionally used in industry can have a larger complexity for configurations with selectors (Seborg, 1999). Being so ubiquitous, an appropriate implementation of classical advanced control structures can have a significant economic impact and the understanding of the topics covered in this thesis may help to close the gap between theory and practice.

Therefore, the objective of this thesis is to introduce a framework for the systematic design and use of advanced control structures<sup>3</sup>. The aim is also to show that in many cases we can achieve optimal operation when we systematically design these control structures and to give some guidelines for this.

---

<sup>2</sup>The regulatory and supervisory control layers are described in Section 2.1.

<sup>3</sup>The aim is not to give the impression that PID-based control structures are always superior to model-based schemes such as MPC, but to give a systematic framework on how to design advanced control structures.

## 1.2 Classical advanced control structures

The purpose of this section is to give an overview of classical advanced control structures. Some of these control structures and blocks, such as selectors, split range control and valve position control, are analyzed in this thesis. For more examples of applications and detailed descriptions, the reader is referred to well-known process control books such as Stephanopoulos (1984); Balchen and Mummé (1988); Marlin (2000); Bequette (2002); Seborg et al. (2003); Skogestad and Postlethwaite (2005); Åström and Hägglund (2006).

Note that in this thesis the terms output ( $y$ ) and controlled variable (CV) are used as synonyms. Similarly, the terms input ( $u$ ) and manipulated variable (MV) are also used as synonyms<sup>4</sup>. The disturbances ( $d$ ) cannot be manipulated, controllers are denoted as  $C$ , single-input single-output (SISO) processes are denoted as  $g$  and multiple-input multiple-output (MIMO) as well as multiple-input and single output (MISO) processes are denoted as  $G$ .

### 1.2.1 Cascade control

Cascade control is shown in Fig. 1.2. This control structure can be used when there is one available input ( $u$ ), one primary output ( $y_1$ ) and one extra measurement ( $y_2$ ) that can be used to improve control of  $y_1$ . In cascade control the primary (master) controller ( $C_1$ ) in an outer loop, that controls  $y_1$ , gives the setpoint ( $y_2^{sp}$ ) for the secondary (slave) controller ( $C_2$ ) that controls  $y_2$  in an inner loop. The inner loop rejects disturbances ( $d_2$ ) on a faster time-scale, improving the dynamic response.

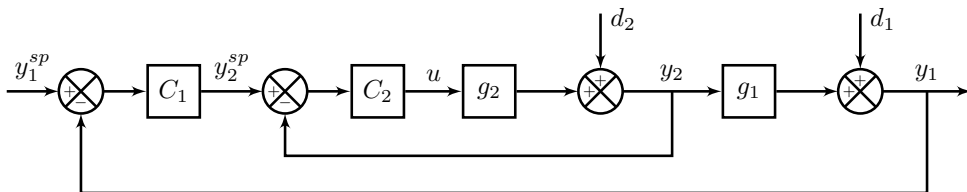


Figure 1.2: Cascade control structure.

The typical example of cascade control is a distillation column where we have a composition controller as a master  $C_1$  to a temperature controller ( $C_2$ )<sup>5</sup>. There are other ways of using extra measurements; for example, using state feedback rather than output feedback (Grimholt and Skogestad, 2016).

<sup>4</sup>This clarification may be required because, for example, in optimization the input vector usually includes the disturbances, which cannot be manipulated.

<sup>5</sup>We are interested in controlling the composition but outlet composition measurement have a long delay compared to a temperature measurement at a tray (location) close to the relevant outlet, which gives a good indication of composition.

### 1.2.2 Feedforward control

Feedforward control is shown in Fig. 1.3. This control structure can be implemented when there is a measured disturbance ( $d$ ) and its effect on the output ( $g_d$ ) is well-known. In feedback control, we adjust the input ( $u$ ) based on the measurement of ( $y$ ). In a feedforward scheme, we use the measured disturbance ( $d_m$ ) to adjust the input ( $u$ ) and keep the output ( $y$ ) constant when there are disturbances. The feedforward controller ( $C_{ff}$ ) is designed based on how  $u$  and  $d$  affect  $y$ , correspondingly,  $g$  and  $g_d$  in Fig. 1.3.

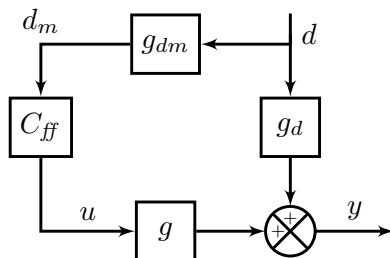


Figure 1.3: Feedforward control structure.

Feedforward is particularly useful if feedback control is not effective because the measurement of  $y$  is not available or the process delay (in  $g$ ) is larger than the disturbance measurement delay (in  $g_{dm}$ ) (Balchen and Mummé, 1988). In industry, the disturbance is sometimes called the “wild variable”. Feedforward control is usually used in combination with feedback in an additive manner. An alternative is a cascade implementation as in ratio control.

### Ratio control

This is the most common feedforward structure. The most common ratio control is with

$$\begin{aligned} u &= q_1 \\ d &= q_2 \\ y &= \frac{q_1}{q_2} \end{aligned}$$

where  $q_i$  are flowrates. In this case, ratio control can be implemented with

$$u = r d$$

where  $r$  is the setpoint for  $y$ . Fig. 1.4 shows this control structure, which can be seen as a simplification of Fig. 1.3, with  $C_{ff} = r$  and  $g = g_d = g_{dm} = 1$ .



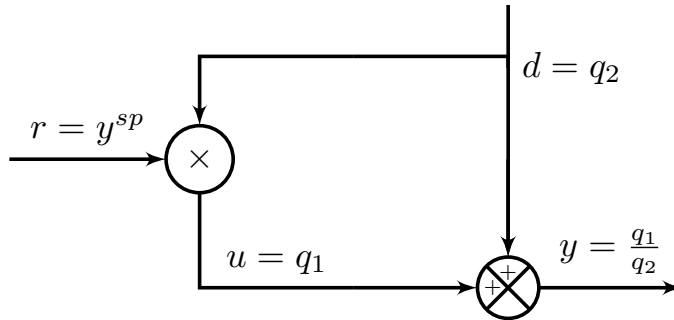


Figure 1.4: Simple ratio control structure.

A more advanced ratio control scheme that handles industrially relevant situations is presented by Hägglund (2017).

Ratio control is typically used in blending/mixing, for example, for the feed streams to a reactor. The idea is that a constant ratio will result in a constant composition (or temperature) of the mixed streams. The setpoint ( $r$ ) is often adjusted by feedback, in a cascade manner. Note that ratio control also has a linearizing effect and the process is decoupled by controlling the ratio of the flows instead of the flows themselves.

## Decoupling

Decoupling is used to reduce interactions and prepare multiple-input multiple-output (MIMO) systems for single-loop control. Decouplers are designed as feedforward control. For example, in a  $2 \times 2$  system,  $u_2$  is considered as a disturbance on  $y_1$  and  $u_1$  is considered as a disturbance on  $y_2$ .

The most common scheme is one-way decoupling (only decoupling for one of the interactions). For two-way decoupling the "inverse" scheme of Shinskey (1977), shown in Fig. 1.5, is recommended because it keeps the gain from  $u_1$  to  $y_1$  ( $g_{11}$ ) unchanged when we add the decoupling element and, correspondingly, the gain from  $u_2$  to  $y_2$  ( $g_{22}$ ) is also unchanged; thus, the decoupled process is

$$G_{decoupled} = \begin{bmatrix} g_{11} & 0 \\ 0 & g_{22} \end{bmatrix}$$

Another advantage of inverted decoupling is that it also accounts for input saturation because the actual process input is being measured and used for decoupling.

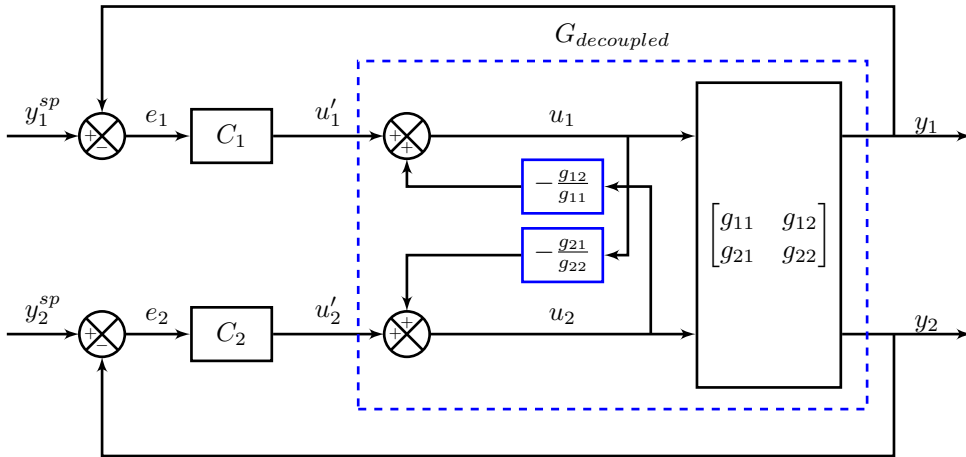


Figure 1.5: Two-way decoupling using the "inverted" implementation.

### 1.2.3 Selectors

Selectors can be seen as *building blocks* for advanced control structures. The output of a *max-selector*<sup>6</sup> is the signal that has the greatest value, while the output of a *min-selector*<sup>7</sup> is the signal with the least value. Similarly, the output of the *mid-selector* is the signal with the value in the middle (median). Note that the *mid-selector* can be obtained with a combination of *min* and *max* selectors.

#### *min* and *max* selectors for (optimal) steady-state operation

In this case there is one available input ( $u$ ) for more than one output ( $y_i$ ). As shown in Fig. 1.6, each output ( $y_i$ ) has a separate controller ( $C_i$ ) and the selector chooses which output to use ( $u$ ) based on the output  $u_i$  from the controllers<sup>8</sup>, either

$$u = \max(u_i)$$

or

$$u = \min(u_i)$$

This is feasible when acceptable control can be achieved for all outputs ( $y_i$ ) at any given time with the single input, either because it is allowed to give up controlling one output ( $y_i$ ) or because  $y_i$  only is constrained by a limit ( $y_i^{lim}$ ). This application is described in Section 2.2.5 of this thesis.

<sup>6</sup>Also called *high-select* and sometimes denoted with a "greater than" symbol (>).

<sup>7</sup>Also called *min-select* and sometimes denoted with a "less than" symbol (<).

<sup>8</sup>If the controlled variables have similar dynamics and the same units (or are normalized), it is possible to use one controller, with the selector block at the controller input. However, this scheme is more limited, and we recommend to implement selectors as in Fig. 1.6.

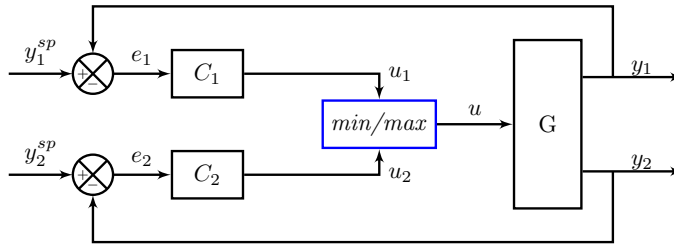


Figure 1.6: Use of a selector for the case with two CVs ( $y_1$  and  $y_2$ ) and one MV ( $u$ ).

Krishnamoorthy (2019) identified cases in which *CV to CV switching* using this scheme is not feasible. Krishnamoorthy and Skogestad (2019) give the example when we have one valve ( $u$ ) and we want to control maximum flow ( $y_1 = F^{max}$ ) and maximum pressure upstream ( $y_2 = P^{max}$ ). To maximize flow, the flow controller ( $C_1$  for  $y_1$ ) would open the valve ( $u = u^{max}$ ). To maximize pressure, the pressure controller (for  $C_2$  for  $y_2$ ) would close the valve ( $u = u^{min}$ ). With a *min* selector, the pressure controller ( $C_2$ ) would always be chosen and with a *max* selector, the flow controller ( $C_1$ ) would always be chosen.

### **mid-selector for steady-state range control**

This alternative is shown in Fig. 1.7. In this case we have one input ( $u$ ) and one output ( $y$ ) and  $y$  has both a lower bound ( $y^{min}$ ) and an upper bound ( $y^{max}$ ). In addition, we have a desired setpoint  $u^{sp}$  for the input. In this case we have two controllers on the same output  $y$ , but with different setpoints ( $y^{min}$  and  $y^{max}$ ). Let  $u^{min'}$  and  $u^{max'}$  be the values calculated by the two controllers. The *mid-selector* selects  $u = \text{mid}(u^{min'}, u^{sp}, u^{max'})$ . This will drive the output away from  $y^{max}$  or  $y^{min}$  when the disturbances are such that the controller outputs  $u^{min'}$  or  $u^{max'}$  reach the setpoint  $u^{sp}$ .

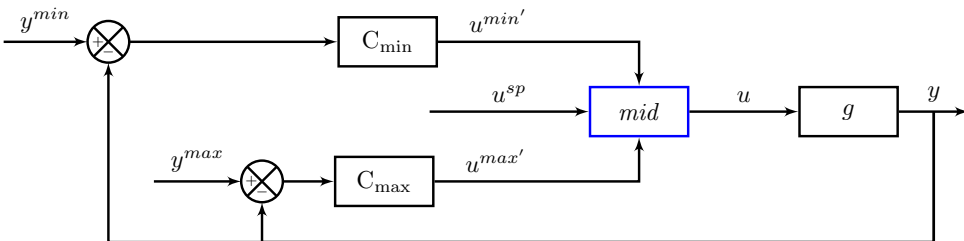


Figure 1.7: *mid-selector* with three inputs for steady-state range control.

Note that the setpoint  $u^{sp}$  can also be given as a feedforward from the disturbance (e.g. inflow for level control with a large filter time constant) or come from a controller

for another output. An example of this application of the *mid*-selector, with  $u^{sp}$  coming from another controller is described in Zotică et al. (2019).

### Override *min* and *max* selector for dynamic operation

In this case, the selector is used for dynamic purposes, that is, to bypass the normal control system when it is reacting so slowly such that some variable has reached an upper or lower limit. An override is not needed for steady-state operation. The override controller often controls a variable which is already controlled with another controller ( $C_m$  with  $y = y^{sp}$ ), but another faster controller is added ( $C_{lim}$  with  $y = y^{lim}$ ) for override purposes. At steady-state the "main" controller  $C_m$  will control  $y$ . This structure is shown in Fig. 1.8.

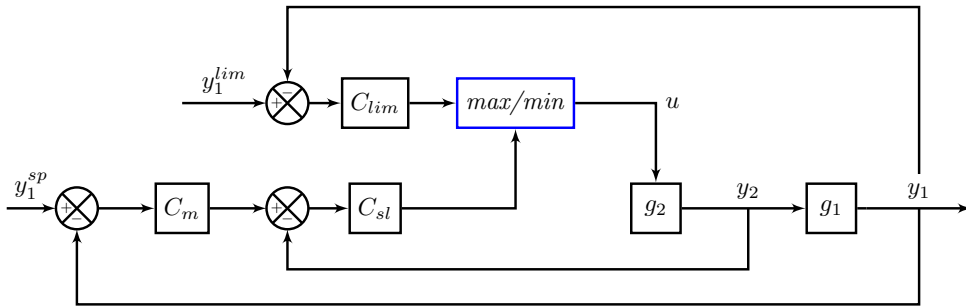


Figure 1.8: Cascade controller with override *min* and *max* selector for dynamic operation.

An example could be a distillation column where we have composition controller  $C_m$  as a master to a temperature controller ( $C_{sl}$ ), in cascade. Here, we add another composition controller  $C_{lim}$  that bypasses the temperature controller ( $C_{sl}$ ) if the composition gets out of bound and reaches  $y^{lim}$ .

### Override *mid*-selector with three controllers

Using a *mid* selector it is also possible to have a desired intermediate setpoint  $y^{sp}$ , that is, to let  $u^{sp}$  be the output of a third controller for the same output. This structure is shown in Fig. 1.9 and analyzed in Chapter 8 (Reyes-Lúa et al., 2018a), where we use P-controllers for  $C_1$  and  $C_3$  and a PI-controller for  $C_2$ .

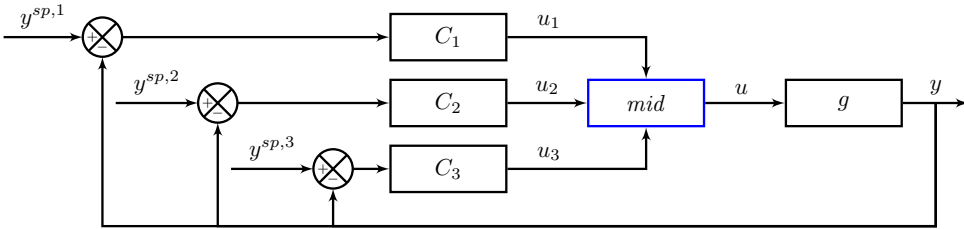


Figure 1.9: Override mid-selector with three controllers.

### 1.2.4 Input (valve) position control (VPC)

Valve position control has also been called input resetting (Skogestad and Postlethwaite, 2005) or mid-ranging control (Allison and Ogawa, 2003).

There are three cases in which it can be used:

1. **Dynamic case with one output and two inputs.** In this case, both inputs ( $u_1$  and  $u_2$ ) are always used, and both have a direct effect on the output  $y$ . Here, VPC is used for improving the dynamic performance for controlling the output ( $y$ ) if the dynamic response for the primary input ( $u_1$ ) is too slow. We could have used only input  $u_1$  to control  $y$ , but  $u_2$  is used dynamically to improve the dynamic performance. Thus, in this case,  $u_2$  is used to control  $y$ , whereas  $u_1$  controls  $u_2$  back to its desired steady-state value. This case is illustrated in Fig. 1.10.

A simple example is if we have a "big, slow" valve ( $u_1$ ) and a "small, fast" valve ( $u_2$ ) to control a desired total flowrate ( $y$ ). We use the "small, fast" valve  $u_2$  for dynamic performance, and we use the "slower, big" valve,  $u_1$ , to maintain  $u_2$  within a convenient range, close to  $u_2^{sp}$ . Note that in this example the flowrate corresponding  $u_1$  is higher than the one corresponding to  $u_2$ .

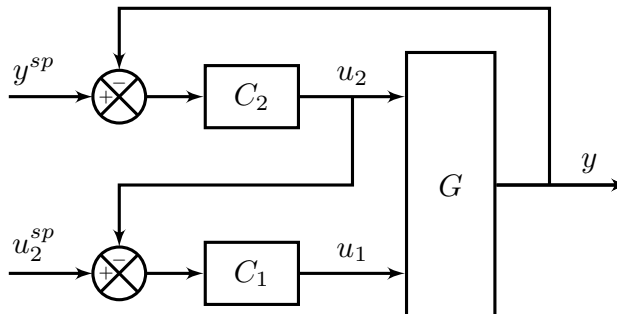


Figure 1.10: Input (valve) position control to improve dynamic performance.

This is an alternative to parallel (multi-input single-output) control where we use two inputs ( $u_1, u_2$ ) to improve the control of one output ( $y$ ).

2. **Dynamic case with one output and one input**, but with two controllers ( $C_y, C_u$ ). This is used for floating control of  $y$ . Floating control is used when it is required that  $y$  (e.g. pressure or temperature) is controlled at all times to stabilize the process or provide (local) disturbance rejection and avoid drift on a fast time scale, but the setpoint of  $y$ , in itself is not important.

We use the input ( $u$ ) to control  $y$  (with controller  $C_y$ ) while using an outer master controller (a VPC,  $C_u$ ) to bring  $u$  to a setpoint ( $u^{sp}$ ) and keep it away from its constraint. This is done by manipulating the setpoint  $y^{sp}$  for controller  $C_y$ . Here,  $y$  should be away from both boundaries<sup>9</sup>. This case is illustrated in Fig. 1.11.

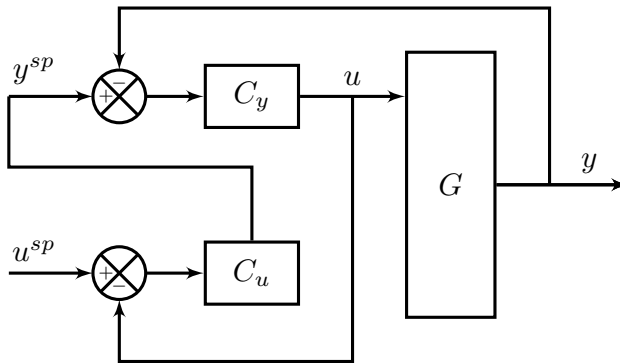


Figure 1.11: Input (valve) position control for floating control (of  $y$ ).

3. **Steady-state case with one output and two inputs**. In this case, VPC is used for extending the steady-state range for controlling  $y$  when the primary input  $u_1$  saturates. Here,  $u_2$  is only used to avoid that  $u_1$  saturates, so most of the time  $u_2$  is not used, which is different from the dynamic case where  $u_2$  is actually the input that controls  $y$ . In summary, in this case  $u_1$  is used to control  $y$ , whereas  $u_2$  controls  $u_1$  only when  $u_1$  reaches saturation or a limit. This is illustrated in Fig. 1.12. This use is also explained below in Section 1.2.5, as well as in Chapters 2 and 4.

<sup>9</sup>If the outer loop that adjusts  $u^{sp}$  is controlling a more important output variable ( $y_1$ ), then this is a special case of cascade control with a master and a slave loop, where the slave loop is the inner stabilizing controller.

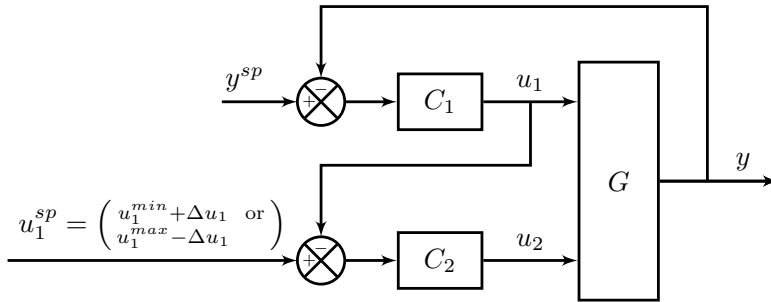


Figure 1.12: Input (valve) position control for range extending (alternative 3 in Section 1.2.5).

### 1.2.5 Input sequencing

Input sequencing is used when we use more than one input ( $u_i$ ) for steady-state control of one output ( $y$ ), in a predefined order. This topic is analyzed in several chapters of this thesis. There are three alternatives:

1. Split range control (Fig. 1.13)
2. Controllers on the same output with different setpoints (Fig.1.14)
3. Valve position control (VPC) (Fig. 1.12)

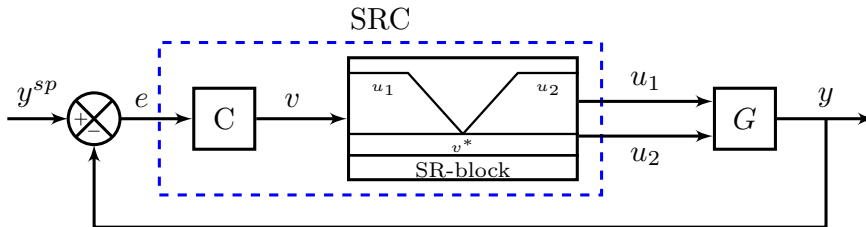


Figure 1.13: Alternative 1 for range extending: (standard) split range control.

In alternatives 1 and 2 we always use only one input at the time, whereas in alternative 3 (VPC), we use two inputs when the primary input is close to saturation. If we want to maintain  $y = y^{sp}$  and use the whole range, alternatives 2 and 3 are not steady-state optimal in the sense that there will be back-off either from the setpoint for  $y$  ( $\Delta y^{sp}$  in Fig. 1.14) or from the input constraint value ( $\Delta u_1$  in Fig. 1.12). This is discussed in Chapter 4. Standard split range control is analyzed in Chapter 5. In Chapter 7 we analyze in which cases having different controllers with different setpoints is steady-state optimal. The new generalized split range control structure introduced in Chapter 6 is a combination of alternatives 1 and 2.

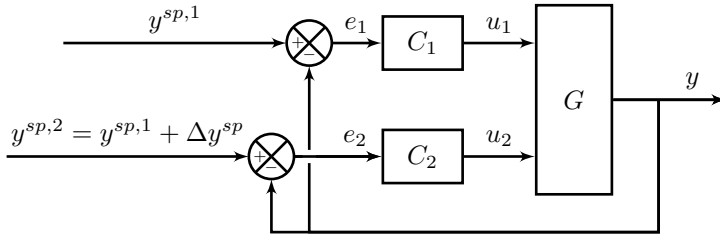


Figure 1.14: Alternative 2 for range extending: different controllers with different setpoints.

**1.2.6 Anti-windup**

This is used to (correctly) initialize the states of the controller when a controller is reactivated, and it can be done in many ways. In some cases we want to avoid that the input "jumps" (bumpless transfer) and then standard input tracking may be used. However, there may be cases where we want the input to "jump" (see Section 6.5.4).

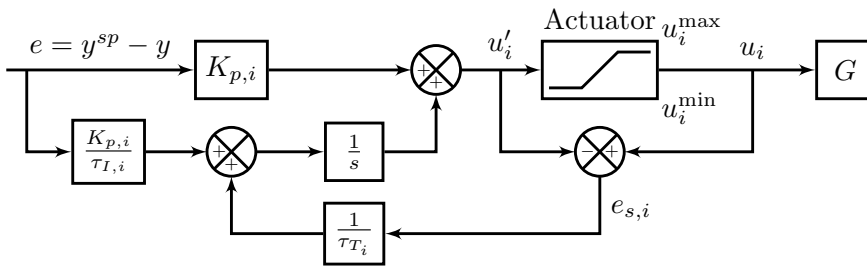


Figure 1.15: Antiwindup with bumpless transfer: input tracking with back-calculation for input  $u_i$ .

Fig. 1.15 illustrates input tracking with back-calculation as described by Åström and Hägglund (2006). Time constant  $\tau_T$  defines how fast the controller is driven to the actual actuator position.

**1.2.7 Other blocks**

Examples of other blocks that can be used to build control structures and are not described here are: more general logic blocks (e.g. *if... then*); calculation block (usually a static non-linear element used for decoupling, disturbance rejection or as a soft sensor); cross-limiting control, to prevent one or more variables from violating boundaries, and force on/off elements to cycle faster to improve dynamic performance.



### 1.3 Main contributions of this thesis

PID-based control structures are commonly used in process industries. However, their design mostly relies on the experience of the control and process engineers, and not always considers optimal operation. This thesis introduces a framework for the design of the supervisory control layer using PID-based control structures, considering steady-state optimal operation. Additionally, design and tuning guidelines for control structures used in industry are proposed and new control structures are introduced. Most of the work is focused on (but not limited to) control structures that extend the operating range by using more than one manipulated variable.

The main contributions of this work are:

1. Proposing a systematic design procedure for the supervisory control layer using PID-based advanced control structures (Chapters 2, 3 and 4).
  - Suggesting the use of *priority lists of constraints* for the design of PID-based control structures (Reyes-Lúa et al., 2018d).
  - Identifying the relevant active constraint switches that can be handled with PID-based control structures in the supervisory control layer.
  - Identifying which control structures can be used in each case.
  - Identifying limitations and advantages of the alternative control structures for extending the range of controlled variables.
2. Proposing a systematic procedure to design standard split range controllers, which is probably the advanced control structure most used for extending the operating range of controlled variables (Chapter 5).
3. Proposing a generalized split range control structure in which we can use independent controllers for each manipulated variable and avoid interactions by having only one active manipulated variable at a time (Chapter 6).
4. Identifying that in linear processes where the optimal control problem can be formulated as a combination of linear term for input usage and a quadratic term for setpoint deviation, it is optimal to extend the range of the controlled variable by using different controllers with different setpoints, and proposing a procedure to find these optimal setpoints (Chapter 7).
5. Proposing a control structure and setpoints to control the level as well as satisfying level boundaries in surge tanks (Chapter 8).

## 1.4 Organization of the thesis

- **Chapter 1** introduces the motivation and scope of the thesis and describes commonly used PID-based advanced control structures.
- In **Chapter 2** (Reyes-Lúa and Skogestad, 2019c), we propose a systematic procedure to design the supervisory control layer using single-loop classical advanced control structures such that the process achieves steady-state optimal operation when the active constraints change. We identify the types of active constraint changes and which control structures can be used in each case. The proposed methodology is illustrated with two case studies, one mixing process and one distillation column. This chapter gives the framework in which the results of the following chapters can be applied.
- In **Chapter 3** (Reyes-Lúa et al., 2019a) we design the control structure of a refrigeration cycle with heat recovery using the methodology proposed in Chapter 2.
- In **Chapter 4** we analyze the structures identified in Chapter 2 for MV to CV switching and compare them in a case study in which we control the temperature and the flow in a cooler (Reyes-Lúa et al., 2018d). Then, we compare split range control (the structure with the best results) with model predictive control (MPC) (Reyes-Lúa et al., 2018b) and show that both alternatives are steady-state optimal.
- In **Chapter 5** (Reyes-Lúa et al., 2019b) we introduce a procedure to design standard split range control, considering the different dynamic effects of each MV on the controlled variable, as well as (steady-state) economics.
- In **Chapter 6** (Reyes-Lúa and Skogestad, 2019a) we introduce a generalized control structure for split range applications that overcomes the limitations of standard split range control by using multiple independent controllers with the same setpoint. We illustrate our proposed structure with a case study for temperature control.
- In **Chapter 7** (Reyes-Lúa and Skogestad, 2019b), we identify that having different setpoints can be optimal in some cases and propose a procedure to find the optimal setpoint deviations. We illustrate our procedure with a case study for room temperature control.
- In **Chapter 8** (Reyes-Lúa et al., 2018a) we propose a simple control structure with a *mid*-selector, two P-controllers and a PI-controller in a "split-range scheme" to reduce flow variations in a surge tank, satisfying level constraints and compare the simulation results with MPC.

- In **Chapter 9** the thesis is concluded with some general final remarks and some suggestions for future work.
- **Appendices A-E** give supporting information such as optimization results, models and tuning parameters for the case studies in the thesis.
- In **Appendix F** (Reyes-Lúa and Skogestad, 2020) we apply the generalized split range control structure introduced in Chapter 6 in a case study in which we must switch the manipulated variable used to control an important controlled variable when the primary manipulated variable saturates.
- In **Appendix G** (Reyes-Lúa et al., 2016) we identify the active constraint regions of a different distillation column than the one studied in Chapter 2 including the thermodynamic equations as part of the optimization problem.

## 1.5 Publications

The chapters of this thesis are based on the following publications:

### 1.5.1 Journal papers

- Chapter 2 corresponds to:  
Reyes-Lúa, A. and Skogestad, S. (2019c). Systematic design of active constraint switching using classical advanced control structures. *Industrial & Engineering Chemistry Research*
- Chapter 6 corresponds to:  
Reyes-Lúa, A. and Skogestad, S. (2019a). Multi-input single-output control for extending the operating range: Generalized split range control using the baton strategy. *Journal of Process Control (Under review)*
- Chapter 7 corresponds to:  
Reyes-Lúa, A. and Skogestad, S. (2019b). Multiple-Input Single-Output Control for Extending the Steady-State Operating Range—Use of Controllers with Different Setpoints. *Processes*, 7(12):941

### 1.5.2 Peer-reviewed conference papers

- Chapter 3 is based on:
 

Reyes-Lúa, A., Andreasen, G., Larsen, L. F. S., Stoustrup, J., and Skogestad, S. (2019a). Optimal operation of a  $CO_2$ - refrigeration system with heat recovery. In *Proceedings of the 29th European Symposium on Computer Aided Process Engineering (ESCAPE)*, Eindhoven, Netherlands. Computer-aided chemical engineering
- Chapter 4 is based on:
  - Reyes-Lúa, A., Zotică, C., and Skogestad, S. (2018d). Optimal Operation with Changing Active Constraint Regions using Classical Advanced Control. In *10th IFAC Symposium on Advanced Control of Chemical Processes (ADCHEM)*, Shenyang, China. IFAC-Papers OnLine
  - Reyes-Lúa, A., Zotică, C., Das, T., Krishnamoorthy, D., and Skogestad, S. (2018b). Changing between Active Constraint Regions for Optimal Operation: Classical Advanced Control versus Model Predictive Control. In *Proceedings of the 28th European Symposium on Computer Aided Process Engineering (ESCAPE)*, Graz, Austria. Computer-aided chemical engineering
- Chapter 5 is published as:
 

Reyes-Lúa, A., Zotică, C., Forsman, K., and Skogestad, S. (2019b). Systematic Design of Split Range Controllers. In *12th IFAC Symposium on Dynamics and Control of Process Systems, including Biosystems (DYCOPS)*, Florianópolis, Brazil. IFAC-Papers OnLine
- Chapter 8 is published as:
 

Reyes-Lúa, A., Backi, C. J., and Skogestad, S. (2018a). Improved PI control for a surge tank satisfying level constraints. In *3rd IFAC Conference on Advances in Proportional-Integral-Derivative Control (PID18)*, volume 51, pages 835–840, Ghent, Belgium. IFAC-Papers OnLine

The following peer-reviewed conference papers were published or submitted during the Ph.D. period and are included as appendices:

- Included as Appendix F:
 

Reyes-Lúa, A. and Skogestad, S. (2020). Active constraint switching with the generalized split range control structure using the baton strategy. In *21st IFAC World Congress (Submitted)*, Berlin, Germany. IFAC Papers Online

- Included as Appendix G:

Reyes-Lúa, A., Solvik, M., and Skogestad, S. (2016). Inclusion of thermodynamic equations for efficient steady-state process optimization. In *Proceedings of the 26th European Symposium on Computer Aided Process Engineering (ESCAPE)*, Portorož, Slovenia. Computer-aided chemical engineering

The following conference participations were also co-authored during the Ph.D. period, but are not part of this thesis.

- Reyes-Lúa, A., Zotică, C., and Skogestad, S. (2018c). Optimal operation using classical advanced control structures. In *21st Nordic Process Control Workshop (NPCW)*, Turku (Åbo), Finland
- Reyes-Lúa, A. and Skogestad, S. (2016). Optimal operation of vapor-compression cycles in off-design conditions. In Kitanovski, A. and Poredoš, A., editors, *29th International Conference on Efficiency, Cost, Optimisation, Simulation and Environmental Impact of Energy Systems (ECOS)*, Portorož, Slovenia
- Skjervold, V., Gullberg, R., Langørgen, Ø., Berstad, D., Schandera, C., Reyes-Lúa, A., and Jordal, K. (2016). Evaluation of a natural gas-fired CLC boiler for industrial steam generation. In *13th International Conference on Greenhouse Gas Control Technologies*, Lausanne, Switzerland
- Jordal, K., Reyes-Lúa, A., and Langørgen, Ø. (2015). The potential benefit of using CLC in industrial boilers. In *The 8th Trondheim CCS Conference*, Trondheim, Norway



# Systematic design of active constraint switching using classical advanced control structures

This chapter is published as:

Reyes-Lúa, A. and Skogestad, S. (2019c). Systematic design of active constraint switching using classical advanced control structures. *Industrial & Engineering Chemistry Research*

The most important task of the supervisory control layer is to maintain *optimal operation*. To achieve this, we need to change control objectives when constraints become active (or inactive) due to disturbances. In most process plants, the supervisory layer uses classical PID-based advanced control structures, but there is no systematic way of designing such structures.

Here, we propose a systematic procedure to design the supervisory control layer using single-loop classical advanced control structures such that the process achieves steady-state optimal operation when the active constraints change. The active constraints can be on the manipulated variable (MV, input) or on the controlled variable (CV, output). In this chapter, we consider all three possible cases:

- CV-CV switching, which involves selectors
- CV-MV switching, which does not need any special structure if we pair according to the input saturation pairing rule
- MV-MV switching, which uses split range control or a similar structure.

We illustrate our methodology with two case studies, a mixing process and a distillation column.

## 2.1 Introduction

The control hierarchy typically used in process plants decomposes the overall control problem on a time scale basis, as shown in Fig. 2.1. The upper layers are related to long-term economic optimization, whereas the two lower layers are control layers, with the objective to keep the controlled variables (CVs) at their desired setpoints.

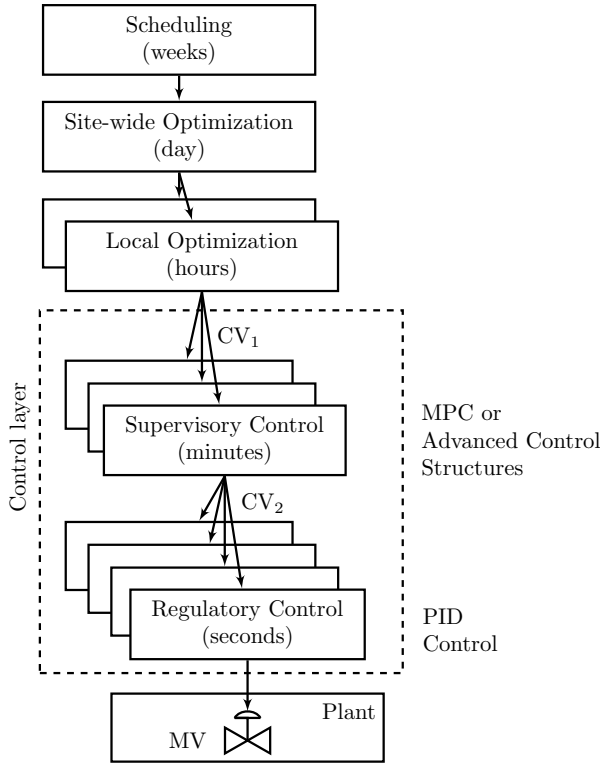


Figure 2.1: Typical control hierarchy in a process plant.

The control layer is sub-divided into a supervisory control layer and a regulatory or stabilizing control layer. The main objective of the regulatory layer is to stabilize the process and avoid drifting away from the desired steady-state, and to reject disturbances on a fast time scale (Lu, 2003; Skogestad, 2004). The supervisory control layer should follow the setpoints for the controlled variables computed by the optimization layer ( $CV_1$ ). Importantly, this involves switching between active constraint changes in  $CV_1$ . It also calculates the setpoints for the regulatory layer ( $CV_2$ ), and avoids steady-state saturation of the manipulated variables (MVs) used by the regulatory layer<sup>1</sup>.

<sup>1</sup>Note that in this work, the terms output ( $y$ ) and controlled variable (CV) are used as synonyms.



Skogestad (2004) proposed a systematic procedure for control structure design for complete process plants. The procedure is separated in two main parts: top-down analysis and bottom-up design. The top-down analysis focuses on identifying the steady-state optimal operation, usually based on economics. The bottom-up part focuses on the design of the control layer structure. The procedure is as follows:

– *Top-down analysis:*

- S1: Define a cost ( $J$ ) to be minimized (economics), and identify constraints that must be satisfied during operation.
- S2: Identify the degrees of freedom ( $u$ , MVs) and determine the optimal operation conditions (including active constraints) for expected disturbances (usually at steady-state).
- S3: Identify candidate measurements ( $y$ ) and, from these, select controlled variables ( $CV_1$ ). Active constraints should always be controlled for optimal operation. For the remaining unconstrained degrees of freedom we should control "self-optimizing" variables, which, when kept constant, indirectly minimize the cost (Skogestad, 2000).
- S4: Select the location of the throughput manipulator (TPM) (Price and Georgakis, 1993), which is where the production rate is set. This is a dynamic decision. For maximizing production (reduce back-off), the TPM should be located at the bottleneck (Aske and Skogestad, 2009).

– *Bottom-up design of the control structure:*

- S5: Select the structure of the regulatory PID control layer. Select "stabilizing" control variables ( $CV_2$ ) and, since single-loop control is preferred in this layer, choose pairings for  $CV_2$  with manipulated variables (MVs).
- S6: Select the structure of the supervisory control layer. It can be model-based (using MPC), but in this chapter we consider the use of classical advanced control elements.
- S7: Select the structure for the online optimization layer (RTO), if required. The RTO layer may be avoided if one can switch between active constraints in the supervisory layer, and can identify good self-optimizing variables (Skogestad, 2000) for the remaining unconstrained degrees of freedom.

This procedure can be followed sequentially, but one decision directly influences the others, such that the procedure may be iterative (Skogestad, 2004; Skogestad

---

Similarly, the terms input ( $u$ ) and manipulated variable (MV) are also used as synonyms and refer to the physical input variables.

and Postlethwaite, 2005). In this work we focus on step S6, specifically on how to handle switching between active constraints. The decisions taken in the *top-down* part of the procedure, especially the identified active constraints, directly affect the design of the supervisory control layer, and we assume that the former decisions are already taken.

*Active constraints* are variables that should optimally be kept at their limiting value (step S3). These can be either *manipulated variable* (MV, input) constraints or *controlled variable* (CV, output) constraints. The maximum pressure in a unit is a CV constraint, while the maximum opening of a valve is a typical example of an MV constraint. We need to be a bit cautious about what we mean by MV constraint because the term MV generally denotes the degrees of freedom in any layer. For example, when referring to the supervisory layer, it may refer to the setpoint for the CV<sub>2</sub> in the regulatory layer. However, in the context of this work, *MV constraints* mean minimum or maximum values of the *physical* manipulated variable (e.g. valve opening or pump rotational speed).

If there are remaining unconstrained degrees of freedom in step S3, then one should identify associated self-optimizing variables to keep at constant setpoints (Skogestad, 2004). Controlling the self-optimizing variable to its optimal setpoint keeps the process at (near-)optimal operation (Skogestad, 2000; Skogestad and Postlethwaite, 2005). Self-optimizing variables can be a specific measurement, a combination of measurements ( $c = Hy$ ) (Alstad et al., 2009), or a measurement or estimate of the gradient of the cost ( $J_u = dJ/du$ ). Note that the self-optimizing variables generally will change when we enter a different active constraint region.

If there were no changes in the operating point and, in particular, no changes in the active constraints, optimal operation would always be achieved by using the same control structure and constant setpoints in the regulatory control layer. However, all plants are subject to disturbances which may cause changes in the optimal operation point and the active constraints. Typical disturbances include changes in feed rate, feed composition, product specifications, prices, and drift in process parameters such as efficiencies.

In terms of economics, the most important role of the supervisory control layer is to keep the operation in the right *active constraint region*, which is a region in the disturbance space defined by which constraints are active within it (Jacobsen and Skogestad, 2011). Stephanopoulos (1984) states that an optimizing control strategy in the supervisory layer must identify when the plant must be moved to a new operating point (changes active constraint region) and then make the appropriate setpoint changes to bring the plant to the new optimum operating point.

The supervisory control layer is sometimes designed with Model Predictive Control

(MPC). The main advantage of MPC in terms of economics is that it can handle many constraints and that it represents a unified systematic procedure to control multivariable processes (Mayne, 2014). The main drawback of MPC is that it requires a dynamic model of the process, which is not always available or is costly to generate and update. Furthermore, standard MPC may not handle changes in active constraints effectively, except by the indirect use of weights in the objective function (Allison and Isaksson, 1998; Forbes et al., 2015).

The supervisory control layer can alternatively be designed using *classical advanced control structures* with PID-controllers and simple blocks, and this is the most common control approach in industry. The main reason is that classical structures can be gradually implemented in the existing “basic” control system using little model information (Skogestad, 2015). Some classical advanced control elements (also called blocks or idioms (Bristol, 1980)) used in addition to PID controllers include (Shinsky, 1988; Seborg et al., 2003):

- cascade
- feedforward and ratio
- decoupling
- calculation block
- valve position (input resetting)
- selector (*max*, *min*)
- split range (input sequencing)

These structures have been used since the 1940s (Eckman, 1945; Fink, 1945). However, there has been limited academic work and most implementations are *ad-hoc*. The lack of a systematic procedure to design control structures was pointed out by Foss (1973) in his famous paper with title *Critique of chemical process control theory*. He writes that *"the central issue to be resolved by the new theories of chemical process control is the determination of control system structure"*. Following this, some research was initiated to design control structures in a systematic way (e.g. Vandebussche (1975), Govind and Powers (1978), Govind and Powers (1982), Bristol (1980), Stephanopoulos (1984)). Although some good ideas were introduced, this research has had limited impact. More recently Hägglund and Guzmán (2018) pointed out that little research and development has been presented to the use of the basic control structures, even in the regulatory layer. To the knowledge of the authors, there is no systematic procedure to design the supervisory control layer structure (step S6) using classical advanced control elements. In this work, we present such

a systematic procedure and show its applicability in two industrially relevant case studies.

## 2.2 Design procedure for constraint switching using classical advanced control structures

The proposed procedure to design constraint switching strategy for the supervisory layer (step S6) using *Advanced control structures* has five main steps:

Step A1: Define the control objectives (CVs), manipulated variables (MVs) and constraints. Distinguish between CV and MV constraints.

Step A2: Organize the constraints in a *priority list*. That is, identify which setpoints or constraints can be given up in order to guarantee feasible operation.

Step A3: Identify possible and relevant active constraint switches.

Step A4: Design the control structure for normal operation.

Step A5: Design the control structures to handle the identified active constraint switches.

We will now detail each step.

### 2.2.1 Step A1: Define the control objective, MVs and constraints

The control objectives in the supervisory layer are specified in terms of *controlled variables* (CVs) with setpoints. These follow from step S3 in the top-down analysis. These were called  $CV_1$  earlier, but for simplicity we will now just call them CV in the rest of the thesis. Note that the CVs from step S3 may also include MVs. The main objective of step S6 is to implement this in practice. The main problem is that the variables that we need to control may change during operation due to changes in active constraints.

A detailed analysis in step S3 results in a number of active constraints regions, each with a specific set of *controlled variables*. However, in practice, such a detailed analysis usually is too time consuming to perform. Instead we may, based on a partial analysis in step S3 and engineering judgment, list the expected controlled variables:

1. Outputs (CVs) with setpoints (denoted CV equality constraints in the following). For example, product specifications and operating pressures and temperatures.

2. Input variables with desired values or setpoints (denoted MV equality constraints in the following). For example, a desired value for rotational speed of a compressor.
3. Output (CV) constraints. These may become optimally active at certain steady-state operating points.
4. Input (MV) constraints. These may become optimally active at certain steady-state operating points.
5. Self-optimizing CVs. These are associated with unconstrained degrees of freedom and keeping them at constant setpoints should indirectly minimize the economic cost.
6. Desired throughput (production rate). Typically, but not always, a flowrate (MV or CV) with a given setpoint.

Sometimes the throughput is given and may enter as an MV equality constraint. However, in many cases with good market conditions, optimal operation (minimum cost,  $J$ ) is achieved by maximizing the throughput. In this case, one may set an unachievable high setpoint for the production rate, and optimal operation (maximum production) is achieved when one reaches the bottleneck, which is when there are no more constraints that can be given up (Aske and Skogestad, 2009).

The best self-optimizing CV will change when the active constraints change, but for simplicity we often try to use the same “self-optimizing” CV in several regions. This will imply that its setpoint may need to vary depending on the disturbance value; for example, the feed rate. To identify self-optimizing variables and their setpoints, we generally need a process model. Note that otherwise, the procedure proposed in this chapter does not need explicit model information.

### 2.2.2 Step A2: Organize the constraints in a list of priorities

At some operating conditions, it may not be feasible to satisfy all the constraints using the available MVs. In this case, one may use a priority list to decide which constraints can be given up to make operation feasible. This will also help us in making decisions regarding pairing of CVs and MVs.

Physical MV constraints, which of course cannot be violated, are placed at the highest priority. This means that they cannot be given up. Economic objectives such as desired throughput and self-optimizing setpoints are at the lower end of the priority list. By placing the most important constraints at the top, the *priority list* typically has the following structure:

- (P1) *Physical MV inequality constraints.* It is physically impossible to give them up. Typical examples are: maximum or minimum opening of valves, or maximum pump speed.
- (P2) *Critical CV inequality constraints.* These may possibly be given up for a short period. These are often safety constraints such as maximum temperature or maximum pressure.
- (P3) *Non-physical MV and less critical CV constraints* (both equality and inequality constraints). These may be given up; for example a desired pressure (CV equality constraint). By non-physical MV constraints, we mean a constraint that is not related to a fully open or closed valve (control element). For example, it could be the minimum liquid flow in a distillation column to ensure proper wetting of the packing, or maximum flow to avoid excessive wear.
- (P4) *Desired throughput.* These are MV or CV equality constraints, which must be given up when we reach a bottleneck. Typically, this happens when we reach a physical MV inequality constraint and there are no variables with lower priority that can be given up.
- (P5) *Self-optimizing variables.* These are economic CV equality constraints, which can be given up.

It is important to note that the ordering of items P2, P3 and P4 may vary depending on the specific case. Often, the desired throughput has a higher priority than a CV inequality constraint; for example, a desired setpoint for a byproduct concentration. Within the constraints in P3, there might be CV or MV equality constraints with a higher priority than others. It should also be noted that constraints in P3, P4 and P5 may include the same variables that are already used in P1 and P2, but with different bounds.

Usually, few physical MV constraints are active in the base case operating point. When disturbances occur and we operate away from this point, we may reach physical MV constraints and we have to give up controlling some other CV or MV constraint. The order in which constraints should be given up as we move away from nominal operation follows the reverse of the priority list. We first give up the constraints at the end of the list (with the lowest priority) and continue satisfying those with higher priority.

### 2.2.3 Step A3: Identify active constraint switches

Once all possible constraints have been identified and prioritized, we proceed to identify active constraint switches. This will occur when disturbances cause a CV or MV to reach a new inequality constraint and we have to give up controlling some

other variable or, inversely, that an inequality constraint is no longer active, and we can start controlling another variable. Therefore, the priority list from step A2 is very useful for identifying likely switches.

One may believe that we need to obtain all the active constraint regions as a function of all the disturbances. However, obtaining this information is usually very time consuming, even for quite simple processes and, fortunately, it is not necessary (Jacobsen and Skogestad, 2011). We only need to know which active constraint switches are relevant. We do not need the actual point (value of the disturbance) at which we change from one active constraint region to the other, as this will be indirectly identified online with the value of the MVs and CVs. It is insightful to know the maximum number of active constraint regions, which is given by (Jacobsen and Skogestad, 2011):

$$n_r^{\max} = 2^{n_c} \quad (2.1)$$

where  $n_c$  is the number of constraints. In practice, there are usually much fewer possible and even fewer relevant active constraints regions ( $n_r$ ), so

$$n_r < n_r^{\max} \quad (2.2)$$

The following criteria is useful to find possible active constraint regions, so that we can design the control structure considering only the regions of interest:

- Certain constraints are always active (reduces effective  $n_c$  in Eq. (2.1)).
- Certain constraint combinations are not possible. For example, maximum or minimum bounds on the same variable cannot be reached at the same time.
- Certain constraints (or regions) cannot be reached by the assumed disturbance set.
- At a given time, the number of active constraints is limited by the number of degrees of freedom (MVs).

#### 2.2.4 Step A4: Design control structure for base case operation

The next step is to design a control structure for the base case operating point, which is typically the nominal operating point. This is often a case with relatively few active constraints and in which most, if not all, constraints in the priority list can be satisfied. In this step, we should follow standard guidelines for designing control structures (Price and Georgakis, 1993; Price et al., 1994; Aske and Skogestad, 2009; Skogestad, 2015). For example, we should follow the *pair close* rule for a good dynamic response (minimize delays) (Minasidis et al., 2015).

When designing the base case control structure for optimal operation, we should note that a constrained MV does not need to be actively controlled. Thus, if it is optimal to maintain a valve fully open or fully closed, such as in a bypass, then we do not need to implement a controller to achieve this. We simply set it fully open or fully closed. In order to reduce the need of repairing of loops as we go away from the base case, we recommend to pair MVs with CVs according to the following rule (Minasidis et al., 2015):

**Input saturation pairing rule:** *A manipulated variable (MV) that is likely to saturate at steady state, should be paired with a controlled variable (CV) that can be given up.*

By "can be given up" we mean that it is near the bottom of the priority list. If we do not follow the input pairing rule, then we need to find another MV to take over controlling the CV. An alternative formulation of the rule is *pair an MV which is unlikely to saturate with an important CV.*

### 2.2.5 Step A5. Design control structures for active constraint switching

There is a fundamental difference between MV and CV constraints because we need an MV to control a CV, whereas an MV can simply be set at its constraint value. Considering this, the following constraint switches can occur:

Case 1: *CV (output) to CV (output) constraint switching*

Case 2: *MV (input) to MV (input) constraint switching*

Case 3: *MV (input) to CV (output) constraint switching*

#### Case 1: CV to CV constraint switching

This case typically happens when we have one input (MV) which switches between controlling two alternative CVs, meaning that only one CV is controlled at any given time.

To switch between the CVs, we can use two independent controllers and a *max/min* selector, so that the active CV constraint is always selected. Fig. 2.2 shows the block diagram with two CVs ( $y_1$  and  $y_2$ ) and one MV ( $u$ ). It is important to note that anti-windup must be implemented in both controllers ( $C_1$  and  $C_2$ ).

A possible misconception here is that all the CVs ( $y_1$  and  $y_2$  in Fig. 2.2) need to be of the same type (e.g. temperature) as in *auctioneering*, where we have one controller and the selector is on the input of the controller and we select to control



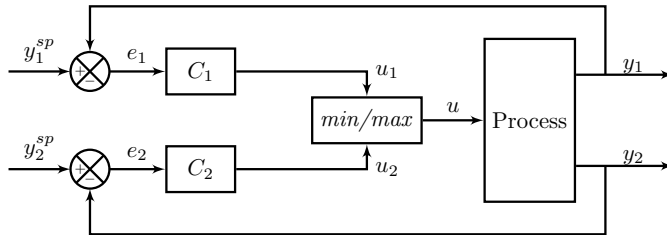


Figure 2.2: CV to CV switching using a selector for the case with two CVs ( $y_1$  and  $y_2$ ).

one of several similar outputs (Skogestad and Postlethwaite, 2005). However, in general the CVs may be of different type if the selector is on the output from the controller (Wade, 2004), as in Fig. 2.2.

As an example, Fig. 2.3 shows a flowsheet in which the coolant flow, actually its setpoint ( $u = \dot{m}_w^{sp}$ ), is the only available MV to control either the reactor temperature ( $y_1$ ) or concentration ( $y_2$ ), both of which can reach their corresponding maximum constraints. A selector on the controller output signals ( $u_1$  and  $u_2$ ), allows for the CV switching between temperature ( $y_1$ ) and composition ( $y_2$ ).

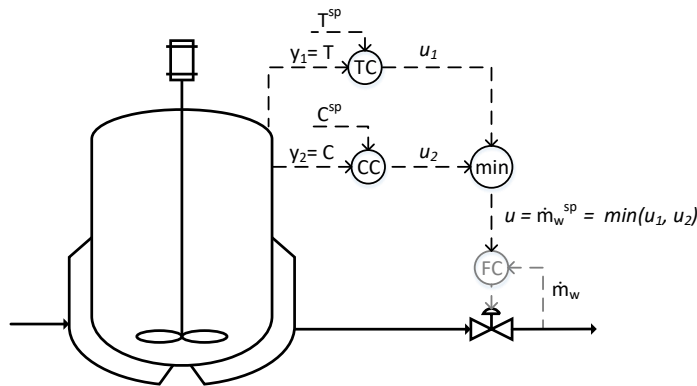


Figure 2.3: Typical example of CV to CV switching based on controller output signals. The regulatory layer is dimmed in gray.

Such schemes are sometimes called *override control* (Shinkskey and Shunta, 1995; Wade, 2004; Schuurmans, 2019). However, we prefer to call it *CV to CV switching* to avoid the connotation of "error" or "emergency" of the term *override*. On the contrary, the CV to CV switching is desirable and economically optimal. It is also worth mentioning that this is a logical switch. It is not single-input multiple-output (SIMO) control, which usually refers to the use of *one* controller to control two CVs in some

weighted or average manner (e.g. Freudenberg and Middleton (1999), Amezcuita-Brooks et al. (2017)). For a more detailed discussion of CV to CV switching for optimal operation, the reader is referred to Krishnamoorthy (2019).

### Case 2: MV to MV constraint switching

This case typically happens when the primary MV saturates, and an extra MV is added to cover the whole steady-state range and maintain control of the CV. Three alternative schemes can be used for input-to-input constraint switching:

1. Split range control (SRC).
2. More than one controller for the same CV, each with a different setpoint.
3. Input (valve) position control.

In the first two schemes, only one MV is actively controlling the CV, while the other MVs are fixed at a limiting (minimum or maximum) value.

*Split range control* is the most common scheme. It has been in use for more than 75 years (Eckman, 1945; Fink, 1945), and it is still commonly implemented in industry (Sun et al., 2015). Some other names that have been used for split range control are *dual control agent* (Eckman, 1945), *range extending control* (Bristol, 1980) and *valve sequencing* (Lipták, 1985). Fig. 2.4 shows the block diagram of a split range controller (SRC) with two MVs ( $u_1$  and  $u_2$ ) for one CV ( $y$ ). When the internal control signal ( $v$ ) is below the split value ( $v^*$ ),  $u_1$  is used to control  $y$ , while  $u_2$  is fixed at a limiting value; when  $v$  is above  $v^*$ ,  $u_2$  is used to control  $y$ , while  $u_1$  is fixed at a limiting value.

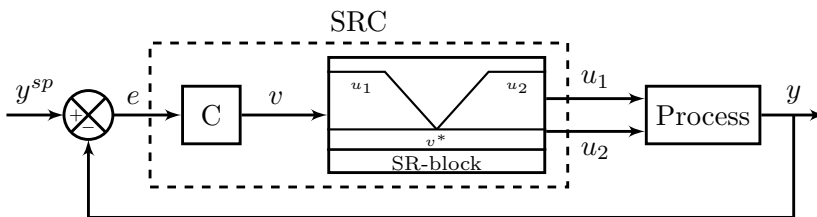


Figure 2.4: MV to MV constraint switching using split range control (SRC) for a case with two MVs ( $u_1$  and  $u_2$ ) and one CV ( $y$ ).

Split range controllers should be designed considering the different dynamic effects of each MV on the output, as well as steady-state economics. There is a single controller (C) in Fig. 2.4, but independent adjustments of the controller gains are

possible by making use of the location of  $v^*$ , or equivalently, the slopes in the split range block (SR-block) (Reyes-Lúa et al., 2019b). However, for *standard* split range control, other controller parameters like the integral time, have to be the same for both inputs (MVs).

The most common alternative to split range control is to use *one controller for each MV* with different setpoints, e.g.  $y^{sp}$  and  $y^{sp} + \Delta y^{sp}$ , as shown in Fig. 2.5.  $\Delta y^{sp}$  should be large enough such that only one controller is active at a given time, while the other inputs are at their limits (Smith, 2010).

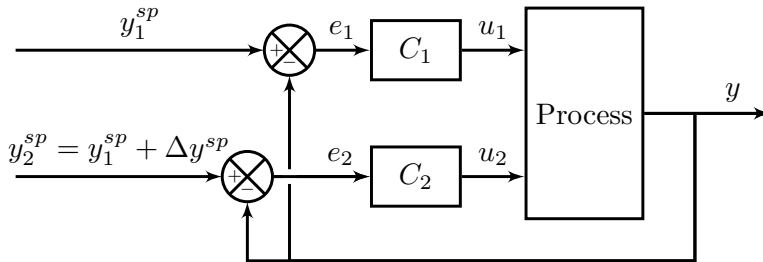


Figure 2.5: MV to MV constraint switching using two controllers with different setpoints.

The third option, shown in Fig. 2.6, is *input (valve) position control* (VPC) (Shinsky, 1978, 1981). It is commonly used to improve the dynamic performance by the use of extra dynamic inputs<sup>2</sup>, and then is sometimes referred to as *input resetting* (Skogestad and Postlethwaite, 2005; Sun et al., 2018) and *mid-ranging control* (Allison and Ogawa, 2003).

However, here we are considering it as an alternative to split range control to extend the steady-state range (Reyes-Lúa et al., 2018d), as shown in Fig. 2.6. In this case,  $u_1$  always controls  $y$ . We cannot let  $u_1$  fully saturate because otherwise control of  $y$  is lost. If the input ( $u_1$ ) approaches its limit  $u_1^{lim}$  (upper or lower), given by  $u_1^{sp}$  (for example,  $u_1^{sp} = u_1^{min} + \Delta u_1$ ), then input  $u_2$  indirectly takes over the control of  $y$  by keeping  $u_1$  close to this value ( $u_1^{sp}$ ).  $\Delta u_1$  is the "back-off", i.e.  $\Delta u_1 \neq 0$ . The advantage with this scheme is that the output ( $y$ ) is always controlled with the same "primary" input  $u_1$ . The disadvantages are that one cannot utilize the full steady-state range of this "primary" input ( $u_1$ ), and that tuning of the outer controller ( $C_2$  in Fig. 2.6) may be challenging (Sun et al., 2018).

<sup>2</sup>When used for dynamic reasons, while  $u_1$  takes care of the fast control,  $u_2$  takes care of the long-term control, and  $u_1$  (usually a valve) is slowly *reset* to a desired *mid-range position* ( $u_1^{sp}$ ) using  $u_2$  (Skogestad and Postlethwaite, 2005; Haugwitz et al., 2005). This way, the MV controlling the CV ( $u_1$ ) does not saturate.

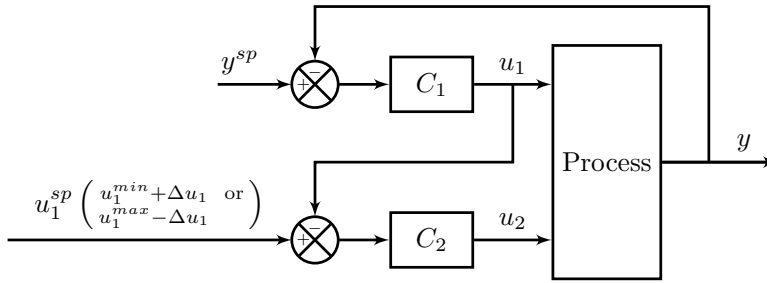


Figure 2.6: MV to MV constraint switching using input (valve) positioning control.

### Case 3: MV to CV constraint switching

This happens when we have saturation of the MV ( $u_1$ ) that we are using to control a CV ( $y_1$ ). In this case there are two possibilities:

1. The *input saturation pairing rule* was followed. This means that the CV ( $y_1$ ) can be given up: This case is shown in Fig. 2.7. Here, the switch is already "built-in". That is, it is not necessary to do anything, except that we must implement anti-windup in  $C_1$  for a good transition performance when control of  $y_1$  is reactivated; that is, when  $u_1$  is no longer saturated.

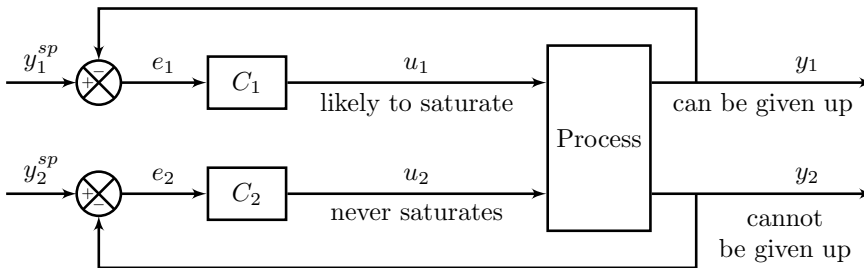


Figure 2.7: MV to CV switching for the case when the input saturation rule is followed, so control of  $y_1$  can be given up.

2. The *input saturation pairing rule* was *not* followed. This means that we cannot give up controlling the CV ( $y_1$ ). Thus, when the MV ( $u_1$ ) reaches its limit (saturates) we need to find another MV ( $u_2$ ) to take over the task. This will generally invoke a reparing, because the new MV ( $u_2$ ) is already controlling a low-priority CV ( $y_2$ ). To do this, we may implement an *MV to MV* switching strategy, such as split range control, in combination with a *min/max* selector (Reyes-Lúa et al., 2018d), as shown in Fig. 2.8.

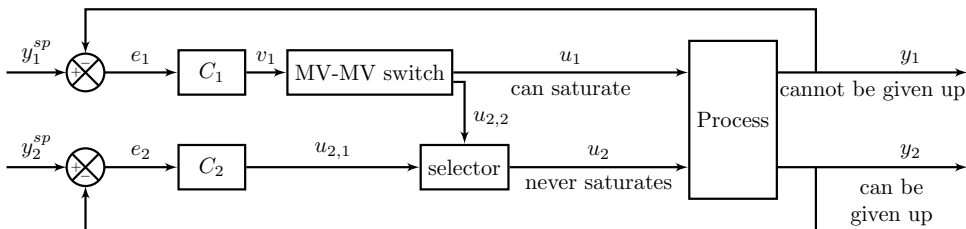


Figure 2.8: MV to CV switching for the case when the input saturation rule is *not* followed; so control of  $y_1$  cannot be given up.

An alternative solution is shown in Fig. 2.9 (Shinsky, 1978). Here, controllers  $C_1$  and  $C_2$ , for  $y_1$  and  $y_2$ , are both designed for using  $u_2$  as the input. We then have a selector for  $u_2$ , followed by a subtraction block which effectively does the split range control.

Controller  $C_2$  is used for controlling  $y_2$  using  $u_2$  as the input.  $C_2$  needs anti-windup because  $u_2$  is reassigned to controlling  $y_1$  when  $u_1$  saturates. Controller  $C_1$ , which controls  $y_1$ , is always active. It uses  $u_1$  to control  $y_1$  when  $u_1$  is not saturated and switches to using  $u_2$  when  $u_1$  saturates. The "extra" control element for input  $u_1$  ( $C'_1$  in Fig. 2.9) can be just a gain, but it can also contain lead-lag dynamics. Note that the subtraction block in Fig. 2.9 provides some built-in decoupling, which may be advantageous dynamically in the unconstrained case when both  $y_1$  and  $y_2$  are controlled.

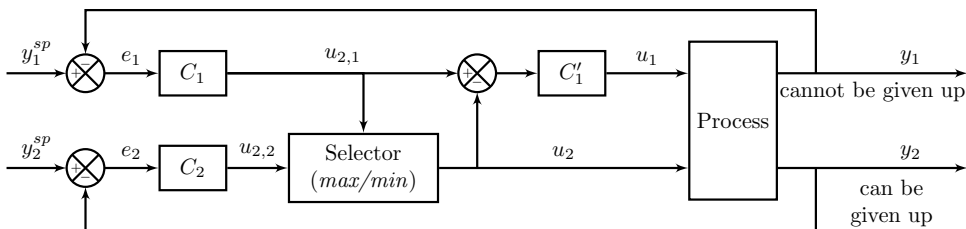


Figure 2.9: Alternative scheme proposed by Shinsky (1978) for MV to CV switching when the input saturation rule is *not* followed.

### Use of anti-windup

When using *min/max* selectors, as in *CV to CV constraint switching* (Fig. 2.2), it is necessary to implement tracking of the actual input (anti-windup) for all the controllers such that the controllers that are *not* selected do not wind up.

In *MV to MV switching* using split range control (Fig. 2.4), there is a single controller ( $C$ ), which always controls the output, so anti-windup is not needed except

if *all* the inputs are saturated, just as for a standard single-input single-output (SISO) controller. In *MV to MV switches*, when using the selector in combination with input position control, the input (valve) position controller ( $C_2$  in Fig. 2.6) winds up when it is not active, and input tracking is required for this controller.

In *MV to CV constraint switching*, when the input saturation rule is not followed (Fig. 2.8), anti-windup is necessary for the controller that usually manipulates the MV that is not coming from the split range controller ( $C_2$  in Fig. 2.8). The split range controller ( $C_1$ ) is always actively controlling the high priority CV ( $y_1$  in Fig. 2.8). If all the inputs ( $u_1$  and  $u_2$  in Fig. 2.8) saturate, anti-windup must also be implemented for  $C_1$  as for a standard single-input single-output (SISO) controller.

## 2.3 Case study I: Mixing of air and methanol

In a formaldehyde production process, air and methanol (MeOH) are mixed in a vaporizer. Air is fed using a blower with limited capacity. The model for the mixing process can be found in Appendix A.

### 2.3.1 Design of the supervisory control layer

The main controlled variable is the methanol molar fraction at the outlet of the vaporizer ( $y_1 = x_{MeOH}$ ) which should be kept at 0.1 (desired), and with a minimum value of 0.08 (more important), such that the reaction can take place. Additionally, we want to control the total mass flow ( $y_2 = \dot{m}_{tot}$ ), and in some cases to maximize it.

#### Step A1: Control objective, MVs and constraints

The controlled variables (CVs) are:

- $y_1 = x_{MeOH}$ : MeOH molar fraction
- $y_2 = \dot{m}_{tot}$ : total mass flow

The two manipulated variables (MVs) for the supervisory control layer are:

- $u_1 = \dot{m}_{air}^{sp}$ : mass flow of air
- $u_2 = \dot{m}_{MeOH}^{sp}$ : mass flow of methanol

Note that the physical MVs are the air blower rotational speed ( $\dot{\omega}_{air}$ ) and the MeOH valve opening ( $z_{MeOH}$ ), but we use a (lower) regulatory control layer with flow controllers for  $\dot{m}_{air}$  and  $\dot{m}_{MeOH}$ , which follow  $u_1 = \dot{m}_{air}^{sp}$  and  $u_2 = \dot{m}_{MeOH}^{sp}$ .

Table 2.1 shows the maximum constraint values and nominal operating conditions. Note that the valve for  $u_2 = \dot{m}_{MeOH}$  is not limited, and only  $y_1 = x_{MeOH}$  and  $u_1 = \dot{m}_{air}$  have relevant constraints.

Table 2.1: Maximum and nominal values for case study I.

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

### Step A2: Priority list of constraints

We generate the priority list for the constraints defined in step A1:

(P1) *Physical MV inequality constraints:*

$$\dot{m}_{air}^{min} \leq \dot{m}_{air} \leq \dot{m}_{air}^{max}; \text{ constraint on } u_1 \quad (2.3a)$$

$$\dot{m}_{MeOH}^{min} \leq \dot{m}_{MeOH} \leq \dot{m}_{MeOH}^{max}; \text{ constraint on } u_2 \quad (2.3b)$$

(P2) *Critical CV inequality constraints:*

$$x_{MeOH}^{min} \leq x_{MeOH} \leq x_{MeOH}^{max}; \text{ constraint on } y_1 \quad (2.4)$$

(P3) *Less critical CV and MV constraints:*

$$x_{MeOH} = x_{MeOH}^{sp}; \text{ setpoint for } y_1 \quad (2.5)$$

(P4) *Desired throughput:*

$$\dot{m}_{tot} = \dot{m}_{tot}^{sp}; \text{ setpoint for } y_2 \quad (2.6)$$

(P5) In this case there are no unconstrained degrees of freedom, and thus, there are no self-optimizing variables.

If there is no feasible solution that satisfies Eq. (2.5) or (2.6) in P3 and P4, then constraints are given up in the order P4, P3, and P2. Constraints in P1 cannot be violated for physical reasons.

**Step A3: Active constraint switches**

At the nominal operating point (defined in Table 2.1), we are able to satisfy all the constraints. It is always possible to control the MeOH concentration; that is, to satisfy (2.5) in P3.

The only relevant constraint switch happens when we reach the maximum bound on constraint (2.3a) in P1,  $u_1 = \dot{m}_{air}^{max}$ . We then lose a degree of freedom ( $u_1$ ) and, according to the *priority list for constraints*, we give up controlling the constraint with the lowest priority,  $y_2 = \dot{m}_{tot} = \dot{m}_{tot}^{sp}$  ((2.6), in P4), which is the desired throughput.

**Step A4: Base case control structure**

We have two available MVs ( $u_1$  and  $u_2$ ) for two CVs ( $y_1$  and  $y_2$ ), and we need to design the control structure. We will now consider two cases:

- Case A: We follow the *input saturation pairing rule*; thus, we pair the MV which may saturate ( $u_1 = \dot{m}_{air}$ ), with the least important CV ( $y_2 = \dot{m}_{tot}$ ). This control structure is shown in Fig. 2.10. Here there is no need for any additional logic for constraint switching, except that we need anti-windup for the air flow controller.
- Case B: There might be some operational situation that prevents us from following the *input saturation pairing rule*. In this case, we pair  $y_1 = x_{MeOH}$  with  $u_1 = \dot{m}_{air}$  and  $y_2 = \dot{m}_{tot}$  with  $u_2 = \dot{m}_{MeOH}$ .

**Step A5: Control structures for active constraints switching (Case B)**

When the *input saturation pairing rule* was not followed (case B), we implement an *MV to MV switching strategy* in combination with a *min* selector.

Fig. 2.11 shows the solution using split range control. We do not need input tracking (anti-windup) for the split range controller because  $y_1 = x_{MeOH}$  is always being controlled; that is, the selected signal in the split range controller will always be active. Anti-windup is implemented for the flow controller for  $y_2 = \dot{m}_{tot}$ , as it will wind up during the period in which it is not selected and we give-up controlling  $y_2 = \dot{m}_{tot}$ .

Fig. 2.12 shows an alternative implementation for Case B, using input (valve) position control (VPC). With this structure,  $u_1$  is reset to 95% of its maximum capacity ( $\omega_{air}^{sp} = 0.95 (\omega_{air}^{max} - \omega_{air}^{min}) + \omega_{air}^{min}$ ) by manipulating  $u_2 = \dot{m}_{MeOH}^{sp}$ . Anti-windup is required for the input (valve) position controller (VPC) that uses  $u_2$  to control  $u_1$ .



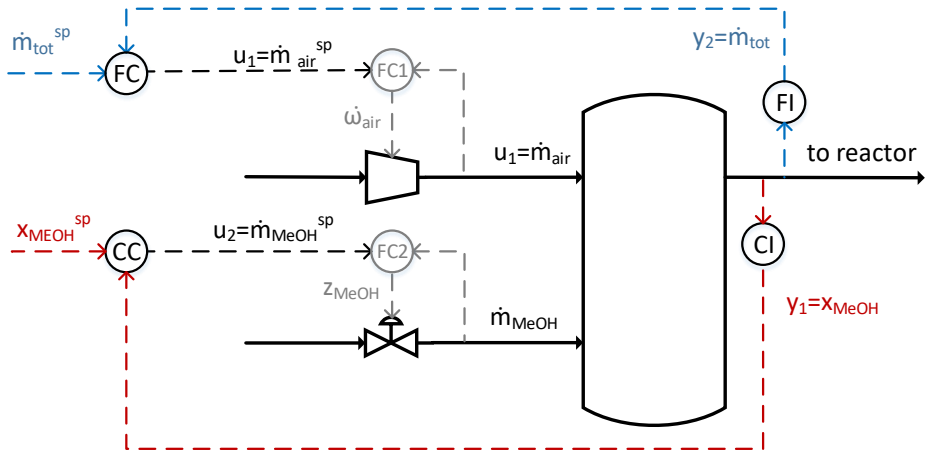


Figure 2.10: Case A: control structure for mixing of MeOH and air following the *input saturation pairing rule*. The (lower) regulatory control layer is dimmed in gray.

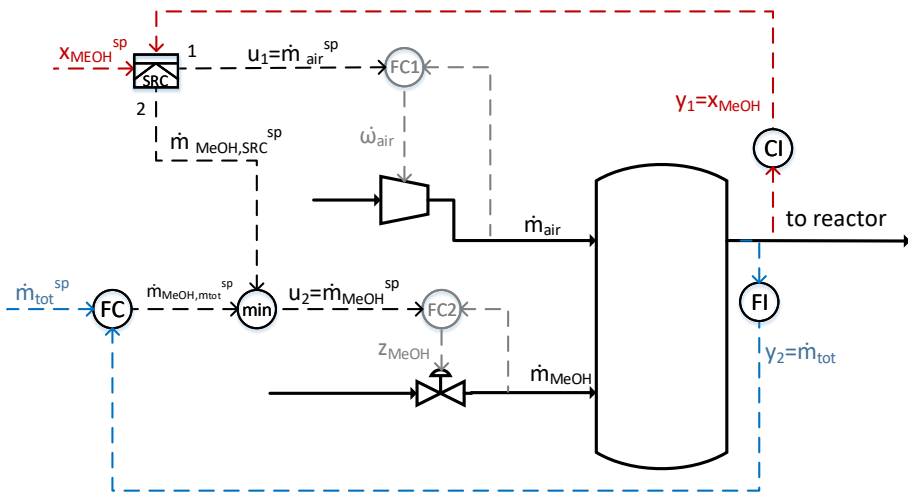


Figure 2.11: Case B-SRC. Control structure for mixing of MeOH and air when *not* following the *input saturation pairing rule* using split range control with a *min* selector.

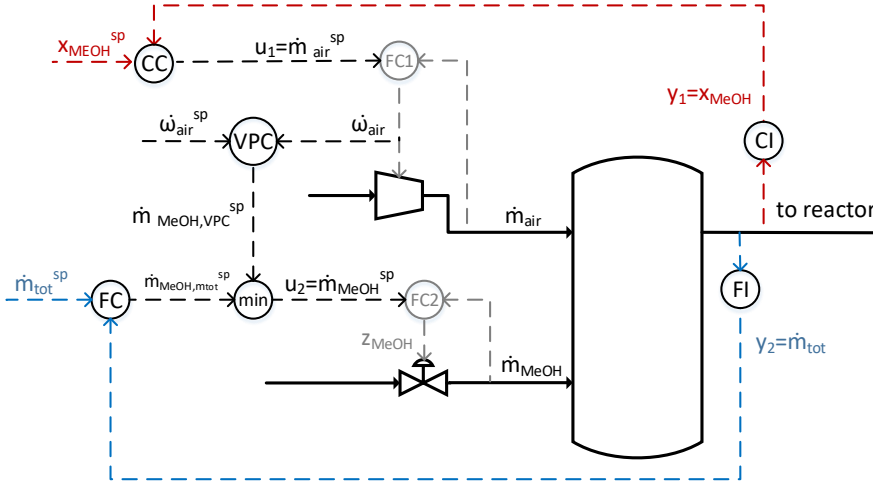


Figure 2.12: Case B-VPC. Alternative control structure for mixing of MeOH and air in Case B, using input (valve) positioning control (VPC).

### 2.3.2 Simulations

Fig. 2.13 shows simulation results for:

- Case A: pairing following the *input saturation pairing rule*, with no need of advanced control structure (Fig. 2.10).
- Case B: pairing *not* following the *input saturation pairing rule*, with no advanced control structure.
- Case B-SRC: pairing not following the *input saturation pairing rule*, but using split range control with a *min* selector (Fig. 2.11).
- Case B-VPC: pairing not following the *input saturation pairing rule*, but using input (valve) positioning control with a *min* selector (Fig. 2.12).

All the structures were tested for a step change in  $y_1^{sp} = x_{MeOH}^{sp}$  of  $-0.005$  (from  $0.100$  to  $0.095$ ) at  $t = 10$  s, followed by a  $10\%$  increase in  $y_2^{sp} = \dot{m}_{tot}^{sp}$  (from  $26860$  kg/h to  $29546$  kg/h) at  $t = 30$  s. In this period,  $y_2 = \dot{m}_{tot}^{sp}$  is not achievable, so the system should maximize the throughput ( $y_2 = \dot{m}_{tot}$ ). Finally, we bring  $\dot{m}_{tot}^{sp}$  back to its initial value at  $t = 70$  s.

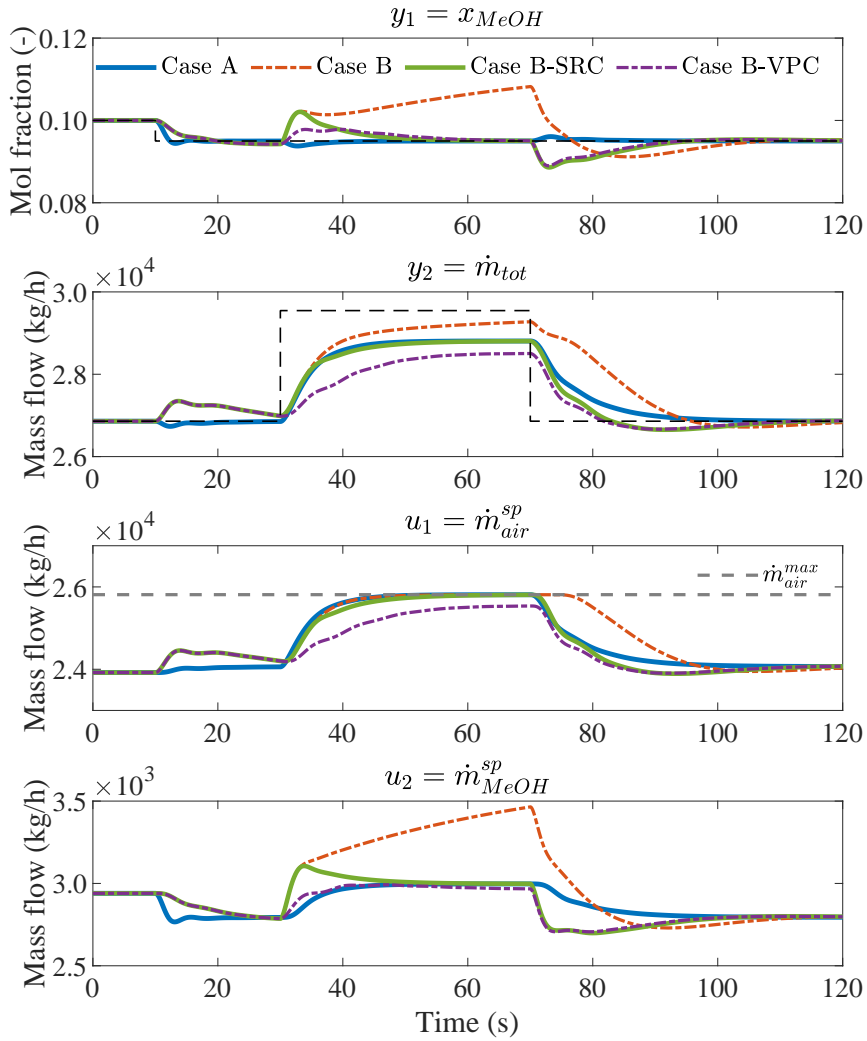


Figure 2.13: Comparison of control structures for mixing of MeOH and air. The best results are achieved with Case A and case B-SRC.

All the tunings were found using the SIMC tuning rules (Skogestad, 2003). The split range control structure was designed using the systematic procedure proposed by Reyes-Lúa et al. (2019b) (Chapter 5).

When we do not follow the input saturation pairing rule and do not implement any advanced control structure (Case B),  $y_2 = \dot{m}_{tot}$  is highest, but it comes at the expense of not keeping  $y_1 = x_{MeOH}$  at its setpoint and thus, violating its maximum constraint (see Table 2.1).

As expected, the dynamic performance is best for Case A, when we follow the *input saturation pairing rule*. This is clear by comparing the response for Case A (blue line) with those for Case B for SRC (green line) or VPC (violet dashed line) in the two upper plots in Fig. 2.13. In case A and in cases B-SRC and B-VPC, we always keep  $y_1 = x_{MeOH}$  at its setpoint and instead give up controlling  $y_2$  (throughput), which has a lower priority. In Case B-VPC, we are not able to fully maximize the throughput because the air blower ( $u_1$ ) at steady-state is limited to 95% of its capacity.

## 2.4 Case study II: Control structure for a distillation column

In this case study, we design the control structure for the conventional two-product distillation column in Fig. 2.14. This column is similar to Column A, introduced by Skogestad and Morari (1988), also described by Skogestad and Postlethwaite (2005). This column splits a binary mixture with relative volatility  $\alpha = 1.5$  and has 41 equilibrium stages, including the reboiler and a total condenser. The feed ( $F$ ) enters at stage 21.

The main assumptions are constant relative volatility, constant molar overflow, constant pressure over the entire column, vapor-liquid equilibrium on every stage, and negligible vapor holdups. The product prices are assumed independent of composition, as long as the purity specifications of 95% are satisfied. Column data and prices are given in Table 2.2. Note that the valuable product is in the bottom.

Dynamically, this distillation column has six available manipulated variables ( $F$ ,  $L$ ,  $V$ ,  $V_T$ ,  $D$ ,  $B$ ). However, the two levels and pressure must be controlled at all times for stable operation. In general, the setpoints to the regulatory controllers remain as degrees of freedom, but the two level setpoints have no steady-state effect and we assume that the pressure setpoint is constant (Skogestad et al., 1990). We choose to use bottoms flow ( $B$ ), distillate flow ( $D$ ), and cooling ( $V_T$ ) for controlling

Table 2.2: Data for distillation case study.

Variable	Units	Value
$z_F$	mol/mol	0.5
$p_F$	\$/mol	1.0
$p_B$	\$/mol	2.0
$p_D$	\$/mol	1.0
$p_V$	\$/mol	0.02 - 0.15
$x_B^{min}$	mol/mol	0.95
$x_D^{min}$	mol/mol	0.95
$V^{max}$	mol/s	4.00

levels and pressure in the regulatory layer (Fig. 2.14)<sup>3</sup>. This is the so-called *LV* configuration (Skogestad and Postlethwaite, 2005), where reflux (*L*) and boilup (*V*) are left as manipulated variables for supervisory control. In addition, the feedrate (*F*) is in principle a manipulated variable, although in most cases it is given, and its setpoint is regarded as a disturbance.

The main disturbances are the feed setpoint ( $F^{sp}$ ) and the energy price ( $p_V$ ). Then,  $d = [F^{sp}, p_V]$ , where  $F^{sp}$  may vary from 1.0 to 1.68 mol/s and  $p_V$  from 0.02 to 0.15 \$/mol. At the nominal point,  $F^{sp} = 1.0$  mol/s and  $p_V = 0.07$  \$/mol.

### 2.4.1 Design of the supervisory control layer

Let us start with the top-down economic analysis (step S1). For this distillation column with one feed stream and two products, the economic optimization problem can be written as (Jacobsen and Skogestad, 2012):

$$\begin{aligned} \min_u J(u, d) &= p_F F + p_V V - p_D D - p_B B \\ \text{s.t. } x_B &\geq x_B^{min} && \text{mol fraction of heavy component in B} && (2.7a) \\ x_D &\geq x_D^{min} && \text{mol fraction of light component in D} && (2.7b) \\ V &\leq V^{max} && \text{boilup} && (2.7c) \end{aligned}$$

$F$ ,  $V$ ,  $D$ , and  $B$  are the molar flowrates of feed, boilup, distillate and bottoms.

#### Step A1: Control objective, MVs and constraints

We have three inputs  $u = [L, V, F]$ . Relevant disturbances are  $z_F$ ,  $p_V$ ,  $F^{sp}$  and  $V^{max}$ , but for this analysis we will consider  $d = [p_V, F^{sp}]$  because we only need to

<sup>3</sup>Flow controllers for *L* and *V* are included in the regulatory layer, but are not shown in Fig. 2.14.

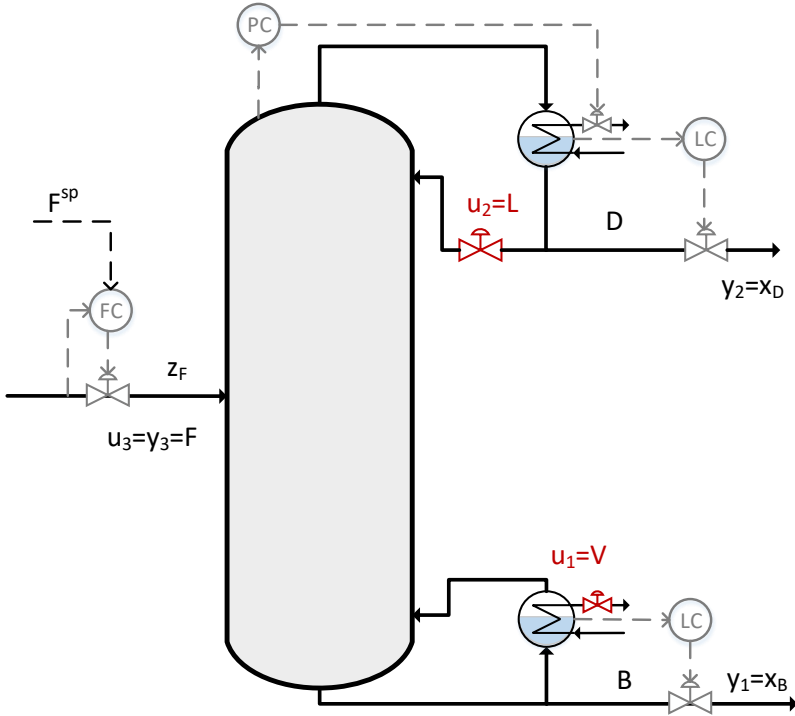


Figure 2.14: Distillation column with regulatory control layer in gray.  $u_1 = V$  and  $u_2 = L$  are MVs for the supervisory control layer.

find which active constraint switches occur, and variations in  $z_F$  and  $V^{max}$  only affect the value at which the constraints become active, but not which constraints become active.

We still have not selected the controlled variables. Since the bottom product is the most valuable, optimal operation always corresponds to having constraint (2.7a) active because this avoids product giveaway (Skogestad, 2007; Jacobsen and Skogestad, 2012), such that optimal operation is achieved when

$$y_1 = x_B = x_B^{min} \quad (2.8)$$

The less valuable distillate product is generally overpurified in order to avoid loss of the heavy component; so, constraint (2.7b) is normally not active.

Under normal operation, the optimal solution is unconstrained, and we will assume that  $x_D$  is a good self-optimizing variable, and (close to) optimal operation

is achieved when

$$y_2 = x_D = x_D^{opt}(p_V) \quad (2.9)$$

where  $x_D^{opt}$  will depend on the energy price ( $p_V$ ).

In addition, we would like to obtain a desired throughput, which is given by the equality constraint

$$y_3 = F = F^{sp} \quad (2.10)$$

Note that the feedrate ( $F$ ) is considered both an MV and a CV, and its setpoint value ( $F^{sp}$ ) is considered a disturbance (DV)<sup>4</sup>.

In addition to these three equality constraints, we should also satisfy inequality constraints (2.7b) on  $x_D$  and (2.7c) on  $V$ . This may not always be feasible and the priority list is as follows.

### Step A2: Priority list of constraints

- (P1) *Physical MV inequality constraints:* maximum boilup, constraint for  $u_2$  (2.7c) ( $V \leq V^{max}$ ).
- (P2) *Critical CV constraints:* none.
- (P3) *Less critical CV constraints:* constraint (2.7a) ( $x_B \geq x_B^{min}$ ) and (2.8) ( $x_B = x_B^{min}$ ) on bottom product composition ( $y_1$ ) and (2.7b) ( $x_D \geq x_D^{min}$ ) on top product composition ( $y_2$ ).
- (P4) *Desired throughput:* constraint (2.10) for  $y_3$  ( $F = F^{sp}$ ).
- (P5) *Self-optimizing variable:* optimum concentration of less valuable product, constraint (2.9) for  $y_2$  ( $x_D = x_D^{opt}$ ).

### Step A3: Active constraint switches

As mentioned, for the valuable bottom product, constraint (2.7a) ( $x_B = x_B^{min}$ ) is always optimally active. Assuming for now that we satisfy the throughput constraint ( $F = F^{sp}$ ), we then have two remaining inequality constraints, on  $x_D$  and  $V$ .

With  $n_c = 2$ , there are  $2^{n_c} = 4$  possible active constraint regions:

- Region I: only  $x_B$  active (constraint (2.7a))

---

<sup>4</sup>Nominally, the MV and the CV are the same ( $y_3 = u_3 = F^{sp}$ ), but in some cases, we must give up controlling  $y_3$  and its setpoint, and instead use the MV ( $u_3$ ) to control a CV with higher priority ( $y_2$  in Fig. 2.18 and  $y_1$  in Fig. 2.21).

- Region II:  $x_B$  and  $V$  active (constraints (2.7a) and (2.7c))
- Region III:  $x_B$  and  $x_D$  active (constraints (2.7a) and (2.7b))
- Region IV:  $x_B$ ,  $x_D$  and  $V$  active (constraints (2.7a), (2.7b) and (2.7c))

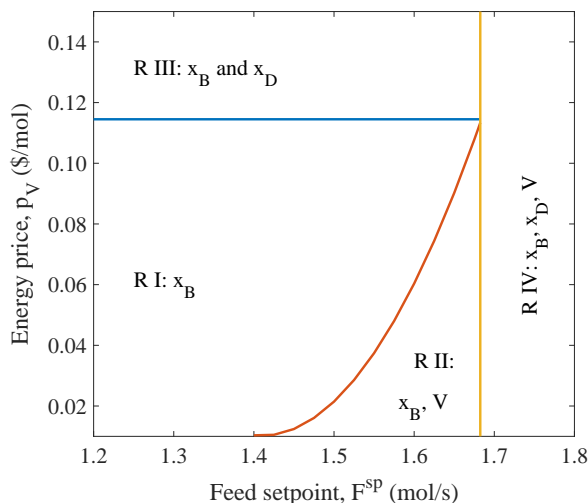


Figure 2.15: Active constraint regions for binary distillation column with the bottom as valuable product.

Region IV, with three active constraints, is infeasible if we also want to have a given throughput ( $F = F^{sp}$ ), because then there are only two available degrees of freedom, and we cannot satisfy three active constraints. Therefore, region IV will correspond to operation at maximum throughput, where we give up achieving  $F = F^{sp}$ . Fig. 2.15 shows the actual active constraint regions for this system, obtained by numerical optimization of the process (see Appendix B). We stress that we include this diagram for illustration purposes, and it is not required to design the control structure. The transition between regions I and III, which corresponds to  $x_D$  reaching  $x_D^{min}$ , is a horizontal line because the column stage efficiency is assumed constant, independent of flow. At  $F = 1.68$  mol/s, all three inequality constraints in (2.7) become active (region IV) and we have to give up controlling  $F = F^{sp}$ .

#### Step A4: Base case control structure

The nominal operating point is in region I, with a low energy price and a low feed rate. The only active inequality constraint is (2.7a) and we must keep  $x_B = x_B^{min}$ . We also control the feedrate (constraint (2.10)) and we select  $x_D$  as the self-optimizing variable associated with the remaining unconstrained degree of freedom (constraint



(2.9)). The optimal concentration  $x_D^{opt}(p_V)$  is given by an equation (see Appendix B). We want to use single-loop control so we have to select pairings. With the standard  $LV$ -configuration in Fig. 2.14, it is obvious that the best pairing is to use boilup ( $V$ ) to control the bottom composition ( $x_B$ ) and reflux ( $L$ ) to control the top composition ( $x_D$ ), as shown in Fig. 2.16.

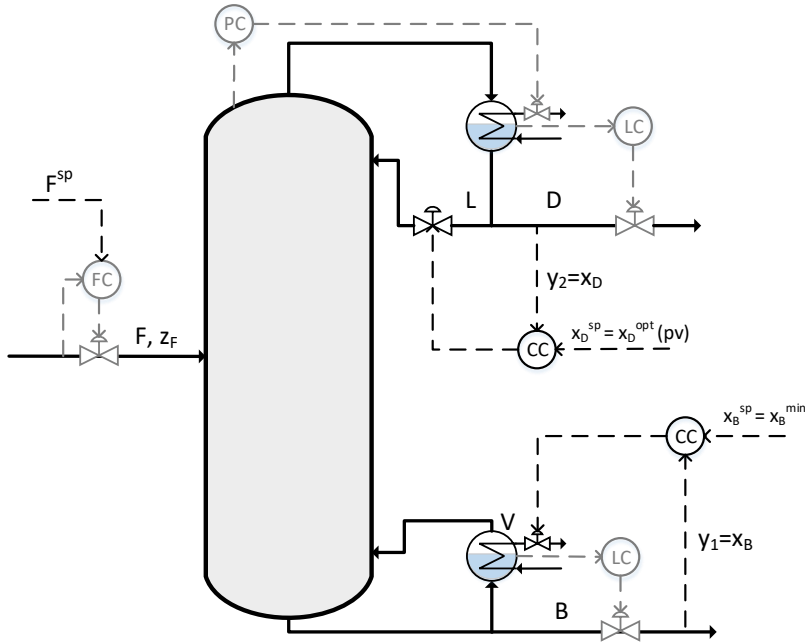


Figure 2.16: Base case control structure for distillation column (region I).

### Step A5: Control structures for active constraints switching

We used the "obvious" pairing following the *pair close* rule for the base case structure in Fig. 2.16. However, this implies that we did not follow the *input saturation pairing rule* since  $u_2 = V$ , which may saturate, is controlling  $y_1 = x_B$ , which is a more important CV than  $y_2 = x_D$ . As we increase the throughput ( $d = F^{SP}$  increases), and the required boilup increases, we eventually reach  $V = V^{max}$  and enter region II.

Following the priority list of constraints, we must then give up controlling the self-optimizing variable  $y_2 = x_D$  and start using  $u_1 = L$  to control  $y_1 = x_B$ . We choose to use split range control with a *min* selector as our *MV to CV constraint switching* strategy, as shown in Fig. 2.17. Alternatively, we could have implemented an input (valve) position control scheme, using  $L$  to prevent  $V$  from saturating.

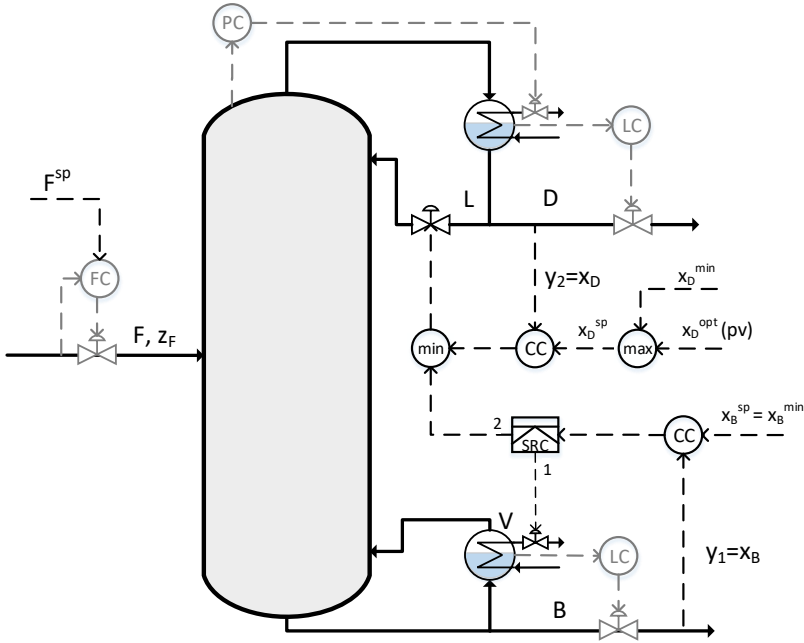


Figure 2.17: Control structure for distillation column for regions I, II and III.

If the energy price for  $V$  ( $p_V$ ) increases, overpurifying the distillate is less favorable and eventually we enter region III, where the constraint for  $x_D(p_V)$  (2.7b) becomes active, and  $x_D = x_D^{min} = 0.95$ . This switch is achieved using a *max* selector for  $x_D$ . The control structure in Fig. 2.17 works for regions I, II and III.

In order to also operate at maximum capacity and also satisfy all three constraints in (2.7) (region IV), we need to give up controlling  $F = F^{sp}$ . Thus, we need a *CV to CV constraint switching* strategy to switch between using  $u_3 = F$  from controlling  $F = F^{sp}$  to controlling  $x_D = x_D^{min}$ . One simple modification of the control structure in Fig. 2.17 is to add a second controller for  $x_D$  (with setpoint  $x_D^{min} + \Delta x_D$ ) and a *min* selector to switch between CV constraints on  $F$  and  $x_D$ . We already have another controller using  $u_2 = L$  to control  $y_2 = x_D^{min}$  in region III, so we need to introduce a back-off ( $\Delta x_D$ ) to make sure that we activate the switch to use  $u_3 = F$  only when needed (region IV). We have  $x_D^{min} = 0.95$ , and select  $\Delta x_D = -0.01$ . Fig. 2.18 shows the suggested control structure valid for all regions.

Table 2.3 shows how each of the MVs ( $L$ ,  $V$  and  $F$ ) is used in each of the active constraint regions when we use the control structure in Fig. 2.18. In region II,  $y_2 = x_D$  is "floating", that is, we are not actively controlling  $x_D$ . Note that

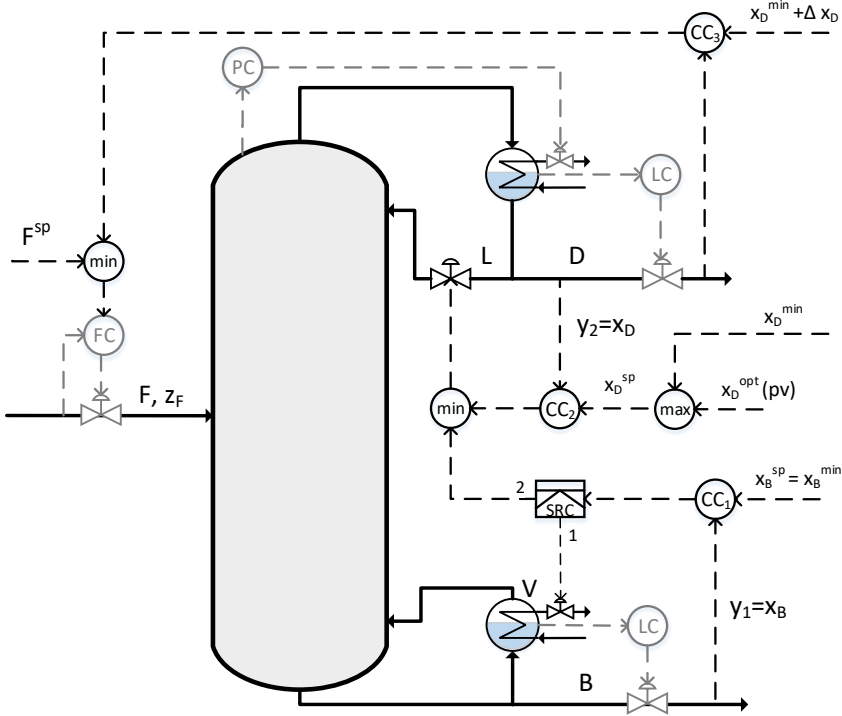


Figure 2.18: Control structure for distillation column for all regions (I, II, III, IV).

composition controllers for  $x_D$  ( $CC_2$  and  $CC_3$  in Fig. 2.18) will not be active at the same time due to the difference in setpoints ( $\Delta x_D$ ).

Table 2.3: Pairings in Fig. 2.18 for each of the active constraint regions.

Region	$L$	$V$	$F$
R I	$x_D = x_D^{opt}$	$x_B = x_B^{min}$	$F = F^{sp}$
R II	$x_B = x_B^{min}$	$V = V^{max}$	$F = F^{sp}$
R III	$x_D = x_D^{min}$	$x_B = x_B^{min}$	$F = F^{sp}$
R IV	$x_B = x_B^{min}$	$V = V^{max}$	$x_D = x_D^{min} + \Delta x_D$

## 2.4.2 Simulations

To test the control structure in Fig. 2.18, we first need to find the self-optimizing setpoint for  $x_D^{opt}(p_V)$  to use in region I. Using Fig. B.3 in the Appendix B, we observe

that there is an almost linear relation<sup>5</sup> between  $x_D^{opt}$  and  $p_V$  in region I.

For the simulations, we start at  $F^{sp} = 1.5$  mol/s and  $p_V = 0.07$  \$/mol, which is inside region I. Then, at  $t = 10$ s, we enter region II by setting  $F^{sp} = 1.65$  mol/s. At  $t = 50$  min, we enter region III by setting  $p_V = 0.13$  \$/mol. Finally, at  $t = 100$  min, we enter region IV by setting  $F^{sp} = 1.75$  mol/s.

Fig. 2.19 shows the simulation results. The changes in active constraint region are marked with vertical gray dashed lines. As expected (see Table 2.3), in region II we give up controlling  $x_D$  when  $V = V^{max}$  and we switch to using  $L$  ( $L_{x_B}$ ) to control  $x_B$ . In region III, with  $V < V^{max}$ , we use  $V$  to control  $x_B$  and  $L$  ( $L_{x_D}$ ) to keep  $x_D = x_D^{min}$ .

Fig. 2.20 shows the value of the cost ( $J$ ) as a function time.

## 2.5 Discussion

### 2.5.1 Optimal operation without a model

In the proposed procedure, we do not need to know the actual value at which each constraint activates, but we need to know which constraints will activate. The switching between active constraints is done online using feedback. In many cases, expected constraint switches can be deduced using engineering insight (Jacobsen and Skogestad, 2012).

It is common to find cases in which optimal operation is the same as maximum throughput. If we can identify the bottleneck and control it, then we do not need to perform an optimization procedure to maximize throughput (Skogestad, 2004; Aske et al., 2007). In case study I, we know that by keeping  $\dot{m}_{air}^{max}$ , and thus, maximizing the total outlet flow, we are operating at optimum conditions. In case study II, operating with the active constraints in region IV will maximize throughput.

### 2.5.2 "Opposing pairing rules"

Sometimes there are *pairing rules* that oppose. In step A4 of case study II (distillation column) the pairing suggested by the *pair close rule* is not the same as the pairing suggested by the *input saturation pairing rule*. In these cases, we have two options:

1. Follow the pairing rule that leads to the structure that will have the better dynamic behavior or most of the time (*pair close rule*).

---

<sup>5</sup>The linear approximation of  $x_D^{opt}$  as function of  $p_V$  in region I is  $x_D^{opt} \approx 0.996 - 0.384p_V$ . We use this equation to calculate  $x_D^{sp}$ .

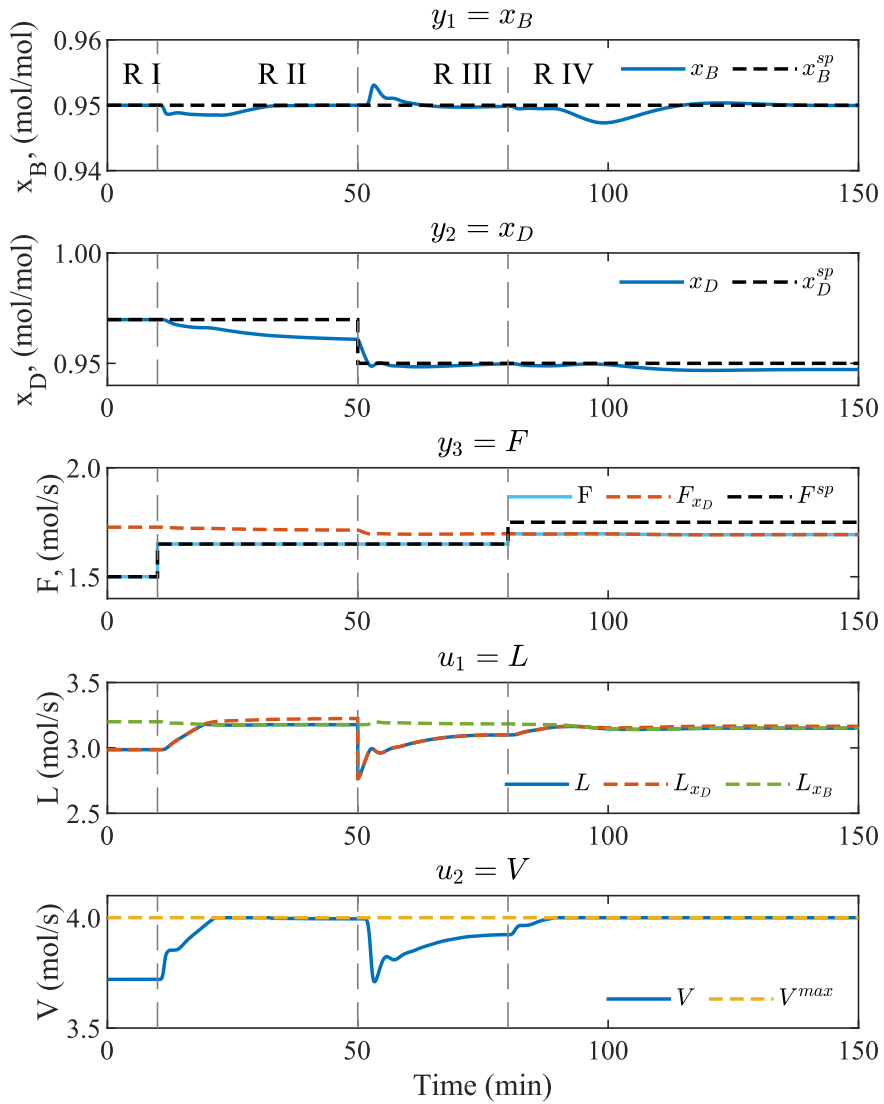


Figure 2.19: Simulation for structure in Fig. 2.18 for case study II.

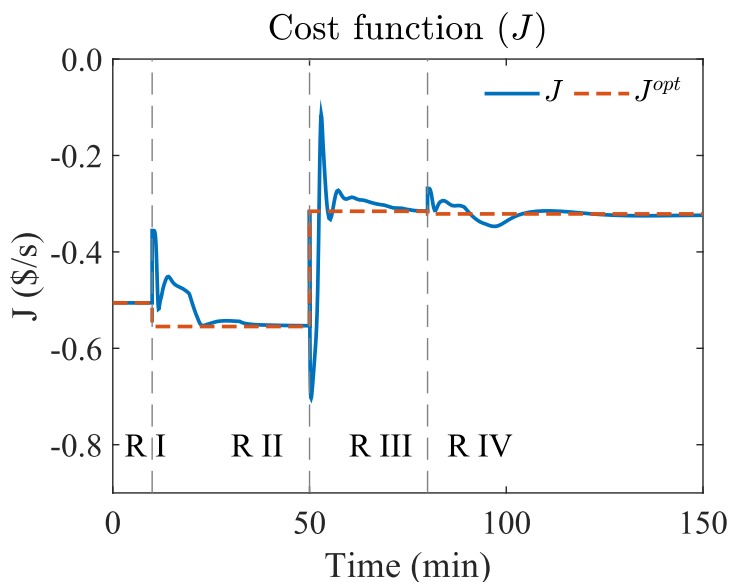


Figure 2.20: Cost for distillation column case study (which should be minimized).

2. Follow the pairing rule that requires less loop reconfiguration when we switch among the relevant active constraint regions (*input saturation pairing rule*).

The decision will depend on each particular case. In case study II, we chose to follow the pairings suggested *pair close* rule, because it gives a better dynamic behavior and we consider that the process will normally operate in region I.

### 2.5.3 Alternative control structures

In step A5 of the proposed procedure, there may be alternative options that achieve the required active constraint switches and achieve the same steady state. However, the alternative control structures, may differ in dynamic behavior.

For example, in case study II, we proposed the control structure in Fig. 2.18. An alternative structure is shown Fig. 2.21, in which we use a split range controller for  $x_B$  with three MVs ( $V$ ,  $L$  and  $F$ ). The numbers 1, 2, and 3 in the split range block (SRC) refer to the order in which each MV is used.

1.  $y_1 = x_B$  is normally controlled using  $u_1 = V$  in region I.
2. If  $V$  saturates (region II), we switch to using  $u_2 = L$ , and

3. if  $L$  has to control  $y_2 = x_D^{min}$  then we switch to using  $u_3 = F$  to control  $y_1 = x_B$ .

The structure in Fig. 2.21 is better from a dynamic point of view in region IV because it is better to use  $F$  rather than  $L$  to control  $x_B$ .

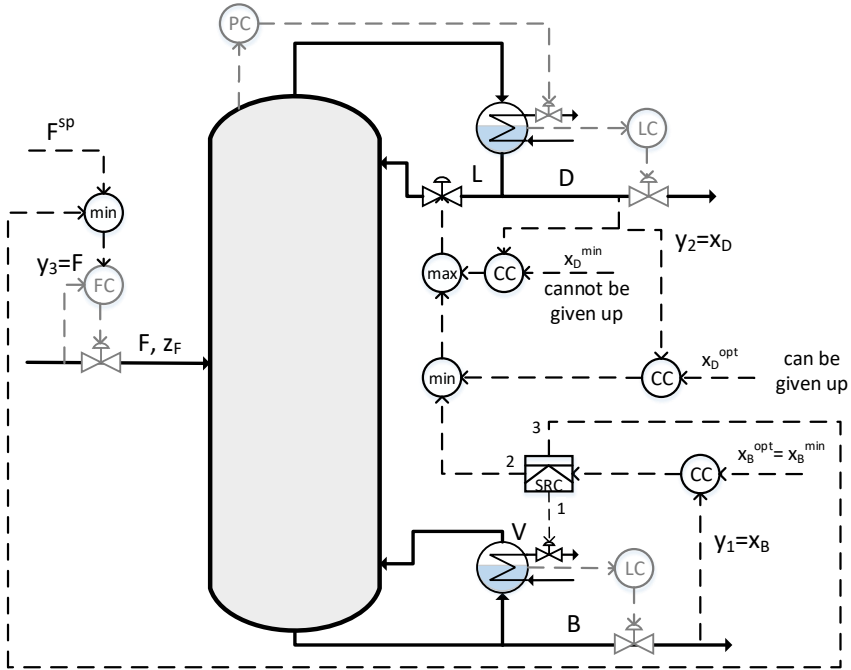


Figure 2.21: Alternative control structure for distillation column, all regions. This structure behaves differently from Fig. 2.18 in region IV.

## 2.6 Concluding remarks

We introduced a systematic procedure to design constraint switching schemes using classical controllers and logics. We distinguish between three kinds of constraint switches:

- CV to CV constraint switching: use selectors
- MV to MV constraint switching: use split-range control or alternatively controllers with different setpoints or input (valve) position control.

- MV to CV constraint switching: use nothing if the *input saturation pairing rule* is followed; otherwise, use an MV to MV scheme with a selector to take over control when the main MV saturates.

In the two presented case studies we used single-loop PID-based control structures to achieve steady-state optimal operation, despite changes in active constraint regions.



# Chapter 3

## Control structure design for CO<sub>2</sub>-refrigeration system with heat recovery

This chapter is based on:

Reyes-Lúa, A., Andreasen, G., Larsen, L. F. S., Stoustrup, J., and Skogestad, S. (2019a). Optimal operation of a CO<sub>2</sub>-refrigeration system with heat recovery. In *Proceedings of the 29th European Symposium on Computer Aided Process Engineering (ESCAPE)*, Eindhoven, Netherlands. Computer-aided chemical engineering

In this chapter, we design a PI(D)-based control structure for a generic supercritical CO<sub>2</sub>-refrigeration system with parallel compression, based on systems for supermarket use. In order to maximize energy efficiency, this system has a “heat-recovery” function, in which part of the heat rejected at high pressure and temperature can be recovered to provide heating. Operating conditions and active constraints are strongly affected by seasonal requirements and ambient temperature. Thus, it is necessary to find a control structure that satisfies operational boundaries and maintains steady-state (near-)optimal operation with different sets of active constraints. In this work, we apply the procedure described in Chapter 2 to define such control structure.

### 3.1 Introduction

An appropriately designed control structure should maintain *near*-optimal operation, also when there are disturbances which cause the system to operate under conditions different than the design point. Optimal operation of a process in the presence of disturbances could be maintained using optimization-based control.

As explained in Chapter 2, in some cases it is possible to design and implement advanced PI(D)-based control structures that also maintain steady-state optimal operation when constraints are reached. The advantage of such a PI(D)-based control structure compared to optimization-based control is simpler tuning and independence of an explicit model for every system.

CO<sub>2</sub>-refrigeration systems with parallel compression are environmentally attractive. Finding optimal design and operating conditions is an ongoing area of research (Gullo et al., 2018). In order to maximize energy efficiency, some systems have a “heat recovery” function, in which part of the heat rejected at high pressure and temperature can be recovered to provide heating (e.g. district heating or tap water) (Sawalha, 2013). The available energy can be increased by increasing the high pressure. This is achieved, however, at the expense of a higher compression work.

### 3.2 Description of the CO<sub>2</sub>-refrigeration system with heat recovery

A flow diagram of the analyzed CO<sub>2</sub>-refrigeration cycle with parallel compression and heat recovery is shown in Fig. 3.1, and the pressure-enthalpy diagram is shown in Fig. 3.2. The main function of this system is to provide cooling ( $\dot{Q}_{ev}$ ) and maintain the desired cabinet temperature ( $T_{cab}$ ) via heat exchange in the evaporator, which operates at low pressure ( $P_l$ ). Low-pressure CO<sub>2</sub> in vapor phase ① is compressed to supercritical conditions ② (high pressure ( $P_h$ ) and high temperature). The available energy ( $\dot{Q}_{hr}$ ) may be used to heat tap water in the heat recovery section ③. Excess heat ( $\dot{Q}_{gc}$ ) is rejected to the ambient air in the gas cooler.

High-pressure CO<sub>2</sub> ④ is expanded to an intermediate (sub-critical) pressure ( $P_{IP}$ ) ⑤ in the high-pressure valve ( $V_{hp}$ ). Vapor ⑥ and liquid ⑦ CO<sub>2</sub> are separated in the liquid receiver. The evaporator valve ( $V_{ev}$ ) regulates the flow of liquid CO<sub>2</sub> from the receiver to the evaporator ⑧. By opening and closing  $V_{hp}$  and  $V_{ev}$ , we regulate the refrigerant charge (mass) at the high and low pressures. Vapor CO<sub>2</sub> from the liquid receiver ⑥ is recycled to the high-pressure side either via parallel compression ( $K2$ ) or the intermediate pressure valve ( $V_{IP}$ ) and the main compressor ( $K1$ ). The total compression work can be reduced by utilizing the parallel compressor instead of the intermediate pressure valve and the main compressor.

#### 3.2.1 High-pressure side

In the supercritical region, there is no saturation condition and the pressure is independent of the temperature. From the control point of view, this means that it is necessary to control the high pressure ( $P_h$ ), since it influences the gas cooler exit enthalpy and, consequently, the evaporator inlet enthalpy and  $\dot{Q}_{ev}$ . In other words,  $P_h$  will determine the specific refrigeration capacity. As  $P_h$  is determined by the relationship between refrigerant charge, inside volume and temperature in the high-pressure side, we can actively control it using  $V_{hp}$  (Kim et al., 2004).

In the case of a refrigeration system, the coefficient of performance (COP) is defined as the ratio between the cooling provided in the evaporator and the total

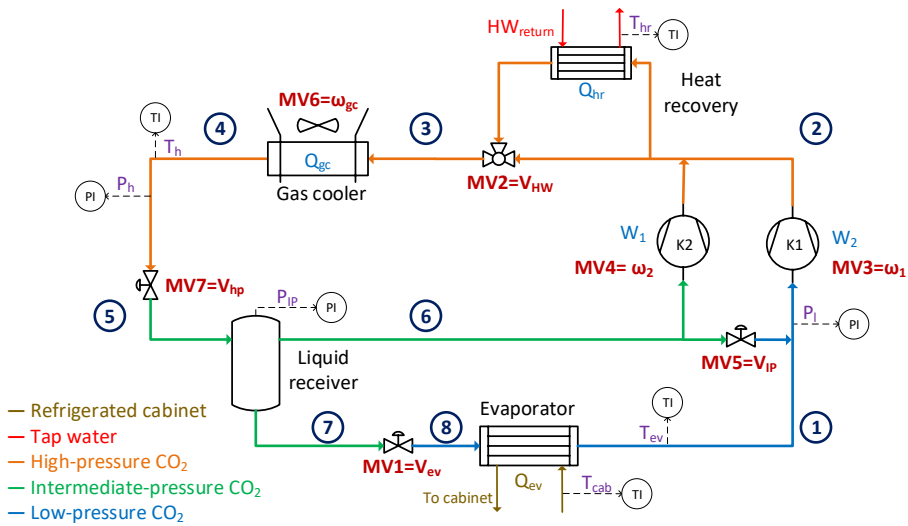


Figure 3.1:  $CO_2$ -refrigeration system with parallel compression and heat recovery. There are seven available manipulated variables (MV).

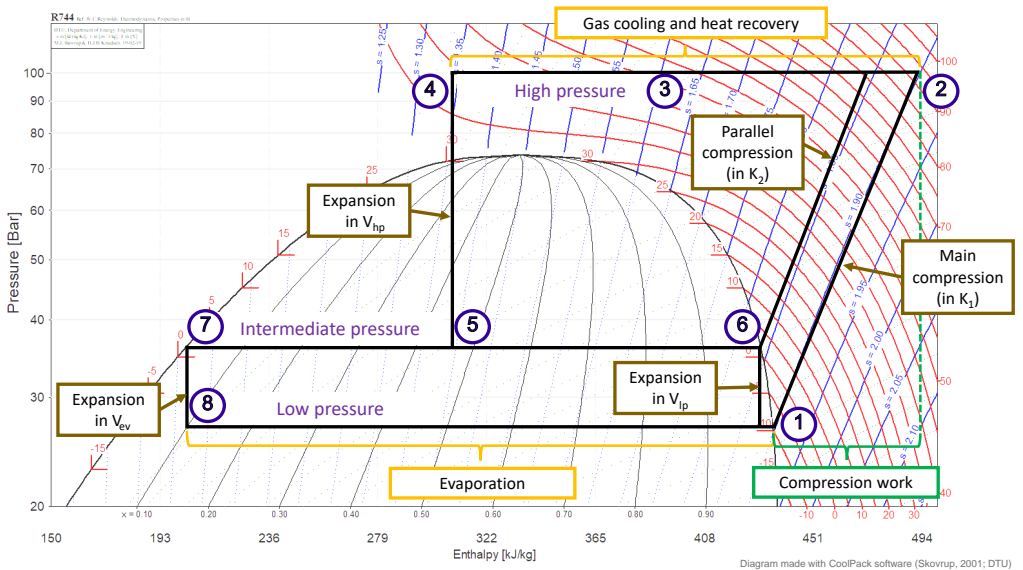


Figure 3.2: Pressure-enthalpy diagram of  $CO_2$ -refrigeration system with parallel compression.

compression work<sup>1</sup>,

$$COP = \frac{\dot{Q}_{ev}}{W_s}$$

It is relevant to analyze the effect of the high pressure ( $P_h$ ) on the COP.  $P_h$  is the pressure for ②, ③ and ④<sup>2</sup>. As the isentropic compression line in the pressure-enthalpy diagram (blue lines in Fig. 3.2) is close-to-linear, compression work will (almost) linearly increase as  $P_h$  increases. On the other hand, in the supercritical region the isotherm (red lines in Fig. 3.2) becomes steeper with pressure, reducing the capacity enhancement from a given increase in  $P_h$ . For this reason, the COP reaches a maximum above which the added capacity no longer fully compensates for the additional work of compression. Thus, there is an optimal high pressure ( $P_h$ ) that maximizes COP (Nekså, 2002).

Note that in the supercritical region, in point ④, at a fixed pressure, a small change in refrigerant exit temperature can produce a large change in gas cooler exit enthalpy and, consequently, in the evaporator inlet enthalpy (and  $\dot{Q}_{ev}$ ). Therefore the COP is very sensitive to variations in the CO<sub>2</sub> temperature at the exit of the gas cooler. Liao et al. (2000); Jensen (2008); Sawalha (2013) have shown that the optimal set-point for the high pressure ( $P_h$ ) should be corrected by the outlet temperature of the gas cooler ( $T_h$ ).

### 3.2.2 Heat recovery section

Part of the heat rejected at high pressure can be recovered to provide hot water in the heat recovery section. For supercritical CO<sub>2</sub>, heat is rejected at gliding temperature (i.e. the temperature decreases from ② to ④). As the temperature profile of the CO<sub>2</sub> matches the heating-up curve of water (sensitive heat), the use of supercritical CO<sub>2</sub> to heat up water has low thermodynamic losses and high efficiency (Kim et al., 2004). As it can be deduced from Fig. 3.2, the available heat for recovery in the supercritical region is much higher than with sub-critical CO<sub>2</sub>. We can also observe that increasing the high pressure increases the available heat for recovery, but this is at the expense of a higher compression work.

---

<sup>1</sup>In Figures 3.1 and 3.2,  $\dot{Q}_{ev}$  is the enthalpy difference between ⑧ and ①, and  $W_s$  is the enthalpy difference between ① and ②

<sup>2</sup>We are neglecting pressure drops in piping and heat exchangers and expansions are isenthalpic.

### 3.3 Design of the supervisory control layer

In this case, we want to maximize the coefficient of performance (COP), subject to the system itself and operational constraints:

$$\min_u -COP(u, x, d) = -(\dot{Q}_{ev} + \dot{Q}_{hr})/(W_1 + W_2)$$

$$\text{s.t. } f(u, x, d) = 0 \quad \text{system equations (model)} \quad (3.1a)$$

$$g(u, x, d) \leq 0 \quad \text{operational and physical constraints} \quad (3.1b)$$

where  $x$  are the internal states,  $u$  are the manipulated variables (MV),

$$u = [\omega_1, \omega_2, \omega_{gc}, z_{Vev}, z_{Vhp}, z_{VHW}, z_{VIP}]^T, \quad (3.2)$$

and  $d$  are the disturbances,

$$d = [\dot{Q}_{ev}, \dot{Q}_{hr}]^T \quad (3.3)$$

Constraints in Eq.(3.1b) are related to minimum and maximum values for pressure ( $P_i$ ), motor velocities ( $\omega_i$ ) and valve openings ( $z_i$ ). Eq.(3.1b) also includes a minimum value for the evaporator outlet temperature ( $T_{sh}$ ) to avoid overflowing.

Additionally, we need to satisfy cooling and heating requirements. Eq. (3.1b) also contains:

$$\dot{Q}_{ev}^{min} \leq \dot{Q}_{ev} \quad (3.4a)$$

$$\dot{Q}_{hr}^{min} \leq \dot{Q}_{hr} \quad (3.4b)$$

In practice, the cooling requirements ( $\dot{Q}_{ev}^{min}$ ) are set by  $T_{cab}^{sp}$ , while the heating requirements ( $\dot{Q}_{hr}^{min}$ ) are set by  $T_{hr}^{sp}$ .

#### Step A1: Control objectives, MVs and CVs for supervisory layer

The analyzed system has seven available manipulated variables ((3.2)), MVs in Fig. 3.1. These degrees of freedom can be used to achieve optimal operation. Note that  $\omega_2$  and  $z_{VIP}$  are not independent, as either would have a similar effect in  $P_{IP}$ .

Available measurements are shown in purple in Fig. 3.1:

$$y = [T_{cab}, T_{ev}, P, T_{hr}, T_h, P_h, P_{IP}]^T \quad (3.5)$$

Optimally, we will not use more energy ( $W_s$ ) to provide more heating or cooling than needed; thus,

$$T_{cab} = T_{cab}^{sp} \quad (3.6a)$$

$$T_{hr} = T_{hr}^{sp} \quad (3.6b)$$

**Step A2: Priority list**

We define the priority list below considering that:

- there are cooling and heating requirements;  $0 < \dot{Q}_{ev}^{min}$  and  $0 < \dot{Q}_{hr}^{min}$ ,
- cooling requirements must be met always, and
- heating requirements should also be met, but may be given up.

- (P1) *Physical MV inequality constraints*: maximum and minimum values for every MV in (3.2). These constraints are included in Eq. 3.1b, specifically:  $\omega_j^{min} \leq \omega_j \leq \omega_j^{max} \forall j$  and  $z_k^{min} \leq z_k \leq z_k^{max} \forall k$ .
- (P2) *Critical CV constraints*: constraints in Eq.(3.1b) related to pressure ( $P_i$ ),  $P_i \leq P_i^{max} \forall i$ ,  $(P_{IP} - P_l)^{min} \leq (P_{IP} - P_l)$ . Here we also include  $T_{sh} > T_{sh}^{min}$ , related to  $T_{ev}$ <sup>3</sup>.
- (P3) *Less critical CV constraints*: cooling requirements,  $T_{cab}$ , (Eq. (3.6a)).
- (P4) *Desired throughput*: desired heating requirements,  $T_{hr}$ , (Eq. (3.6b))<sup>4</sup>.
- (P5) *Self-optimizing variable*: optimal set-points for the remaining CVs ( $P_h$ ,  $T_h$ ,  $P_{IP}$ ,  $P_l$ ).

**Step A3: Active constraint switches**

There are three relevant operating regions:

1. "Unconstrained" case: corresponding to spring/fall operation
2. *Maximum heating*: corresponding to winter, when  $\dot{Q}_{hr} = \dot{Q}_{hr}^{max}$
3. *Maximum cooling*: corresponding to summer, when  $\dot{Q}_{ev} = \dot{Q}_{ev}^{max}$

We do not consider  $\dot{Q}_{ev} = \dot{Q}_{ev}^{min}$ , as it corresponds to shut-down.  $\dot{Q}_{hr} = \dot{Q}_{hr}^{min}$  is included in the "unconstrained" and maximum cooling cases. Relevant switches are from from "Unconstrained" to "Maximum heating", and vice-versa, as well as from "Unconstrained" to "Maximum cooling", and vice-versa.

---

<sup>3</sup>These constraints are related to the integrity of the equipment.

<sup>4</sup>When we are heating, the high pressure  $P_h$  and the flowrate are defined by heating requirements; thus, the throughput manipulator is  $T_{hr}$ .

**Step A4: Control structure for normal operation, "Unconstrained" case**

This is the base case and we can satisfy every constraint. We use  $MV1=z_{V_{ev}}$  to control  $T_{cab}$ , and  $MV2=z_{V_{HW}}$  to control  $T_{hr}$ . We have five remaining unconstrained degrees of freedom, two of which are not independent ( $\omega_2$  and  $z_{V_{IP}}$ ). We pair these degrees of freedom as follows:

1.  $MV3=\omega_1$  controls  $P_l$ .  $P_l^{sp}$  is found by optimization (self-optimizing variable).
2. The parallel compressor ( $MV4=\omega_2$ ) and  $MV5=z_{V_{IP}}$  may be used to control the pressure in the liquid receiver ( $P_{IP}$ ).  $P_{IP}^{sp}$  is defined by optimization. To reduce the flow through the main compressor (and the total  $W_s$ ), we favor using  $K_2$  over  $V_{IP}$ . Normal operation is using  $MV4=\omega_2$ , with  $MV5=z_{V_{IP}} = 0$ .
3.  $MV6=\omega_{gc}$  controls  $T_h$ , in (4).  $T_h^{sp}$  is defined by optimization.
4.  $MV7=z_{V_{hp}}$  controls  $P_h$ . As explained in Section 3.2.1, the set-point is a linear combination (H) of  $P_h$  and  $T_h$ , which is a self-optimizing variable.

This basic control structure is shown in Fig. 3.3.

**Step A5: Control structure for active constraint switching**

The control structure in Fig. 3.4 uses the strategies described in Section 2.2.5 to handle the active constraint switches when maximum heating and maximum cooling are required.

**Maximum heating**

In Fig. 3.3,  $T_{hr}$  is controlled using  $MV2 = z_{V_{HW}}$ . When  $V_{HW}$  becomes fully open (no bypassing of the heat recovery section), we must switch the manipulated variable to continue controlling  $T_{hr}$ . This is handled using split-range control with selectors (MV to CV constraint switching). First, we switch to  $MV7=z_{V_{hp}}$  as manipulated variable and increment the available heat for recovery by increasing  $P_h$ . To implement this, we include a selector for the set-point of the high-pressure controller. Once we reach  $P_h^{max}$ , we get additional capacity for the heat-recovery section by increasing  $T_h$ , using  $MV6=\omega_{gc}$  as manipulated variable. This will increase mass flow rate through the compressors and, as consequence, the discharge temperature.

If we continue to increase  $T_h$  to satisfy  $T_{hr} = T_{hr}^{sp}$ , at some point, liquid in the low-pressure section may be insufficient and  $V_{ev}$  will reach its maximum opening. If this happens  $MV1=z_{V_{ev}}$  will not be able to maintain  $T_{cab} = T_{cab}^{sp}$ , which has a *higher priority* than  $T_{hr} = T_{hr}^{sp}$ . Alternatively, the compressors could reach their maximum capacity ( $\omega_j^{max}$ ) due to the increased mass flow. Compressor capacities

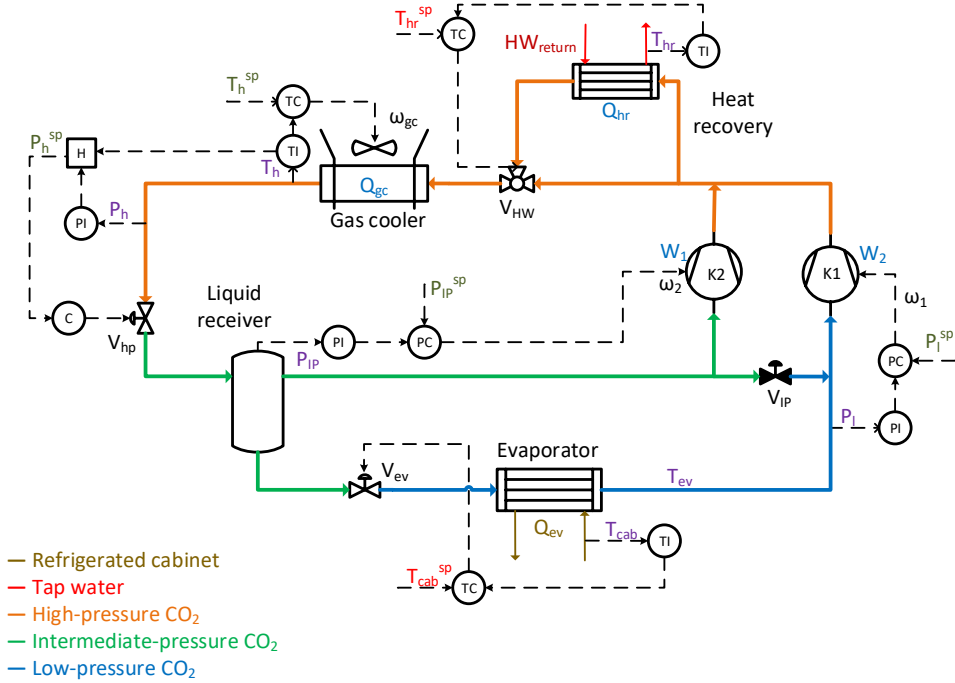


Figure 3.3: Basic control structure for the CO<sub>2</sub>-refrigeration system with heat recovery.

( $\omega_j \leq \omega_j^{max}$ ) are at the top of the *priority list*. Therefore, we need to implement an MV to CV constraint switching strategy. We choose valve-position controllers (VPC) in combination with a *min* selector, which will prevent  $T_h$  from increasing in such a way that either the valve ( $z_{V_{ev}}$ ) or the compressors ( $\omega_1$  or  $\omega_2$ ) saturate.

In order to assure that the evaporator is not over-flooded, which may cause damage to the compressors, we include a controller for the evaporator outlet temperature, with  $T_{sh}^{sp} = T_{sh}^{min}$ .  $T_{sh} > T_{sh}^{min}$  has a higher priority than  $T_{cab} = T_{cab}^{sp}$ . We implement a CV to CV switching strategy and use a *min* selector to define the set-point for MV1= $z_{V_{ev}}$ . The flow going through the secondary compressor ( $K_2$ ) and  $V_{IP}$  will be defined by the self-optimizing set-point for the intermediate pressure ( $P_{IP}^{sp}$ ). As already mentioned, during normal operation, we use  $K_2$  to control  $P_{IP}$  and we keep  $V_{IP}$  closed. However, it may be that the flow going through  $K_2$  is too low and we reach  $\omega_2^{min}$ . Then, we have to use  $V_{IP}$  to control  $P_{IP}$ . We handle this with a split-range controller, which is an MV to MV switching strategy.



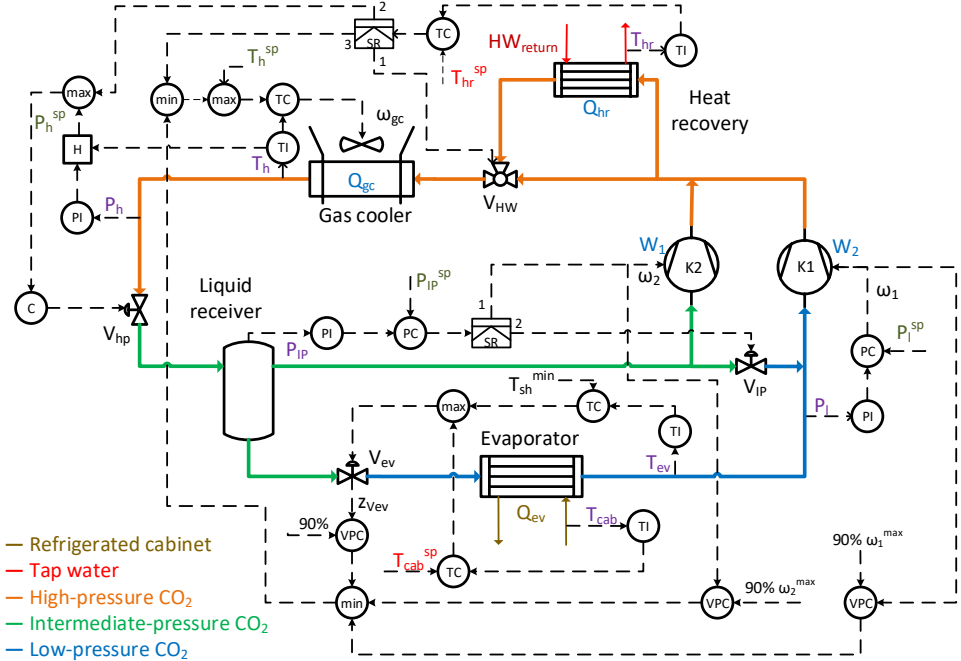


Figure 3.4: Control structure for the CO<sub>2</sub>-refrigeration system with heat recovery and parallel compression. This control structure handles the three operating regions.

### Maximum cooling

As cooling requirements increase,  $z_{V_{ev}}$  will open and reach  $z_{V_{ev}}^{max}$ . The valve position controller for  $V_{ev}$  will adjust  $T_h$  (and indirectly  $P_h$ ) such that the system reaches  $Q_{ev}^{max}$  (optimum). This is handled by the structure already designed.

## 3.4 Concluding remarks

Using the systematic procedure described in Chapter 2, we designed a PI(D)-based control structure for a generic CO<sub>2</sub>-refrigeration system, that follows the *priority list* of constraints and maintains *near-optimal* steady-state operation for three different operating modes. To this end, we applied three different constraint switching strategies (MV to MV constraint switching, CV to CV constraint switching and MV to CV constraint switching). We should point out that pairing on the low-pressure side could be different (e.g. controlling  $T_{cab}$  with  $\omega_1$ , and  $P_l$  with  $z_{V_{ev}}$ ). The final decision would consider system dynamics.



# Chapter 4

## Saturation of manipulated variables: classical advanced control structures versus MPC

This chapter is based on:

Reyes-Lúa, A., Zotică, C., and Skogestad, S. (2018d). Optimal Operation with Changing Active Constraint Regions using Classical Advanced Control. In *10th IFAC Symposium on Advanced Control of Chemical Processes (ADCHEM)*, Shenyang, China. IFAC-Papers OnLine

Reyes-Lúa, A., Zotică, C., Das, T., Krishnamoorthy, D., and Skogestad, S. (2018b). Changing between Active Constraint Regions for Optimal Operation: Classical Advanced Control versus Model Predictive Control. In *Proceedings of the 28th European Symposium on Computer Aided Process Engineering (ESCAPE)*, Graz, Austria. Computer-aided chemical engineering

Control structures must be properly designed and implemented to maintain optimality. The two options for the supervisory control layer are classical advanced control structures and Model Predictive Control (MPC). With "classical advanced control" we mean the control structures that are commonly used in industry for multivariable control. These have been in use for at least 75 years (Eckman, 1945; Fink, 1945), but surprisingly there is little literature published on how to design such structures in a systematic manner. In Chapter 2, we introduced a design procedure to assure optimal operation when active constraint changes occur.

In this chapter, we focus on input saturation, specifically when the *input saturation tuning rule* is not followed. We analyze a case study in which we control the temperature and the flow in a cooler with two degrees of freedom represented by two valves, one for each of the two streams. Either valve can saturate and make a constraint active, changing the set of active constraints and thus, forcing other constraints to be given-up.

For this case study, we first implement three alternative classical advanced control structures that handle input saturation (MV to CV switching). Then, we implement standard MPC. We show that optimal or *near-optimal* operation can be reached with both alternatives. We do a fair comparison of classical advanced control structures and MPC as candidates for the supervisory layer, and provide some guidelines to help steer the choice. Another comparison of split range control with MPC is presented in Section 6.5.3.

## 4.1 Introduction

As shown in Fig. 2.1, on a time-scale basis, the overall control problem of a process plant can be decomposed into different layers. The upper layers are explicitly related to slow time-scale economic optimization, which sends economic set points to the lower and faster control layer. The control layer is divided into supervisory layer and regulatory layer. The latter follows the setpoints given by the former and stabilizes the plant.

Most process are operated under a set of constraints, which can be operational limitations, quality specifications, or safety and environmental requirements. "Active constraints" are related to variables that should be kept at their limiting value to achieve optimality. These can be either manipulated variables (MVs) or controlled variables (CVs). The MVs correspond to the dynamic (physical) degrees of freedom used by the control system, and a typical MV constraint is the maximum opening of a valve. An example of CV constraint is the maximum concentration in a reactor.

Every process is subject to disturbances, such as changes in feed rate or product specification. It is the task of the supervisory or "advanced" control layer to maintain optimal operation despite disturbances. The supervisory control layer has three main tasks (Skogestad, 2012):

1. Switch between the set of CVs and control strategies when active constraint changes occur due to disturbances.
2. Supervise the regulatory layer, avoiding saturation of the MVs used for regulatory control.
3. Follow economic objectives by using the setpoints to the regulatory layer as MVs .

The supervisory control layer is commonly designed using classical advanced control structures with PID-controllers and simple blocks. Alternatively it can be designed with Model Predictive Control (MPC). The main advantage of MPC in terms of

economics is that it inherently handles constraints and represents a unified systematic procedure to control multivariable processes (Mayne, 2014). However, standard MPC may not handle changes in active constraints effectively, except by the indirect use of weights in the objective function, which are selected by trial and error. This scheme may not allow one to give up completely controlling a variable. Furthermore, there is no systematic procedure to find tuning parameters, and the tuning to achieve a certain level of performance is not unique (Allison and Isaksson, 1998; Forbes et al., 2015). To handle such cases, one must either introduce logic, slack variables with penalty functions or implement a two-stage MPC.

In many cases, optimal or *near*-optimal operation can be achieved using classical advanced control structures in the supervisory layer. Allison and Isaksson (1998) compared valve position control with a generalized predictive control formulation of MPC for two systems with two available MVs for one CV. In their work, they concluded that tuning is key to achieve the desired performance. In this chapter, we evaluate different strategies to handle input saturation. In Section 4.2 we give an overview of active constraint switches using advanced control structures and MPC. We introduce the case study in Section 4.3. In Section 4.4 we evaluate three different control structures that handle input saturation; namely,

- split range control with selector
- valve position control with selector
- two controllers with different setpoints and a selector

A standard implementation of MPC is described in Section 4.5. We then compare the structure with the best performance with standard MPC in Section 4.6. We give our concluding remarks in Section 4.7.

## 4.2 Optimal control in the presence of active constraint changes

Regardless of whether we choose advanced control structures or MPC, we should identify and prioritize all constraints to systematically design the supervisory control layer. Prioritization of constraints has been implemented in a few industrial MPC applications (Qin and Badgwell, 2003). In Chapter 2 we proposed a guideline to generate a priority list of constraints that can be used also for advanced control structures. Under this scheme, the constraints with the lowest priority should be the first given-up when it is not feasible to fulfill all constraints. This way, controlling a high-priority constraint will never be sacrificed in order to fulfill a low-priority constraint.

It is useful to visualize how disturbances may cause new constraints to become active. In some cases, we can generate a plot showing the active constraint regions (optimal operation) as a function of variations in important disturbances by solving a series of optimization problems. This may be very time consuming and, in some cases, difficult due to the lack of an appropriate model. Moreover, it can also be difficult to visualize for more than two variables. Alternatively, we can use process knowledge and engineering insight to minimize the need for numerical calculations (Jacobsen and Skogestad, 2011). This information is useful regardless of the type of controller used in the supervisory layer.

#### 4.2.1 Advanced control structures in the supervisory layer

Using advanced control structures requires a choice of pairings, which can become challenging with changing active constraints. When implementing advanced control structures, in Chapter 2 we propose to start designing the control system for the nominal point, with few active constraints and with most of the priorities satisfied. Then, to minimize the need for reassignment of pairings when there are changes in active constraints, we should pair MVs with CVs according to the *input saturation pairing rule* (Minasidis et al., 2015): an important controlled variable (CV) (which cannot be given up) should be paired with a manipulated variables (MV) that is not likely to saturate.

When a disturbance occurs and the process starts operating in a different active constraint region, two types of constraints might be reached<sup>1</sup>:

- MV constraint: we must give up controlling the corresponding CV. If the *input saturation pairing rule* is followed, this MV is paired with a low priority CV, which can be given-up. If it is not possible to follow the *input saturation pairing rule*, the high priority CV must be reassigned to an MV which is controlling a low priority CV. This requires the use of an MV to MV switching strategy (e.g. split range control or valve position control) combined with a selector block.
- CV constraint: we should give up controlling a CV with a lower priority. We can do this using a *min/max* selector.

#### 4.2.2 Model predictive control in the supervisory layer

MPC uses an explicit process model to predict the future response of the plant and, by computing a sequence of future MV adjustments, optimizes the plant behavior. The first input of the sequence is applied to the plant, and the entire calculation is repeated at every sampling time (Qin and Badgwell, 2003).

---

<sup>1</sup>This section is a simplified summary of Section 2.2.5.

The main challenge when using MPC is that expertise and a good model are required. This is either difficult to have ready at startup, or the modeling effort is too expensive. To achieve a truly optimal operation, the model would need to be perfect, and all the measurements would need to be available and reliable, which is unrealistic from a practical point of view. There are methods to circumvent this, but there is no universal solution and this analysis is out of the scope of this work.

When an application lacks degrees of freedom to meet all control specifications, standard textbook MPC does not handle changes in active constraints effectively. The standard approach is to use weights in the objective function to assign the priorities. Having weights in the objectives function implies a trade-off between the control objectives. An optimal selection of weights can assure that a CV is completely given-up, or that the solution will lie at the constraint. However, there is no systematic way of choosing the weights. As there are no tuning rules for MPC, this has to be done by trial and error.

An alternative approach consists of implementing a two-stage MPC<sup>2</sup>, in which we first generate a *priority list* by ranking the constraints. In the first stage, we solve a sequence of local steady-state optimization problems, each time adding a new constraint, following the *priority list*. This stage provides information regarding feasibility of the control objectives. In the second stage, we use the gathered information to formulate the dynamic optimization problem for the MPC. This way, we assure satisfying high priority constraints over lower priority constraints (Qin and Badgwell, 2003; Strand and Sagli, 2004; Aske et al., 2005).

### 4.3 Case Study: Optimal Control of a Cooler

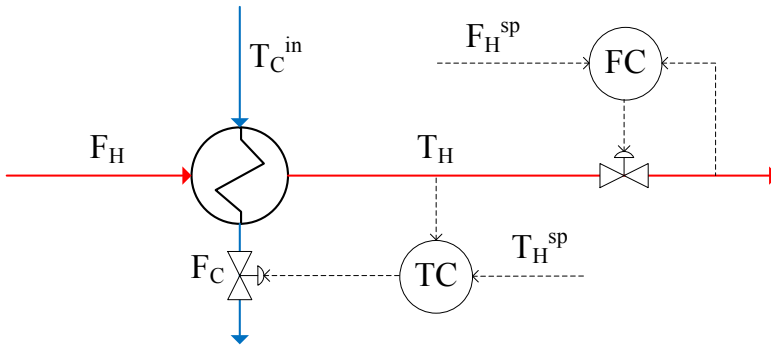
We consider the system in Fig. 4.1. The heat exchanger is a countercurrent cooler, represented by the dynamic lumped model in Appendix C. The main control objective of this case study is to keep the outlet temperature in the hot stream at its set point ( $T_H = T_H^{sp}$ ) by using cooling water ( $F_C$ ). Additionally, we would like to set the throughput of the hot stream ( $F_H = F_H^{sp}$ ), ideally at  $F_H^{sp} = F_H^{max}$ . The process has two MVs to achieve these objectives,

- $F_C$ : cooling water
- $F_H$ : throughput of hot stream

The main disturbance is the cooling water temperature ( $T_C^{in}$ ).

---

<sup>2</sup>The term "multi-stage MPC" can also be applied to the use of a "moving scenario" tree to account for uncertainties (Lucia and Engell, 2012; Lucia et al., 2013). In this work, we do not consider uncertainties.

Figure 4.1: Cooler with temperature and flow control<sup>3</sup>.

### 4.3.1 List of Priorities

Table 4.1 shows the constraints for the studied system.

Table 4.1: Constraints for the cooler system.

MV constraints	CV constraints
$F_C \leq F_C^{max}$	$T_H = T_H^{sp}$
$F_H \leq F_H^{max}$	$F_H = F_H^{sp}$

We now define the priority list of constraints according to Section 2.2.2. The physical MV constraints define the feasibility region. As they must always be met, these should be placed at the highest priority.  $T_H = T_H^{sp}$ , which is the control objective, is in priority level 2. Finally, we have the desired throughput at the lowest priority. In this case, there are no critical CV inequality constraints or self optimizing variables. Having  $F_H = F_H^{sp}$  at the lowest priority means that we can accept  $F_H \neq F_H^{sp}$  in order to achieve  $T_H = T_H^{sp}$ .

(P1) Physical MV inequality constraints:

- $F_H \leq F_H^{max}$
- $F_C \leq F_C^{max}$

(P2) CV constraints:

- $T_H = T_H^{sp}$

(P3) Desired throughput:

- $F_H = F_H^{sp}$

---

<sup>3</sup>Note that the flow controller is not shown in the following figures.



### 4.3.2 Active Constraint Regions

There are two MVs, two MV inequality constraints, and two CV equality constraints. As  $T_H = T_H^{sp}$  must be controlled always, we have one remaining degree of freedom and three potential constraints. This results in three possible active constraint regions, which are shown as a function of  $F_H$  and  $T_C^{in}$  (disturbance) in Fig. 4.2:

- Region 1:  $F_H = F_H^{sp} < F_H^{max}$
- Region 2:  $F_H = F_H^{max}$
- Region 3:  $F_C = F_C^{max}$

In all regions  $T_H = T_H^{sp}$ . Note that in region 1, none of the two inequality constraints are reached, and it is possible to keep  $T_H = T_H^{sp}$  and  $F_H = F_H^{sp}$  using both available MVs. In regions 2 and 3,  $F_H = F_H^{sp}$  must be given up. In region 2,  $F_C$  must be manipulated in order to keep  $T_H = T_H^{sp}$ , while in region 3,  $F_C = F_C^{max}$  and  $F_H$  needs to be reduced below its maximum in order to keep  $T_H = T_H^{sp}$ .

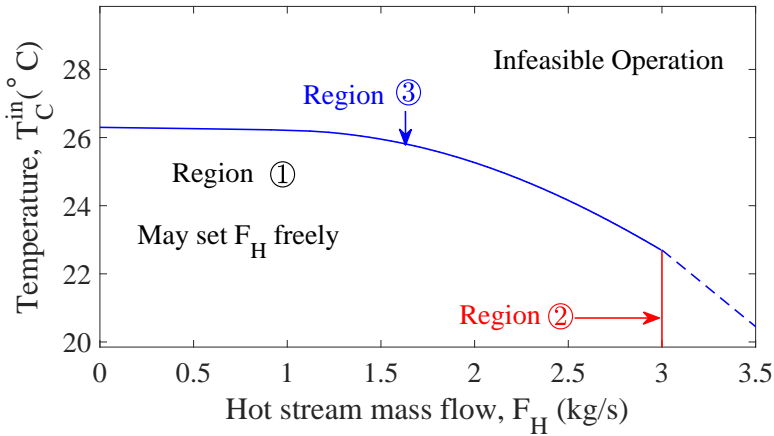


Figure 4.2: Active constraint regions for the studied cooler.

## 4.4 Evaluation of classical advanced control structures for input saturation

When using PID control for this system, the obvious control strategy is to use the cooling water ( $F_C$ ) to control  $T_H$ , and use a flow controller for  $F_H$ , as shown in Fig. 4.1. If the valve controlling  $F_H$  saturates, no logic is needed, except that the controller requires anti-windup.  $F_C$  may saturate for a high throughput ( $F_H$ ), as it is required that  $T_H$  is always controlled, reconfiguration may be needed. Here, we evaluate three control strategies that handle this case:

1. SRC: split range control
2. VPC: valve position control
3. two controllers ( $TC$  and  $TC2$ ) with different setpoints.

The performance of each option is tested for rejection of disturbances in  $T_C^{in}$  of  $+2^\circ C$  at  $t = 200 s$ , and an additional  $+4^\circ C$  at  $t = 2000 s$ . It should be noted that the three evaluated structures include a *min* selector for CV selection; that is, giving up controlling  $F_H = F_H^{sp}$ . The tuning procedure is described in Appendix C.2.

#### 4.4.1 Split range control (SRC)

Fig. 4.3 shows the implementation of SRC with a *min* selector. In Fig. 4.4 we observe that once  $F_C = F_C^{max}$ ,  $F_H$  is used as MV to control  $T_H$ , reaching optimality. The advantage of this structure is that there is only one temperature controller.

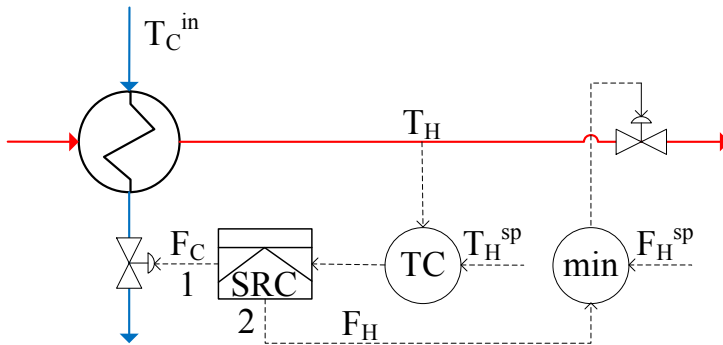


Figure 4.3: Split range control structure for cooler.

#### 4.4.2 Valve position control (VPC)

With this alternative,  $T_H$  is always controlled using  $F_C$ , while the VPC block takes the cooling water valve to 95% opening by regulating  $F_H$ , as shown in Fig. 4.5. However, because of the *min* selector, the VPC block only becomes active when  $F_C$  exceeds 95% of its maximum value. The advantage of this structure is that it always uses the same controller for  $T_H$  ( $TC$ ), avoiding any change of dynamics.

The performance of this case is tested for the same disturbances in  $T_C^{in}$  as in the SRC case. Fig. 4.5 shows a more oscillating response of VPC, compared to SRC. VPC is tightly tuned ( $\tau_c = \theta$ ) which results in an aggressive behavior and oscillations. To remove the oscillations for this example,  $\tau_c$  for VPC should be increased to at least  $30\theta$ , which in turn would result in an extremely slow response, with much poorer performance.

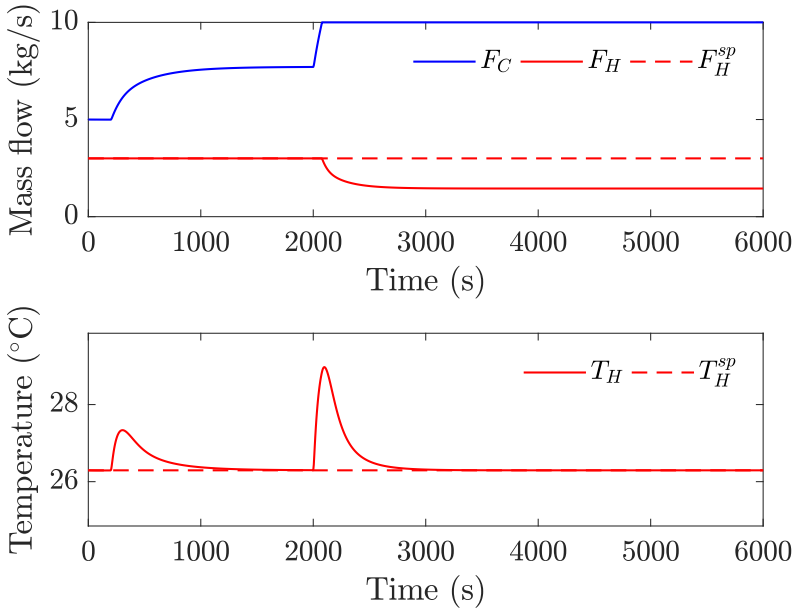


Figure 4.4: Disturbance rejection simulation results with split range control for cooler (Fig. 4.3).

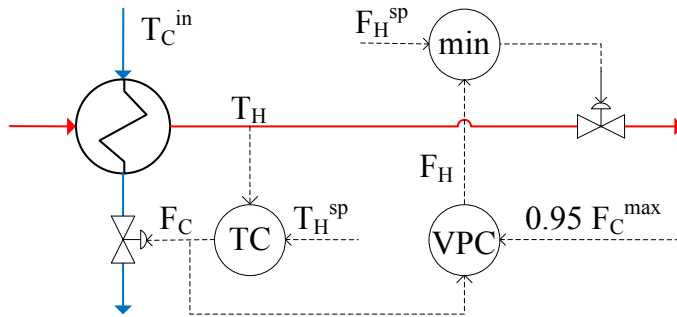


Figure 4.5: Valve positioning control (VPC) structure for cooler.

Anti-windup is required on the VPC block because a *min* selector is implemented (so the output of the VPC is not always applied). Strictly,  $F_H$  should be manipulated by the VPC block once  $F_C$  reaches  $0.95 F_C^{max}$ . However, when anti-windup with back-calculation is implemented, it starts acting before the setpoint for  $F_C$  is reached. This explains why the temperature overshoot for VPC (Fig. 4.6) is slightly smaller than for SRC (Fig. 4.4).

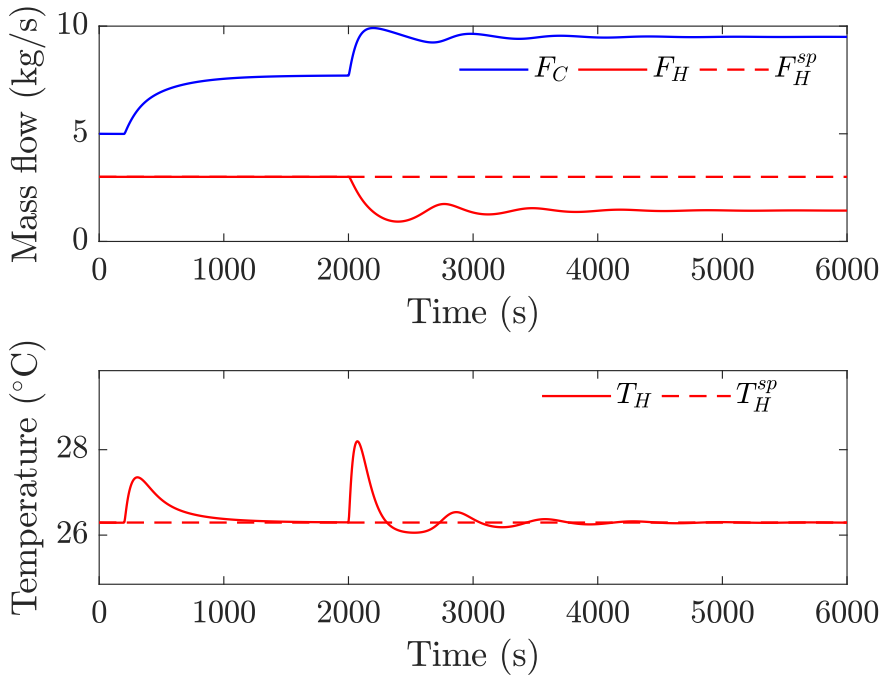


Figure 4.6: Disturbance rejection simulation results with VPC for cooler (Fig. 4.5).

#### 4.4.3 Different controllers with different setpoints

A third possible control structure is to implement two temperature controllers with different set points, as shown in Fig. 4.7. Different set points are needed to avoid interactions between the two controllers.  $TC$  uses  $F_C$  as MV, and has a set point  $T_H^{sp} = 26.3^\circ C$ . On the other hand,  $TC2$ , with set point  $T_H^{sp} + \Delta T_H^{sp} = 27.3^\circ C$ , uses  $F_H$  as MV. This solution is not optimal for this case study. Unlike SRC and VPC, when  $F_C$  is saturated,  $T_H \neq T_H^{sp}$  at steady state, as observed in Fig. 4.8.

Based on Figs. 4.4, 4.6, and 4.8, we observe that split range control is the only strategy that achieves optimal operation. In the presented example, split range control also shows the best dynamic performance. Thus, we will compare split range control with MPC.

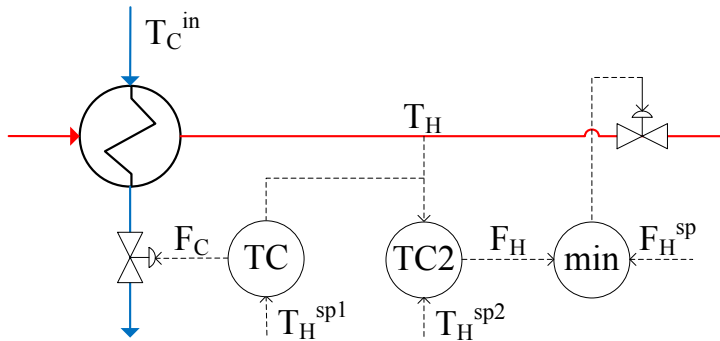


Figure 4.7: Control structure using two controllers for cooler.

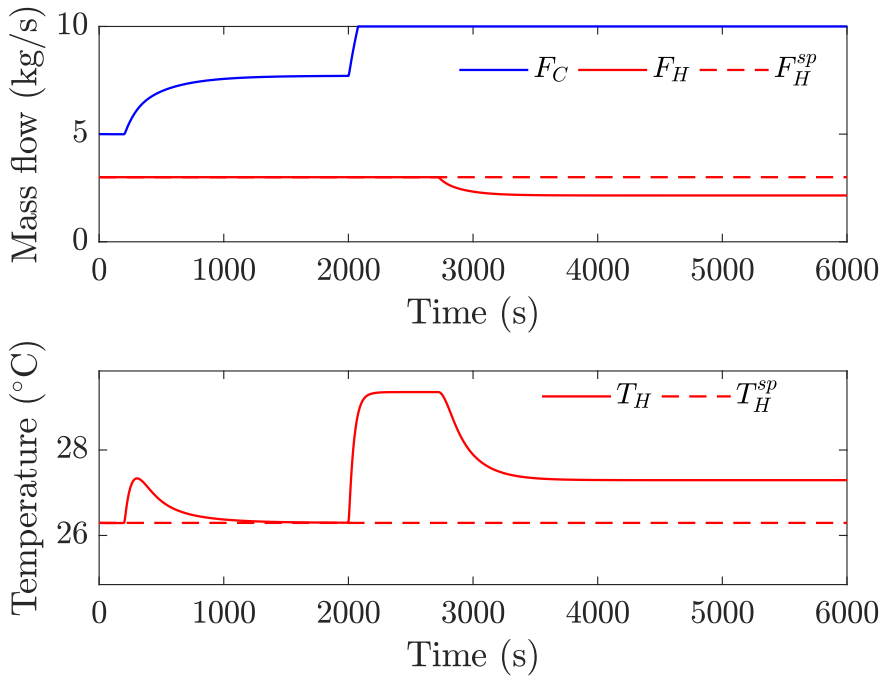


Figure 4.8: Disturbance rejection simulation results using two controllers for cooler (Fig. 4.7).

## 4.5 Model predictive control (MPC) for cooler case study

To implement MPC in this case study, the optimal control problem is discretized into a finite dimensional optimization problem divided into  $N = 40$  control intervals. We use a third order direct collocation scheme for a polynomial approximation of the system dynamics for each time interval.

The dynamic optimization problem is setup in CasADi (Andersson, 2013), which is an algorithmic differentiation tool. According to Eq. C.1, the dynamic model is non-linear. The resulting NLP problem is thus solved using IPOPT (Wächter and Biegler, 2005). The prediction horizon is set to 400 s with a sampling time of  $\Delta t = 10$  s. We assume we have full state feedback and the disturbance,  $T_c^{in}$ , is measured.

Here we chose to implement the standard NMPC formulation given by Eq. 4.1, and to assign different weights for the two control objectives. A high weight is assigned to the high priority CV ( $T_H$ ) and a low weight is assigned to the low priority CV ( $F_H$ ). The values  $\omega_1 = 3$  and  $\omega_2 = 0.1$  are used. These were found by trial and error. In addition, the MVs are restricted to a rate of change of 10% of  $F_H^{max}$  and  $F_C^{max}$  respectively.

$$\begin{aligned}
\min \quad & \sum_{k=1}^N \left( \omega_1 \| (T_{H_k} - T_H^{sp}) \|^2 + \omega_2 \| (F_{H_k}^{max} - F_{H_k}) \|^2 \right) \\
\text{s.t.} \quad & \left. \begin{aligned} & \mathbf{T}_{k,i} = f(T_{H_k,i}, T_{H_k,i-1}, T_{C_k,i}, T_{C_k,i+1}, F_{H_k}, F_{C_k}) \\ & 0 \leq F_{H_k} \leq F_H^{max} \\ & 0 \leq F_{C_k} \leq F_C^{max} \end{aligned} \right\} \quad \forall k \in \{1, \dots, N\} \\
& \left. \begin{aligned} & 0 \leq \Delta F_{H_k} \leq 0.1 F_H^{max} \\ & 0 \leq \Delta F_{C_k} \leq 0.1 F_C^{max} \end{aligned} \right\} \quad \forall k \in \{1, \dots, N-1\}
\end{aligned} \tag{4.1}$$

where  $\Delta F_k = F_k - F_{k-1}, \forall k \in \{1, \dots, N-1\}$ . For  $k = 1$ ,  $F_{k-1}$  represents the flow at the nominal operation point.

In this formulation:

- there is no terminal cost because we want that the valve for the cold flow is saturated at the end, while the terminal cost prevents this resulting in a steady-state offset.
- as mentioned, the disturbance is passed from the plant to the NMPC to avoid plant-model mismatch. This could be avoided with a linear MPC which has set point feedback.
- all states are scaled in the model equation with respect to the maximum temperature, which is  $T_H$  at the inlet, such that all values are [0:1].
- there are constraints for the input rate of change.

## 4.6 Comparison of split range control with MPC

Fig. 4.9 shows the simulation results for the case study.  $T_H^{sp}$  is  $26.3^\circ C$ . MPC and SRC structures are tested for the same step disturbances in  $T_C^{in}$ :  $+2^\circ C$  at  $t = 10\text{ s}$ , and an additional  $+4^\circ C$  at  $t = 1000\text{ s}$ .

Both MPC and SRC follow the priority list and reach optimal operation at steady state. Once  $F_C = F_C^{max}$ , the control structure gives-up controlling  $F_H = F_H^{max}$ , and  $F_H$  is used as MV to maintain  $T_H = T_H^{sp}$ .

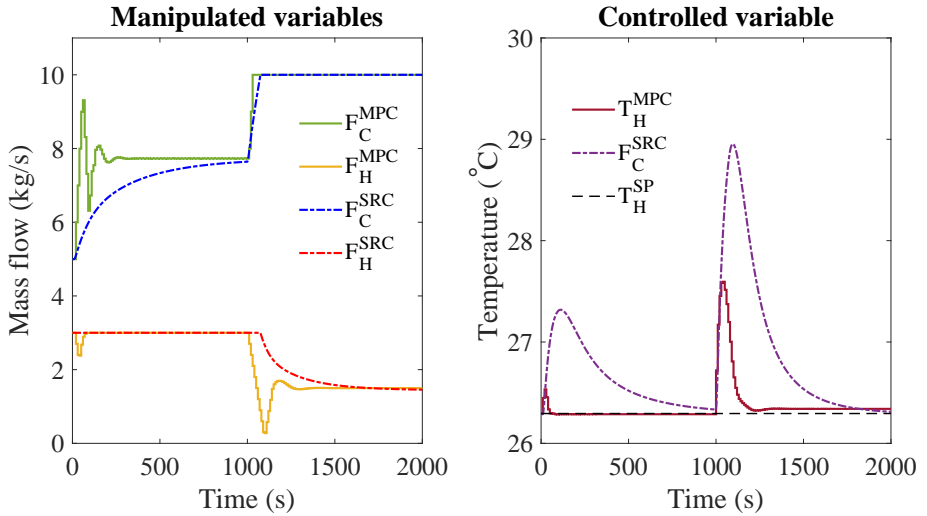


Figure 4.9: Simulation results for MPC and SRC.

As it is seen in Fig 4.9, for a short transient time during the first disturbance in the MPC implementation,  $F_H \neq F_H^{max}$ . This could be improved by increasing  $\omega_2$  relative to  $\omega_1$ . This would however be at the expense of having a larger offset for  $T_H$  from  $T_H^{sp}$ , as its weight in the objective function would be smaller. Also, with the selected weights the response of MPC is more aggressive than the response using SRC.

## 4.7 Concluding remarks

We recommend to use priority lists as a tool for analyzing and designing the supervisory layer. Understanding the process is an important step to decide which controller should be implemented. In this case study, optimality is defined by achieving the maximum possible throughput while maintaining  $T_H$ , a high-priority CV, at its set-point. Thus, we need to handle input saturation when the MV controlling  $T_H$  saturates.

In this case study we compared three MV to CV switching strategies with classical advanced control structures. Considering the control objectives in this case, split range control (SRC) is the strategy that allows optimal steady-state operation because it keeps the MV controlling the high-priority CV at the constraint while achieving maximum throughput (using the whole range of the MV). Using either valve position control (VPC) or two controllers requires a back-off from optimality<sup>4</sup>.

We then compared the performance of split range control with MPC, and both follow the priority list and reach operation at steady-state. MPC uses the manipulated variables to achieve optimal operation by design, but it requires expertise and a model, which may be difficult to obtain. Compared to advanced control structures, MPC implementation requires more effort as the tuning of weights in the objective function is more challenging because it is done by trial and error. In this case, we used NMPC in which the disturbance is passed from the plant to the NMPC. Perhaps a fairer comparison with SRC would be with linear MPC without the measured disturbance as input.

Both, advanced control structures and MPC, have advantages and disadvantages. The designer of the control layer should be aware of these. In simple cases such as the presented case study, advanced control structures seem better fitted due to achieving optimality with much less model information and less implementation effort, as PID-based control structures do not require an explicit model and tuning can be done in a systematic way. In multivariable systems with more interactions, MPC should be considered as the most convenient alternative.

In the presented case study, a different MPC implementation or tuning may achieve better performance, especially on the input usage. In the same line, the performance of advanced control structures depends on the tuning of the controller. Specifically, for split range control, the performance depends on the design and tuning of the common controller. This is also discussed in Section 6.5.3, where we compare our proposed generalized split range control structure with MPC.

---

<sup>4</sup>This is also mentioned in Section 1.2.5



# Chapter 5

## Systematic design of split range controllers

This chapter is published as:

Reyes-Lúa, A., Zotică, C., Forsman, K., and Skogestad, S. (2019b). Systematic Design of Split Range Controllers. In *12th IFAC Symposium on Dynamics and Control of Process Systems, including Biosystems (DYCOPS)*, Florianópolis, Brazil. IFAC-Papers OnLine

Split range control is a common advanced control structure in the process industry. It is primarily used to extend the steady-state operating range by using more than one manipulated variable (MV). More generally, it is used to switch to another MV when the original MV saturates. We propose a systematic procedure to design a split range controller considering the (different) dynamic effects of each MV on the controlled variable, as well as (steady-state) economics<sup>1</sup>. We illustrate this procedure with a practical example.

### 5.1 Introduction

Classical advanced control uses several standard functions (blocks) to cover cases not handled by the simple single-input single-output feedback controllers. Some examples are: cascade control, feedforward control, decoupling, selectors, split range control and valve positioning control. Multivariable controllers, such as Model Predictive Control (MPC), represent an alternative for some of these applications. However, MPC requires an explicit dynamic model. Furthermore, standard MPC does not allow to give-up completely controlling a variable and there is no systematic tuning procedure for MPC (Forbes et al., 2015).

---

<sup>1</sup>The basic concepts of this procedure are summarized in Section 6.2.

This chapter focuses on split range control (SRC), which is used when there are two or more manipulated variables (MVs) associated with one controlled variable (CV). The most common use of split range control is to extend the steady-state range by switching to another MV when the primary MV saturates; for example, to switch to electric heating when the hot water saturates. Some other names that have been used for split range control are *dual control agent* (Eckman, 1945) and *valve sequencing* (Lipták, 1985). Although split range control has been used for more than 75 years (Eckman, 1945; Fink, 1945), there is no systematic procedure for the design of split range controllers, to the best of the authors' knowledge.

This chapter is organized as follows: in Section 5.2 we describe the split range control structure, while in Section 5.3, we describe how to get the desired controller gain for each MV by adjusting the slopes in the split range block. Section 5.4 proposes a systematic procedure for the design of split range control. We then implement this procedure in a case study in Section 5.5. In Section 5.6 we discuss about alternative control structures for split range control, and we make our final remarks in Section 5.7.

## 5.2 Classical split range control

Let the manipulated variables ( $MV_i$ ) be denoted by  $u_i$  and the controlled variable (CV) be denoted by  $y$ . As shown in the block diagram in Fig. 5.1, most applications have two MVs ( $u_1$  and  $u_2$ ) and one CV ( $y$ ). There is one single-input single-output controller (C) that calculates the internal signal ( $v$ ) to the split range block (SR). C is commonly a PI controller. The split range block splits  $v$  into the two MVs ( $u_1$  and  $u_2$ ).

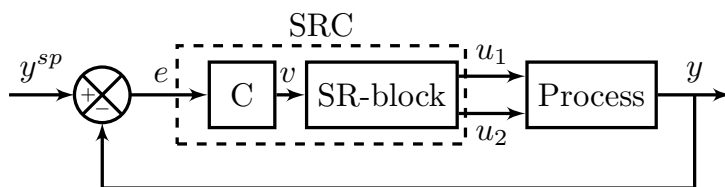


Figure 5.1: Block diagram for standard implementation of split range control with two MVs and one CV.

Fig. 5.2 depicts a typical split range block. When  $v$  is below the split value ( $v^*$ ),  $u_1$  is used to control the CV ( $y$ ), while  $u_2$  is saturated. At the split value,  $u_1$  becomes saturated, and the controller starts using  $u_2$  to control  $y$ .

The split value is located at the mid-point ( $v^* = 50\%$ ) in Fig. 5.2, but there is no reason to use this particular value. Instead,  $v^*$  should be used as a design parameter

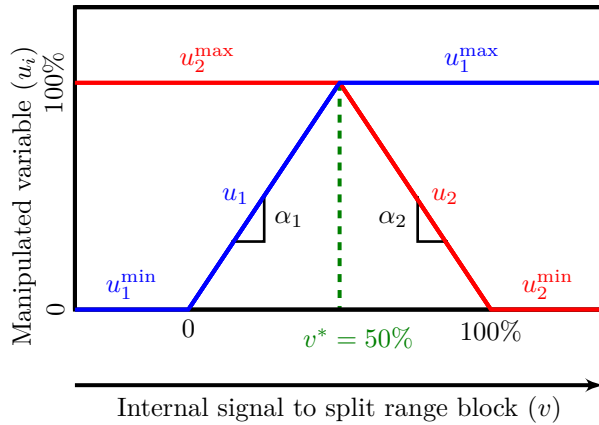


Figure 5.2: Split range block with extended  $v$ , giving the relationship between  $v$  and two MVs ( $u_1$  and  $u_2$ ). In general, the split value ( $v^*$ ) should not be fixed at 50%.

for the split range block to adjust the dynamic response (Lipták, 1985; Glemmestad, 1997; Hägglund, 1997; Alsop, 2016).

Fixing  $v^*$  at a given value (e.g. 50%) is related to a common misconception, also found in most textbooks (e.g. Stephanopoulos (1984); Marlin (2000); Bequette (2002); Seborg et al. (2003)). The misconception is that  $v$  is the “controller output”, and thereby the signal sent to the valves. However, the actual controller output are the signals  $u_i$  coming out of the split range block, whereas  $v$  is an internal signal in the controller with limited physical significance.

In Fig. 5.2, the MV signals ( $u_i$ ) (on the y-axis) are physically limited to be within the given range from 0% (e.g., fully closed valve) to 100% (e.g., fully open valve). The internal signal ( $v$ ) (on the x-axis) is also shown to be scaled in the range 0% to 100%, but here there are no physical limits and  $v$  can be outside the range 0% to 100%. This follows from the fact that the internal signal ( $v$ ) is in deviation variables, whereas the outputs from the split range block ( $u_i$ ) are physical variables. For example, when we are operating on the right side of Fig. 5.2, we have:

$$u_2 = u_2^{\max} + \alpha_2(v - v^*) \quad (5.1)$$

Let us try to explain why  $v^*$  is actually a design parameter. At a given operating point, the integral action in the controller will drive the physical MV<sub>2</sub> ( $u_2$ ) to a given steady-state value. From Eq. (5.1), this means that the difference  $v - v^*$  will have a given value. However, if we let  $v^*$  have another value, then  $u_2$  and  $(v - v^*)$  will remain the same, but the internal signal ( $v$ ) will change.

The slopes in the split range block ( $\alpha_i$ ) have physical significance as controller gain contributions for each MV $_i$  ( $u_i$ ). Considering the signs of the slopes, we can distinguish two main cases.

The first case is when the MVs have opposite effects (gains) on the CV. One of the earliest descriptions of this case is the use of split range control to maintain constant temperature by using steam when the surrounding temperature is low and cold water when it is high (Eckman, 1945; Fink, 1945).

The second case is when the MVs have the same effects, but one MV is preferred for economic reasons. For example, Fink (1945) considers the case with three MVs for temperature control of a reactor with an exothermic reaction: two for cooling and one for heating. In this example, one should first use cold water for cooling, and when the cold water valve cannot handle the heat load, one should also use the more expensive refrigerated water to maintain the reactor at the desired temperature.

### 5.3 Selection of slopes

In split range control, several MVs ( $u_i$ ) are calculated from the same internal signal ( $v$ ), but at a particular time, only one of them is being used to control the CV. However, each of the MVs ( $u_i$ ) has a different dynamic and static effect on the CV ( $y$ ), and this should be considered when designing the split range controller.

In some implementations in which the output of the split range block is in deviation variables (e.g. Bequette (2002)), this signal is modified by a multiplication factor to indirectly consider the different gains of the MVs in the process. However, as mentioned earlier, this is not necessary as we should instead adjust the slopes.

From Eq. (5.1) and Fig. 5.2 it is evident that the slopes in the split range block ( $\alpha_i$ ) correspond to the gains from the internal signal ( $v$ ) to the value of each MV $_i$  ( $u_i$ ). As a generalization of Eq. (5.1), the split range block can be represented as the linear function

$$u_i = u_{i,0} + \alpha_i v \quad \forall i \in \{1, \dots, N\} \quad (5.2)$$

where  $u_i$  is the value of each MV $_i$ ,  $v$  is the internal signal to the split range block,  $\alpha_i$  is the gain from  $v$  to  $u_i$  (the slopes in Fig. 5.2,  $\Delta u_i / \Delta v$ ),  $N$  is the number of MVs, and  $u_{i,0}$  is the bias. Note that Eq. (5.1) and (5.2) are identical, with  $u_{2,0} = u_2^{\max} - \alpha_2 v^*$ .

What value should be select for the slopes ( $\alpha_i$ )? As a starting point, it seems reasonable to select them to counteract the differences in the static loop gain ( $K_{p,i}$ ) for each MV $_i$  and to select  $|\alpha_i|$  proportional to  $1/|K_{p,i}|$ . However, this is too simplified, as one should also consider the dynamic response for each MV $_i$ .

Let the desired controller for  $MV_i$  be denoted  $C_i(s)$ . For example, it could be a PI controller with gain  $K_{C,i}$  and integral time  $\tau_{I,i}$ . This is the controller we would like to have if we were free to choose any controller.  $C_i(s)$  should be compared with the common controller  $C(s)$ , see Fig. 5.1, which could be a PI controller with gain  $K_C$  and integral time  $\tau_I$ .

Including the split range block (where we have the slopes  $\alpha_i$ ), we see that the actual controller in Fig. 5.1 for  $MV_i$  is  $\alpha_i C(s)$ . However, since we only have one free parameter,  $\alpha_i$ , it is not possible in general to make  $\alpha_i C(s)$  equal to the desired  $C_i(s)$ . The best we can do is to use  $\alpha_i$  to match the desired controller at the desired crossover (or bandwidth) frequency, which will be at frequency  $\omega_{c,i} = 1/\tau_{c,i}$ , where  $\tau_{c,i}$  is the desired closed-loop time constant for  $MV_i$ .

Consider a PI controller

$$C(s) = K_C \left( 1 + \frac{1}{\tau_I s} \right) \quad (5.3)$$

At frequency  $\omega_c = 1/\tau_c$ , we then find that the frequency response is given by  $C(j\omega_c) = K_C(1 - j\frac{\tau_c}{\tau_I})$ . From this, we find that  $C(j\omega_c) \approx K_C$  for  $\tau_c \ll \tau_I$ , and  $C(j\omega_c) \approx -j\frac{K_C}{\tau_I}\tau_c$  for  $\tau_c \gg \tau_I$ . We then have two main cases, also depicted in Fig. 5.3:

1. “Slow” (*integrating or close-to integrating*) process, where  $\tau_c \ll \tau_I$  (high frequency,  $\omega$ , in Fig. 5.3). The proportional gain ( $K_{C,i}$ ) is the most important controller parameter because  $C_i(j\omega_{c,i}) \approx K_{C,i}$ . We select the slopes ( $\alpha_i$ ), or equivalently the break points, to achieve:

$$K_{C,i} = \alpha_i K_C \quad \forall i \in \{1, \dots, N\} \quad (5.4)$$

Here  $K_{C,i}$  is the desired controller gain for each  $MV_i$  and  $K_C$  is the proportional gain in the common PI controller in Fig. 5.1.

2. “Fast” process, where  $\tau_c \gg \tau_I$  (low frequency,  $\omega$ , in Fig. 5.3). Here, the most important controller parameter is the integral gain ( $K_{I,i} = K_{C,i}/\tau_{I,i}$ ) because  $C_i(j\omega_{c,i}) \approx -j\frac{K_{C,i}}{\tau_{I,i}}\tau_{c,i}$ . Thus, for such processes instead of computing the slope ( $\alpha_i$ ) according to Eq. (5.4), we should compute it according to  $K_{I,i} = \alpha_i K_I$ , or equivalently:

$$\frac{K_{C,i}}{\tau_{I,i}} = \frac{\alpha_i K_C}{\tau_I} \quad \forall i \in \{1, \dots, N\} \quad (5.5)$$

Here  $K_{C,i}$  and  $\tau_{I,i}$  are the desired PI settings for the controller  $C_i(s)$  for  $MV_i$ , whereas  $K_C$  and  $\tau_I$  are the settings used in the common PI controller.

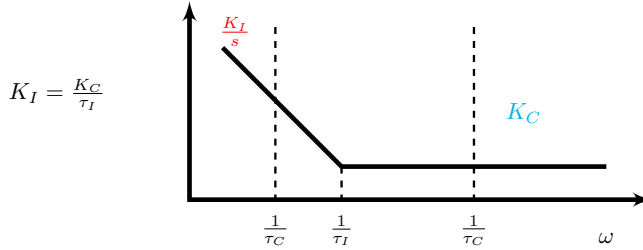


Figure 5.3: Significant controller parameters according to frequency.

### 5.3.1 Controller tunings

In Eq. (5.4) and (5.5),  $K_{C,i}$  and  $\tau_{I,i}$  are the desired PI settings for each MV $_i$ . One way to find good PI settings is to use the SIMC rules (Skogestad, 2003), in which we first identify a first-order plus time delay model<sup>2</sup>,

$$G_i(s) = \frac{K_{p,i}}{\tau_i s + 1} e^{-\theta_i s} \quad (5.6)$$

for each MV $_i$  and then select the desired closed loop time constant ( $\tau_{c,i}$ ) to calculate  $K_{C,i}$  and  $\tau_{I,i}$ :

$$K_{C,i} = \frac{\tau_i}{K_{p,i}(\tau_{c,i} + \theta_i)} \quad (5.7a)$$

$$\tau_{I,i} = \min\{\tau_i, 4(\tau_{c,i} + \theta_i)\} \quad (5.7b)$$

Note that, from Eq. (5.4) and (5.5), if  $K_C$  is positive, then  $\alpha_i$  has the same sign as  $K_{C,i}$ , which from Eq. (5.7a) has the same sign as the process gain  $K_{p,i}$ . We also note that selecting  $\tau_{I,i} = 4(\tau_{c,i} + \theta_i)$  in Eq. (5.7b) corresponds to a "slow" process (case 1) and selecting  $\tau_{I,i} = \tau_i$  corresponds to a "fast" process (case 2).

What value should we select for the integral time ( $\tau_I$ ) in the common controller? There is no simple answer to this. If one particular MV, let us say  $u_k$ , is used most of the time, then it is reasonable to select  $\tau_I = \tau_{I,k}$ . In other cases, one may select  $\tau_I$  as some average of the desired  $\tau_{I,i}$  for the individual loops. What value should one choose to be on the "safe" side with respect to stability? It depends on whether we are matching  $K_C$  or  $K_I$ . If we have a "slow" process and are matching  $K_C$  according to Eq. (5.4), then selecting a large value for  $\tau_I$  is safer. On the other hand, if we have a "fast" process and are matching  $K_I$  according to Eq. (5.5), then selecting a small value for  $\tau_I$  is safer.

<sup>2</sup> $K_{p,i}$  is the process gain,  $\tau_i$  is the open-loop time constant and  $\theta_i$  is the time delay.

### 5.3.2 Bias

The bias,  $u_{i,0}$  in Eq. (5.2), is usually constant. However, when using a split-range configuration in combination with a selector, the bias of the MV affected with the selector is variable. In these cases, the integral part of the PI-controller in the split range control structure will bring the CV to its set-point, even when using a constant bias. However, we should point out that updating the bias improves the dynamic response when changing the active MV.

## 5.4 A new procedure for designing the Split Range Block

Here, we propose a systematic procedure to design the split range block considering the different dynamics of each MV $_i$ , as discussed in Section 5.3.

For the first steps, we need to make some decisions:

- S1 Define the range for the internal signal from the controller to the split range block ( $v^{\min}, v^{\max}$ )<sup>3</sup>.
- S2 Find the minimum and maximum values for every MV ( $u_i^{\min}, u_i^{\max}$ ). Here, we typically normalize the MVs, such that  $u_i^{\min}$  and  $u_i^{\max}$  is the same for every MV (e.g. 0% – 100%).
- S3 Decide on the desired controller tunings for each individual MV $_i$ . For example, one may use the SIMC rules (Eq. (5.7)) to find the desired PI controller proportional gain ( $K_{C,i}$ ) and desired integral time ( $\tau_{I,i}$ ).
- S4 For PI control, choose the integral time ( $\tau_I$ ) for the common controller, as discussed in Section 5.3.1.
- S5 Choose the order for the MVs based on physical and economic arguments. In this step, it is useful to make a graphical representation of the split range block (as in Fig. 5.2). This is further explained in Section 5.4.1.

The remaining steps are purely algebraic:

- S6 From Fig. 5.2, we note that we must have:

$$v^{\max} - v^{\min} = \sum_{i=1}^N \frac{u_i^{\max} - u_i^{\min}}{|\alpha_i|} \quad (5.8)$$

---

<sup>3</sup>  $v$  is an internal signal, not the actual controller output, and it can be re-scaled freely. For example, the range can be -1 to 1 or 0% to 100%.

Use Eq. (5.8) together with Eq. (5.4) for a "slow" process or Eq.(5.5) for "fast" process to find the slopes ( $\alpha_i$ ) for each  $MV_i$  and the common controller gain  $K_C$ .

S7 Find the range of the internal signal covered by each  $MV_i$  ( $\Delta v_i$ ), and thereby the split values ( $v_i^*$ ), using Eq. (5.9):

$$\Delta v_i = v_i^* - v_{i-1}^* = \frac{u_i^{\max} - u_i^{\min}}{|\alpha_i|} \quad \forall i \in \{1, \dots, N\} \quad (5.9)$$

We should note that, as we have a common controller  $C(s)$ , anti-windup should only be activated when all the MVs are saturated. In Fig. 5.2, this would be at  $v < 0\%$  or  $v > 100\%$ .

#### 5.4.1 Ordering the use of MVs (Step S5)

The order of use of the MVs should be defined considering the effect on the process as well as economic aspects.

We suggest to order the MVs in the split range block according to the following procedure:

S5.1 Define the desired or most economical operating point for every  $MV_i$  (e.g. fully closed or fully open valve).

S5.2 Consider the effect of the available MVs on the CV. Then, group the MVs into:

- (a) MVs for which the value of the CV *increases* when we move away from the desired operating condition.
- (b) MVs for which the value of the CV *decreases* when we move away from the desired operating condition.

S5.3 Within each group, (a) and (b), order the MVs according to which one should be used first (less expensive) to which should be used last (more expensive). The MVs that should be used first will be located closest to the point defined in S5.1.

**Example:** Consider temperature control for a room. The CV is the room temperature ( $y = T$ ) and the main disturbance is the ambient (outdoor) temperature ( $d = T^{amb}$ ). The available MVs that affect room temperature ( $y = T$ ) are: heating ( $u_3$ ), cooling ( $u_2$ ) and ventilation ( $u_1$  in summer and  $u_4$  in winter). To order the MVs we note that the desired operating point is to use no heating or cooling (to save money) and to have maximum ventilation (to have the best air quality).



We now follow the procedure to order the use of the MVs:

- S5.1 The desired operating point is when the ambient temperature ( $d = T^{amb}$ ) is equal to the desired room temperature ( $T^{ref} = T^{amb}$ ). At this point, heating and cooling are off, and the ventilation flow is at its maximum, to maintain the best air quality. For example, with a set point  $T^{ref} = 22^\circ C$  for the indoor temperature, the desired operating point is when the outdoor temperature happens to be  $T^{amb} = 22^\circ C$ .
- S5.2 If  $T^{amb}$  increases, we need to cool the room to maintain the desired room temperature. On the other side, if  $T^{amb}$  decreases, we need to heat the room. Then, we can group the MVs:
- (a) MVs that *increase* the room temperature ( $y$ ). These are the MVs that we would use when  $T^{amb} < T^{ref}$ .
    - Heating ( $u_3$ )
    - Ventilation ( $u_4$ ). Note that in the winter, reducing the ventilation will *increase* the room temperature ( $y$ )
  - (b) MVs that *decrease* the room temperature ( $y$ ). These are the MVs that we would use when  $T^{amb} > T^{ref}$ .
    - Cooling ( $u_2$ )
    - Ventilation ( $u_1$ ). Note that in the summer, reducing the ventilation will *decrease* the room temperature ( $y$ )
- S5.3 (a) In the summer, we first use cooling ( $u_2$ ) and only when it reaches its maximum we start reducing the ventilation ( $u_1$ ).
- (b) In the winter, we first use heating ( $u_3$ ) and only when it reaches its maximum we start reducing the ventilation ( $u_4$ ).

Fig. 5.4 shows the resulting split range block.

#### 5.4.2 Systems with "fast" and "slow" MVs

Most process applications are "slow processes", and therefore,  $K_C$  and the slopes  $\alpha_i$  will be found in step S6 by solving a system of equations using Eq. (5.8) together with Eq. (5.4). It is possible that in certain applications, the system behaves as a "slow" process when using some MVs and it behaves as a "fast" process when using other MVs. In those cases, the system of equations in step S6 to find  $K_C$  and  $\alpha_i$  can be defined using correspondingly, Eq. (5.4) for the "slow" process, and Eq. (5.5) for the "fast" process. As we are selecting  $\tau_I$  (in Eq. (5.5)) for the common controller in step S5, the system of equations remains of size  $N + 1$  and can be solved.

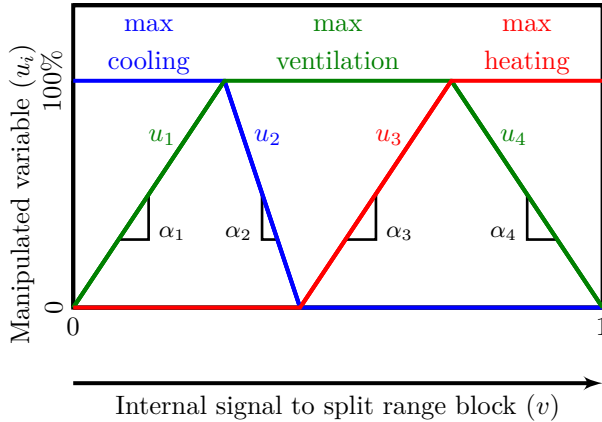


Figure 5.4: Split range block for room temperature control with heating ( $u_4$ ), cooling ( $u_2$ ) and ventilation flow ( $u_1$ ,  $u_4$ ) as MVs.

However, the resulting controller will be a compromise, and a more flexible control structure, as the one described in chapter 6 should be implemented.

## 5.5 Case study

In this section we show a simulation example of a similar room heating process, but in this case ventilation is not available as an MV.

### 5.5.1 Description of the system

We consider a room with two sources of cooling and two sources of heating:

- AC: air conditioning
- CW: cooling water
- HW: hot water (district heating)
- EH: electric heating.

The main disturbance is ambient temperature ( $T^{amb}$ ) and the nominal ambient temperature is  $T_0^{amb} = 18^\circ C$ . This will be chosen as the nominal room temperature  $T = 18^\circ C$ .

The control objective is to keep the room temperature at  $T = T^{ref}$ . Fig. 5.5 shows the block diagram for this process, using one PI controller and a split range block.

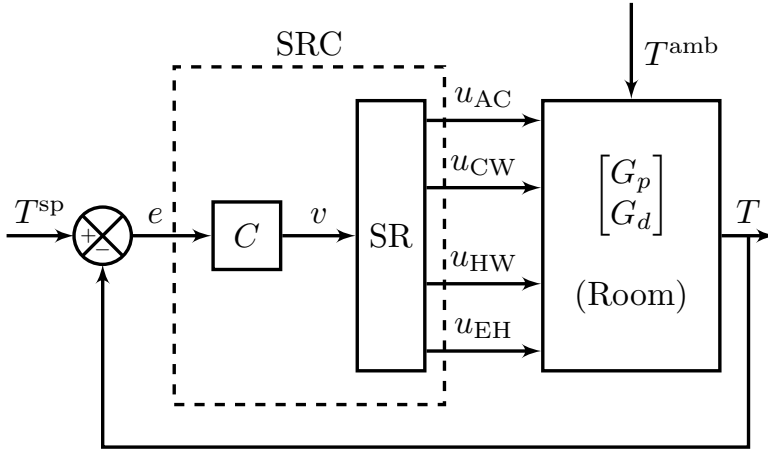


Figure 5.5: Block diagram of split range control for controlling room temperature.

We model the room as a linear system:

$$T(s) = G_p(s) u(s) + G_d(s) d(s) \quad (5.10)$$

where:

$$u = [u_{AC} \ u_{CW} \ u_{HW} \ u_{EH}]^T$$

$$G_p(s) = [G_{AC}(s) \ G_{CW}(s) \ G_{HW}(s) \ G_{EH}(s)]$$

Table 5.1 shows the gains ( $K_{p,i}$ ), time constants ( $\tau_i$ ), and time delays ( $\theta_i$ ) for  $G_p(s)$ .

Table 5.1: Parameters for  $G_{p,i}(s)$  from  $u_i$  to  $T$ .

$G_{p,i}$	$K_{p,i}$	$\tau_i$ [min]	$\theta_i$ [min]
$G_{AC}$	-5	8	2
$G_{CW}$	-10	15	3
$G_{HW}$	12	10	3
$G_{EH}$	8	5	1

The disturbance transfer function from  $T^{amb}$  to the room temperature ( $T$ ) is:

$$G_d(s) = \frac{1}{15s + 1} e^{-6s} \quad (5.12)$$

### 5.5.2 Design of the split range controller

We now follow the procedure in Section 5.4 and design the split range controller.

#### Step S1

The range of the internal signal to the split range block is defined as  $v^{\min} = 0$ ,  $v^{\max} = 1$ ,  $v^{\text{tot}} = 1$ .

#### Step S2

The MVs are scaled such that for every MV<sub>*i*</sub>:  $u_i^{\max} = 1$  and  $u_i^{\min} = 0$ .

#### Step S3

We have the required information to use the SIMC rules, and the PI controller tunings for each MV<sub>*i*</sub> are shown in Table 5.2. Here, we choose tight control ( $\tau_c = \theta_i$ ) for AC and HW, and  $\tau_c > \theta_i$  for CW and EH.

Table 5.2: Tuning parameters for each MV.

$u_i$	$\tau_{c,i}[\text{min}]$	$K_{C,i}$	$\tau_{I,i}[\text{min}]$
$u_{AC}$	2	-0.4000	8
$u_{CW}$	4	-0.2143	15
$u_{HW}$	3	0.1389	10
$u_{EH}$	3	0.1563	5

#### Step S4

We choose  $\tau_I$  for the common PI controller. This is a "slow" process. To be "safe", we might want to use the largest value for  $\tau_{I,i}$  (15 min), but we will use 9.5 min, which is a compromise among all  $\tau_{I,i}$  values.

#### Step S5

The next step is to order the use of the MVs.

- S5.1 The most economical operating point is when  $T^{\text{amb}} = T^{\text{ref}}$ , and we can have all MVs fully closed.
- S5.2 To maintain  $T = T^{\text{ref}}$ , we need to cool the room if  $T^{\text{amb}}$  increases, and to heat the room if  $T^{\text{amb}}$  decreases. With this in mind, we can group the MVs according to their effect on the room temperature. If  $T^{\text{amb}} > T^{\text{ref}}$ , we can use either CW or AC. Likewise, if  $T^{\text{amb}} < T^{\text{ref}}$ , we can use either HW or EH.

S5.3 Finally, we order the use of the MVs. As CW is less expensive than AC, we prioritize the use of CW over AC for decreasing room temperature. This locates CW closest the point where all the MVs are fully closed, and AC further away from this point. Likewise, we prioritize the use of HW over EH. Therefore, as shown in Fig. 5.6, the MV sequence in the resulting split range block is:  $u_1 = \text{AC}$ ,  $u_2 = \text{CW}$ ,  $u_3 = \text{HW}$  and  $u_4 = \text{EH}$ .

### Step S6

We can now proceed to the algebraic steps of the procedure and calculate  $K_C$  and  $\alpha_i$  by solving Eq. (5.4) together with Eq. (5.8). We find  $K_C = 0.0482$  and the values for  $\alpha_i$  reported in Table 5.3.

In this case  $K_C$  is positive. We can observe that both for AC ( $u_1$ ) and CW ( $u_2$ ),  $\alpha_i < 0$  (both decrease room temperature), while for HW ( $u_3$ ) and EH ( $u_4$ ),  $\alpha_i > 0$  (both increase room temperature). This corresponds to the expected physical behavior of these MVs.

### Step S7

Using the calculated values for  $\alpha_i$ , we can find  $\Delta v_i$  from Eq. (5.9). Then, the bias in Eq. (5.2), is:

$$\begin{aligned} u_{\text{AC},0} &= u_{\text{AC}}^{\max} \\ u_{\text{CW},0} &= u_{\text{CW}}^{\max} - (\alpha_{\text{CW}}) (\Delta v_{\text{AC}}) \\ u_{\text{HW},0} &= u_{\text{HW}}^{\min} - (\alpha_{\text{HW}}) (\Delta v_{\text{AC}} + \Delta v_{\text{CW}}) \\ u_{\text{EH},0} &= u_{\text{EH}}^{\max} - (\alpha_{\text{EH}}) (v^{\text{tot}}) \end{aligned}$$

Table 5.3 summarizes the information that describes the split range block for this system, and the final split range block is shown in Fig. 5.6.

Table 5.3: Values for  $\alpha_i$ ,  $\Delta v_i$  and  $u_{i,0}$ .

	AC	CW	HW	EH
$\alpha_i$	-8.3067	-4.4500	2.8843	3.2448
$\Delta v_i$	0.1204	0.2247	0.3467	0.3082
$u_{i,0}$	1.000	1.5357	-0.9954	-2.2448

## 5.5.3 Simulations

### Changes in $T^{\text{ref}}$

Fig. 5.7 shows the closed-loop response for changes in temperature set-point ( $T^{\text{ref}}$ ).

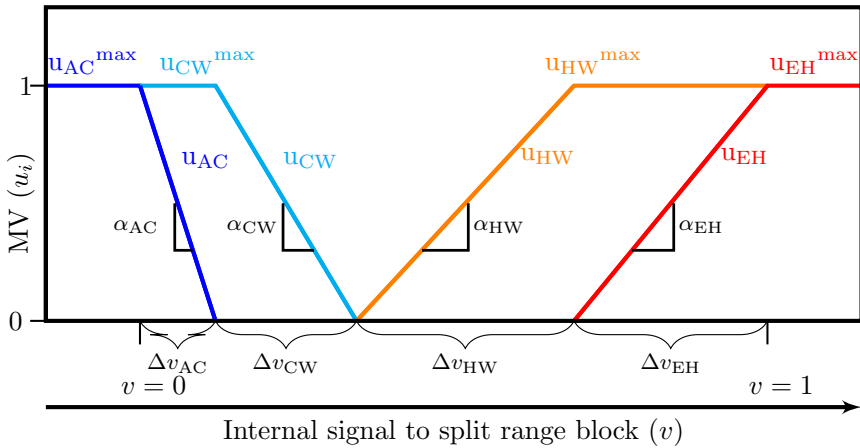


Figure 5.6: Standard split range block for room temperature control with air conditioning (AC), cooling water (CW), hot water (HW), and electric heating (EH).

In the beginning,  $T = T^{ref} = 18^\circ\text{C}$ . At  $t = 10\text{ min}$ , we increase  $T^{ref} + 5^\circ\text{C}$ , from  $18^\circ\text{C}$  to  $23^\circ\text{C}$ . This is easily achieved using hot water (HW). At  $t = 60\text{ min}$ , we further increase  $T^{ref} + 8^\circ\text{C}$ , to  $31^\circ\text{C}$ , and when HW becomes saturated at its maximum value, electric heating (EH) takes over to bring  $T$  to its desired set-point. When  $T^{ref}$  is decreased to  $16^\circ\text{C}$  at  $t = 110\text{ min}$ , both cooling options (CW and AC) saturate initially and anti-windup is used for a short period. We should note that  $\Delta T^{ref} = -15^\circ\text{C}$ , which is large. The AC is used only for a dynamically because at steady state it is sufficient to use cooling water (CW). Finally, when  $T^{ref}$  is decreased by  $-9^\circ\text{C}$ , to  $7^\circ\text{C}$ , CW saturates at its maximum value and we need to use the AC.

### Disturbances in $T^{amb}$

The performance of this implementation is also tested for rejection of disturbances in  $T^{amb}$  of  $+2^\circ\text{C}$  at  $t = 10\text{ min}$ ,  $+10^\circ\text{C}$  at  $t = 60\text{ min}$ ,  $-13^\circ\text{C}$  at  $t = 110\text{ min}$ , and an additional  $-15^\circ\text{C}$  at  $t = 160\text{ min}$ .

Fig. 5.8 shows the closed-loop response. The behavior is similar to the one observed for changes in set-point. At first, CW suffices to maintain  $T = T^{ref}$ , but when  $T^{amb} = 30^\circ\text{C}$ , CW reaches its maximum value and it is necessary to use the AC. Similarly, when  $T^{amb} < T_0^{amb}$ , it is initially enough to use HW, but when  $T^{amb}$  decreases considerably, HW saturates at its maximum value and EH becomes the MV in use.

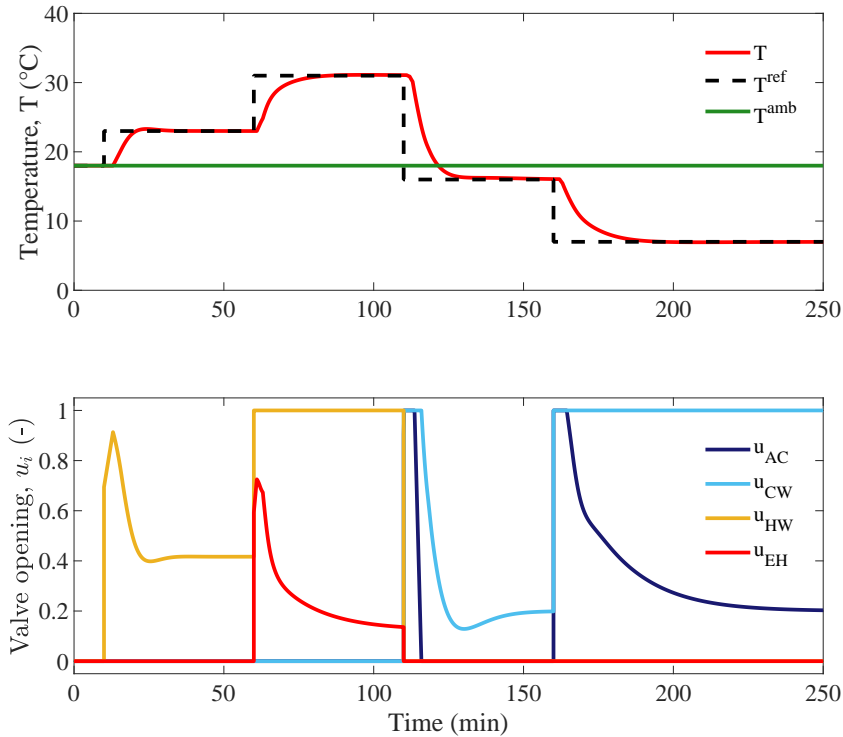


Figure 5.7: Closed-loop response for changes in temperature set-point ( $T^{ref}$ ).

## 5.6 Discussion

It should be noted that every time that split range control is used, there are two other alternative control structures that should be considered. One is to use separate controllers for each MV, but with different CV set-points (see Chapter 7). The ordering of the MV use is then determined by the set-point values. This structure can be economically optimal in certain cases. The second alternative is to use valve position control on the primary MV. This alternative gives a loss (back-off) because one can never reach the constraint for the primary MV, but the advantage is that the same MV is always controlling the CV. These three alternative structures were compared on a simple case study in Chapter 4.

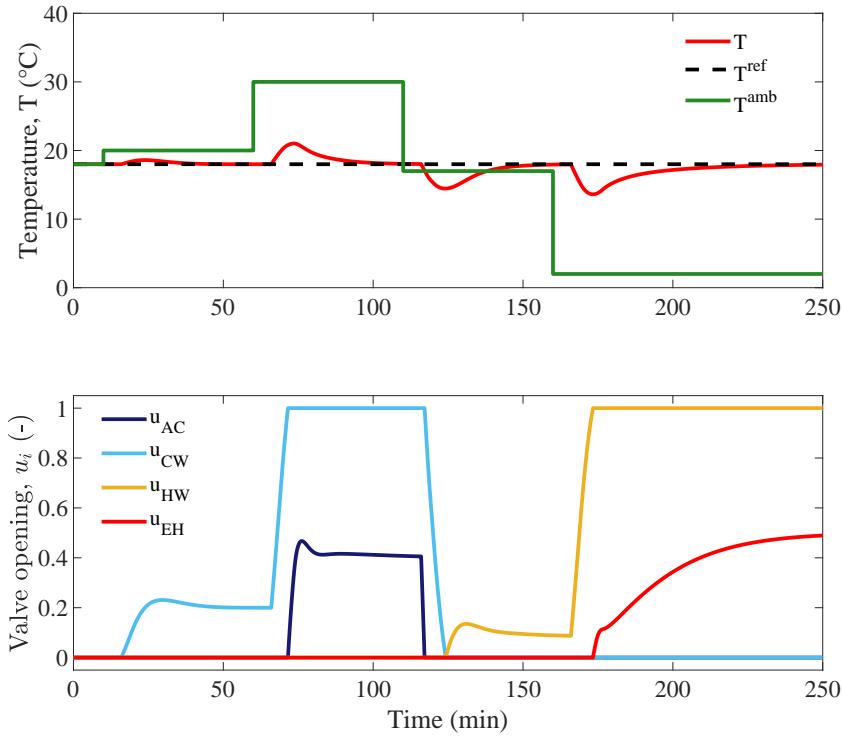


Figure 5.8: Closed-loop response for changes in ambient temperature ( $T^{amb}$ ).

## 5.7 Concluding remarks

Split range control is used when we want to switch manipulated variables (MVs). We show how to use the slopes ( $\alpha_i$ ) in the split range block, or equivalently the split values ( $v^*$ ), as parameters to get the desired controller for each  $MV_i$ , using Eq. (5.4) and Eq. (5.5).

Based on this, we propose a systematic procedure to design split range control structure. An important step of this procedure is the ordering of MVs in step S5. This procedure can be applied to any number of MVs that are used to control one controlled variable (CV).



# Chapter 6

## Generalized Split Range Control

This chapter corresponds to:

Reyes-Lúa, A. and Skogestad, S. (2019a). Multi-input single-output control for extending the operating range: Generalized split range control using the baton strategy. *Journal of Process Control (Under review)*

Split range control is used to extend the steady-state operating range for a single output (controlled variable) by using multiple inputs (manipulated variables). The standard implementation of split range control uses a single controller with a split range block, but this approach has limitations when it comes to tuning.

In this chapter, we introduce a generalized split range control structure that overcomes these limitations by using multiple independent controllers with the same setpoint. Undesired switching between the controllers is avoided by using a baton strategy where only one controller is active at a time. As an alternative solution we consider model predictive control (MPC), but it requires a detailed dynamic model and does not allow for using only one input at a time.

### 6.1 Introduction

Classical advanced control extends the single-loop PID-controller to cover more difficult control tasks and includes, for example, cascade control, feedforward control, decoupling, selectors, split range control, parallel control, and valve position control (also called input resetting or mid-ranging control) (e.g. Buckley (1964); Shinskey (1988); Seborg et al. (2003)). When we need more than one input ( $u_i$ ) to cover the steady-state operating range for a single output ( $y$ ), we can use three alternative classical control structures:

1. Standard split range control (Fig. 6.1),
2. One controller for each input, each with a different setpoint for the output (Fig. 6.2),
3. Input (valve) position control (Fig. 6.3).

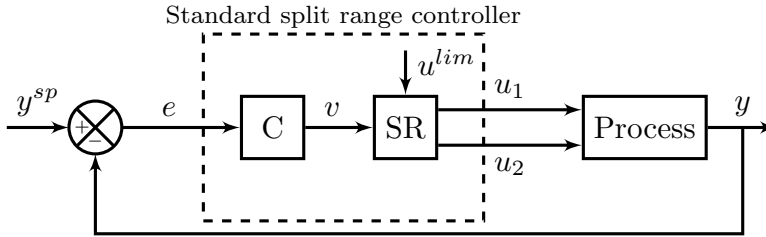


Figure 6.1: Classical structure 1: Standard implementation of split range control with two inputs ( $u_i$ ) and one output ( $y$ ). A typical SR-block is shown in Fig. 6.4. Note that  $v$  is a non-physical internal signal, whereas  $u_i$  is the physical input.  $u^{lim}$  contains information about the maximum and minimum input values, which the SR-block uses to decide on the input switching.

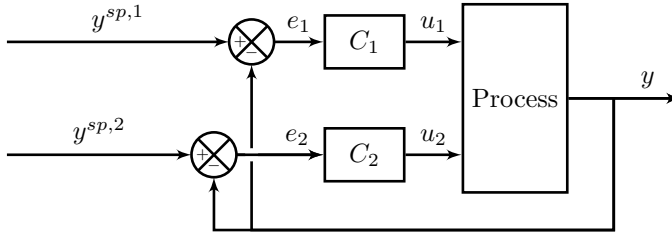


Figure 6.2: Classical structure 2: Two controllers and two inputs for the same output ( $y$ ), each controller with a different setpoint ( $y^{sp,1}$  and  $y^{sp,2}$ ).

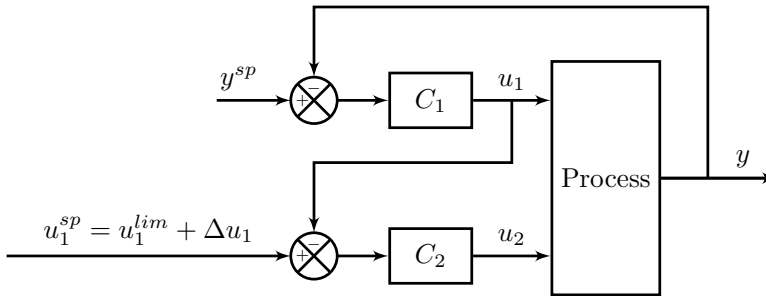


Figure 6.3: Classical structure 3: Input (valve) position control to control one output ( $y$ ) with two inputs ( $u_1$  and  $u_2$ ).

Strategies in Figs. 6.1, 6.2 and 6.3 can be used to extend the steady-state range when the primary input  $u_1$  reaches its limit ( $u_1^{lim}$ ). For example, we may have two sources of heating and we use the second most expensive source only when the first one has reached its maximum. In other cases, the available inputs have opposite effects on the controlled variable; for example, a process that requires both heating and cooling. In this case, switching occurs when heating or cooling reach their lower limit of zero.

Split range control (Fig. 6.1) has been in use for more than 75 years (Eckman, 1945; Fink, 1945). Some other names that have been used for split range control are *dual control agent* (Eckman, 1945), *range extending* (Bristol, 1980) and *valve sequencing* (Lipták, 1985). Split range control has been extensively applied in industry (Lipták, 1985; Sun et al., 2015), but except for basic descriptions and examples of applications (see Stephanopoulos (1983, 1984); Hägglund (1997); Marlin (2000); Bequette (2002); Seborg et al. (2003); Åström and Hägglund (2006); Lieberman (2008); Smith (2010)), we have not found a systematic design procedure, and there are almost no academic studies. Therefore, in Chapter 5 (Reyes-Lúa et al., 2019b), we proposed a systematic procedure to design the standard (classical) split range controller in Fig. 6.1. However, as we explain in Section 6.2, standard split range control has limitations in terms of tuning. For example, we must use the same integral time for all inputs, which is generally not desirable for dynamic performance.

To allow for independent controller tunings, one alternative is to use multiple controllers with different setpoints (Fig. 6.2). For example, when controlling the temperature in a room ( $y = T$ ), one may use  $y^{sp,1} = 23^\circ C$  as the setpoint for cooling ( $u_1$ ) and  $y^{sp,2} = 21^\circ C$  as the setpoint for heating ( $u_2$ ) (Reyes-Lúa and Skogestad, 2019b). Then, on hot days, we use cooling ( $u_1$ ) and keep the temperature at  $y^{sp,1} = 23^\circ C$ . If we have a disturbance in the outdoor temperature so that it decreases, say below  $20^\circ C$ , the controller will reduce the cooling until it reaches its lower limit,  $u_1^{min} = 0$ , and we temporarily lose control of the output ( $y = T$ ). Eventually, the room temperature will decrease to  $y^{sp,2} = 21^\circ C$  and the second controller will start using the heating ( $u_2$ ). The use of different setpoints is to avoid undesired switching between the controllers and possible non-uniqueness when using two controllers with integral action to control the same output (Åström and Hägglund, 2006). The use of multiple controllers with different setpoints is analyzed in Chapter 7.

The third classical control structure for extending the steady-state range is shown in Fig. 6.3. Here, the output ( $y$ ) is always controlled with the same input ( $u_1$ ), but if  $u_1$  approaches its limit ( $u_1^{lim}$ ), then input  $u_2$  is activated and keeps  $u_1$  away from its limit. Normally, input  $u_2$  is not used, that is, we have  $u_2 = u_2^{lim}$ , where typically  $u_2^{lim} = 0\%$ .

The advantage with input (valve) position control is that we always use the same input ( $u_1$ ) to control  $y$ , but the disadvantage is that we cannot utilize the full range for  $u_1$  as we need a back-off ( $\Delta u_1$ ) from the limit. For example, we may select to always use cooling ( $u_1$ ) to control the room temperature ( $y = T$ ). On cold days, we use heating ( $u_2$ ) to avoid that the cooling reaches its lower limit ( $u_1^{min} = 0$ ). Thus, even on cold days we will use a little cooling ( $u_1$ ). Note that the term "valve position control" is more commonly used for another case than in Fig. 6.3, namely when  $u_1$  is used to improve the *dynamic performance* of  $y$ . In this case  $u_2$  is the main manipulated variable for steady-state control and  $u_1$  is always controlled to its setpoint ( $u_1^{sp}$ ), which is typically a "midrange" value.

In the present chapter, we propose a generalized split range control structure (Fig. 6.5), where the controller for each input can be designed independently. To avoid the use of different setpoints (Fig. 6.2), we use a *baton strategy*, in which undesired switching is avoided by allowing only one controller to be active at a time.

This chapter is organized as follows. In Section 6.2 we briefly describe standard split range control and its limitations with respect to tuning. In Section 6.3 we present the new generalized structure, which overcomes these limitations. In Section 6.4 we use a case study to illustrate our proposed generalized structure and compare it to standard split range control. Then, in Section 6.5 we discuss possible alternative implementations and control strategies for multiple-input single-output control for extending the operating range, including model predictive control (MPC). We conclude the chapter in Section 6.6.

## 6.2 Standard split range controller

As shown in Fig. 6.1, in standard split range control there is one common controller ( $C$ ) which computes the internal signal ( $v$ ) to the split range block (SR-block), which assigns the value (e.g., the valve opening) for each input ( $u_i$ ). Importantly, at any particular time, only one input ( $u_i$ ) is being used to control the output ( $y$ ), whereas the remaining inputs are fixed at the values given by  $u^{lim}$ , typically at their maximum or minimum values.

### 6.2.1 The split range block

The split range block has also been called *characterization function* (Smith, 2010), *splitter block* (Blevins, 2011), and *function generator* (Smith, 2014). Fig. 6.4 depicts a typical split range block for two inputs ( $u_1$  and  $u_2$ ) for a case when  $u_1$  has a positive effect on the output ( $y$ ) and  $u_2$  has a negative effect.

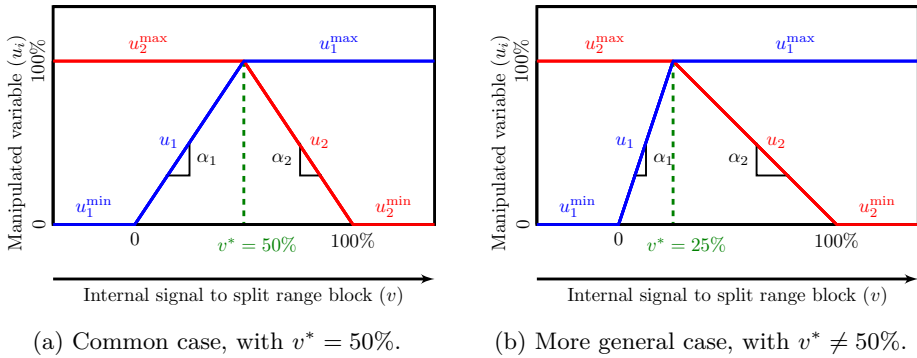


Figure 6.4: Typical split range (SR) block for Fig. 6.1.

For example, normally we may want to control room temperature ( $y$ ) with heating ( $u_1$ ) and with the ventilation rate ( $u_2$ ) set at its maximum. However, on a very cold day we may reach maximum heating ( $u_1^{\max}$ ), and to maintain temperature control ( $y$ ), we can reduce the ventilation rate ( $u_2$ ).

### 6.2.2 Slopes ( $\alpha_i$ ) in split range block

In Fig. 6.4a, the split value is located at the mid-point ( $v^* = 50\%$ ) and the slopes have the same magnitude ( $|\alpha_1| = |\alpha_2|$ ). This choice is used in most examples in the literature (see Mercer (1968); Price et al. (1994); Piovoso et al. (1995); Seborg et al. (2003); Bequette (2002); Wang (2010); Smith (2010); Fonseca et al. (2013); Shen-Huii et al. (2011); Fatani et al. (2017); Bahadori (2017)).

However, each input ( $u_i$ ) has a different dynamic and static effect on the output, and the split value  $v^*$  (or equivalently, the slopes  $\alpha_i$ ) should generally be located at some other value to compensate for this, as illustrated in Fig. 6.4b (see Lipták (1985); Häggglund (1997); Glemmestad (1997); Bastani and Chen (1988); Blevins (2011); Gupta et al. (2015); Reyes-Lúa et al. (2019b)). For example, with a PI-controller (with parameters  $K_C$  and  $\tau_I$ ), we can get the desired controller gain<sup>1</sup> for each input ( $K_{C,i}$ ) by selecting the slopes ( $\alpha_i$ ) such that  $\alpha_i K_C = K_{C,i}$ . However, we need to use a common integral time ( $\tau_I$ ) which must be a compromise among the desired  $\tau_{I,i}$  for every  $u_i$ .

<sup>1</sup>The desired tunings for these controllers can be found, for example, from the SIMC PID tuning rules (Skogestad, 2003).

### 6.3 Generalized split range control structure

#### 6.3.1 Proposed baton strategy

The dynamic behavior of each input is generally different and using a common controller ( $C(s)$ ), as in standard split range control (Fig. 6.1), represents a compromise. Fig. 6.5 depicts our proposed generalized control structure for split range control applications, where each input has its own controller ( $C_i(s)$ ). Here,  $C_i(s)$  can be any type of controller, but it is commonly a PID controller<sup>2</sup>. Each controller produces a suggested input  $u'_i$ , and the *baton strategy logic* block in Fig. 6.5 selects and computes the actual inputs  $u_i$ .

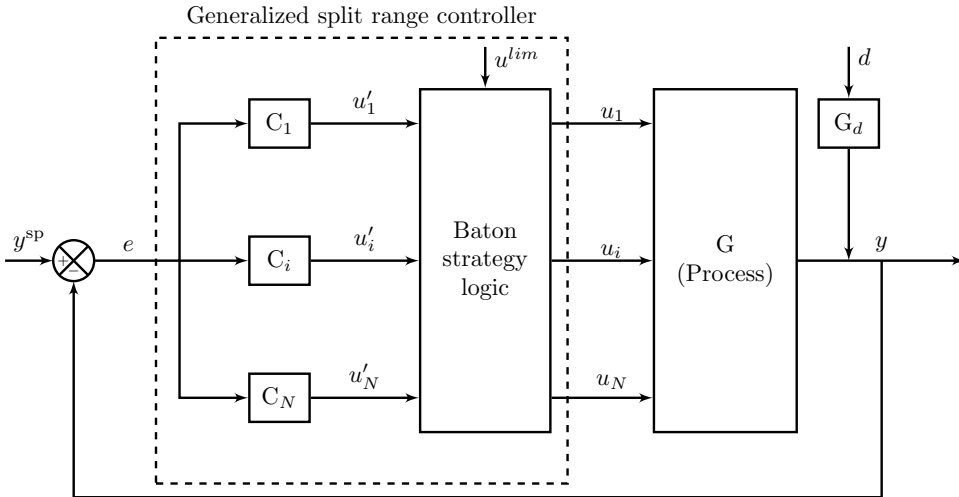


Figure 6.5: Proposed generalized split range control using the baton strategy. Each controller computes a suggested input  $u'_i$  and the baton logic decides on the actual input  $u_i$ .

In order to use multiple controllers for the same output, we want to make sure that *only one input ( $u_i$ ) is actively controlling the output ( $y$ ) at any given time*. The other inputs are required to be at fixed values ( $u_i^{min}$  or  $u_i^{max}$ ), as given in  $u^{lim}$ . We propose to do this by using a *baton strategy logic*, similar to what is used by runners in a relay (Fig. 6.6), where only the runner who holds the baton is active at any given time, and the active runner decides when to pass the baton. This avoids the need for a centralized supervisor.

<sup>2</sup>Having independent controllers ( $C_i(s)$ ) allows one to individually tune the controller for each input  $i$ , without any compromise. To design  $C_i(s)$  we suggest using a systematic tuning procedure, such as the SIMC PID tuning rules (Skogestad, 2003).

In other words, we *let the active input decide when to switch to another input*. The *active input* remains active as long as its not saturated ( $u_i^{\min} < u_i < u_i^{\max}$ ) and will only pass the baton to another input once it becomes saturated (reaches  $u_i^{\min}$  or  $u_i^{\max}$ ).



Figure 6.6: Baton strategy for relay.

### 6.3.2 Sequencing of inputs

Before actually designing the baton strategy logic, we need to make some initial decisions. First, we need to define the minimum and maximum values for every input ( $u_i^{\min}$ ,  $u_i^{\max}$ ) This is decision D1. Then, we need to choose the sequence of use of the inputs (decision D2). This should be defined considering their effect on the output ( $y$ ) as well as economic aspects. In some cases, operational aspects may be taken into account. The following steps are used for decision D2:

D2.1 Define the desired or most economical operating value for each input (e.g. fully closed or fully open valve).

D2.2 Consider the effect of every input ( $u_i$ ) on the output ( $y$ ). Then group the inputs into:

- (a) Inputs for which the value of the output ( $y$ ) *increases* when we move  $u_i$  away from its desired operating value (fully opened or fully closed).
- (b) Inputs for which the value of the output ( $y$ ) *decreases* when we move  $u_i$  away from its desired operating value (fully opened or fully closed).

D2.3 Within each group, (a) and (b), order the inputs according to which one should be used first (less expensive) to which should be used last (more expensive).

D2.4 In our experience, it is usually helpful to graphically summarize the final sequence in a standard split range block, as the one in Fig. 6.4 (and Fig. 6.8 in the case study), but note that the slopes and the split values have no significance when we use the generalized split range control structure that we are proposing.

### 6.3.3 Baton strategy logic

Once that the sequence of inputs is defined, we can formulate the logic for the baton strategy. Consider that input  $k$  is the active input (*has the baton*). The proposed baton strategy is then:

B.1 Controller  $C_k$  computes  $u'_k$ , which is the suggested value for the input  $k$ .

B.2 If  $u_k^{min} < u'_k < u_k^{max}$

- a) keep  $u_k$  active, with  $u_k \leftarrow u'_k$
- b) keep the remaining inactive inputs  $i$  at the relevant limit value ( $u_i^{min}$  or  $u_i^{max}$ ).

B.3 If  $u'_k \leq u_k^{min}$  or  $u'_k \geq u_k^{max}$

- a) Set  $u_k = u_k^{min}$  or  $u_k = u_k^{max}$ , depending on which limit is used, and *pass the baton* to the new active input  $j$ . The new active input is selected according to the predefined sequence, depending on which limit is met ( $j = k + 1$  or  $j = k - 1$ ).
- b) Set  $k = j$  and go to step B.1.

### 6.3.4 Anti-windup strategy

One needs to avoid windup for the inputs which are not active. Thus, when switching, one needs to decide on how to initialize the new active controllers. There are several alternatives. Since we only want one controller to be active at a time, the simplest and most obvious strategy is to set all the states of the non-active controllers to zero. For a PI controller (Eq. (6.1)), this means that the integral action starts at the time of the switching,  $t_b$ , *when the baton is passed*.

$$u'_k(t) = u_k^0 + K_{C,k} \left( e(t) + \frac{1}{\tau_{I,k}} \int_{t_b}^t e(t) \right) \quad (6.1)$$

The value of the bias  $u_k^0$  is equal to  $u_k^{min}$  or  $u_k^{max}$ , depending on from which side the baton was received. Note that the integration in Eq. (6.1) starts from  $t_b$  and not from 0.

This simple strategy is used in the case study. Alternative anti-windup implementations are described in the discussion (Section 6.5.4).



## 6.4 Case Study: Control of room temperature

In this section, we demonstrate the implementation and performance of our proposed generalized split range control structure with a temperature control case study. We compare our proposed generalized control structure with the standard split range control described in Section 6.2 and Chapter 5. We want to control room temperature ( $y = T$ ) using four inputs ( $u_i$ ), two sources of cooling and two sources of heating:

- $u_{AC}$ : air conditioning
- $u_{CW}$ : cooling water
- $u_{HW}$ : hot water (district heating)
- $u_{EH}$ : electric heating.

The setpoint for the room temperature is  $T^{sp} = 18^\circ\text{C}$ . The main disturbance is ambient temperature ( $d = T^{amb}$ ), which is not measured and is nominally the same as the setpoint; thus,  $T_0^{amb} = 18^\circ\text{C}$ . This means that no heating or cooling is required at the nominal operating point ( $u_i = 0 \ \forall i$ ), which is desired for economic reasons. In this example, all four inputs ( $u_i$ ) are scaled from 0 to 1.

### 6.4.1 Model

For simplicity, we model the room as a linear system:

$$y(s) = G_p(s) u(s) + G_d(s) d(s) \quad (6.2)$$

where:

$$y = T \quad (6.3a)$$

$$u = [u_{AC} \ u_{CW} \ u_{HW} \ u_{EH}]^\top \quad (6.3b)$$

$$d = T^{amb} \quad (6.3c)$$

$$G_p(s) = [G_{AC}(s) \ G_{CW}(s) \ G_{HW}(s) \ G_{EH}(s)] \quad (6.3d)$$

Table 6.1 shows the gains ( $K_{p,i}$ ), time constants ( $\tau_i$ ) and delays ( $\theta_i$ ) for  $G_{p,i}(s)$  and  $G_d(s)$ , modeled as first-order transfer functions). Note that since the gain for the disturbance in ambient temperature ( $d = T^{amb}$ ) is 1 and the inputs ( $u_i$ ) are scaled in the range 0 to 1, the gains  $K_{p,i}$  tell us the disturbance range that each input can handle. For example, since  $K_{p,HW} = 12$  we can handle ambient temperatures down to  $T^{amb} = T_0^{amb} - K_{p,HW} = 18^\circ\text{C} - 12^\circ\text{C} = 6^\circ\text{C}$  before we must switch from hot water (HW) to electric heating (EH). Furthermore, since  $K_{p,EH} = 8$  we can

handle ambient temperatures down to  $6^\circ\text{C} - 8^\circ\text{C} = -2^\circ\text{C}$  before we lose control of room temperature ( $y = T$ ) because both heating sources (HW and EH) are at their maximum. In the other direction, we can handle ambient temperatures up to  $T_0^{amb} - K_{p,CW} - K_{p,AC} = 18^\circ\text{C} + 5^\circ\text{C} + 10^\circ\text{C} = 33^\circ\text{C}$  before we lose control of  $y = T$  because both cooling sources (AC and CW) are at their maximum.

Table 6.1: Parameters for  $G_{p,i}(s)$  from  $u_i$  to  $y = T$  and  $G_d(s)$  from  $d = T^{amb}$  to  $y = T$ .

$G_i$	$K_{p,i}$	$\tau_i$ (min)	$\theta_i$ (min)
$G_{AC}$	-5	8	2
$G_{CW}$	-10	15	3
$G_{HW}$	12	10	3
$G_{EH}$	8	5	1
$G_d$	1	15	6

### 6.4.2 Standard implementation of split range control

Fig. 6.7 shows the block diagram for the standard implementation of split range control for this process, using one common PI controller ( $C$ ) and the split range block in Fig. 6.8. For the common PI controller we choose  $K_C = 0.0592$  and  $\tau_I = 15$  min. Table D.1 in Appendix D.1 summarizes the parameters for the standard split range block in Fig. 6.8. This control structure was designed and tuned as proposed in Chapter 5 (Reyes-Lúa et al., 2019b).

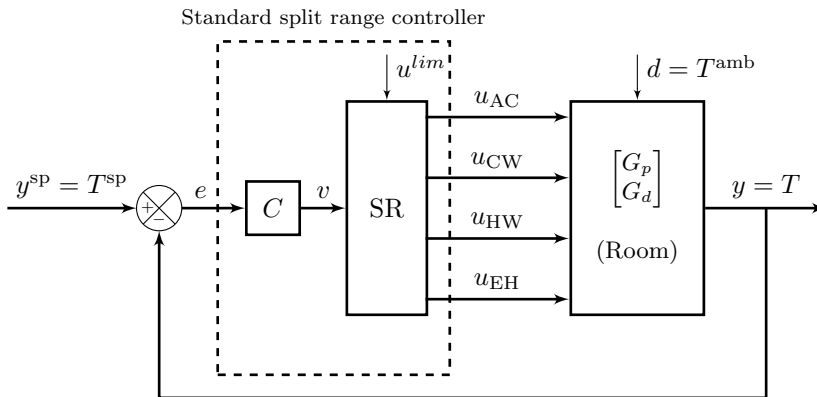


Figure 6.7: Block diagram for standard split range control for room temperature control. The SR block is shown in Fig. 6.8.

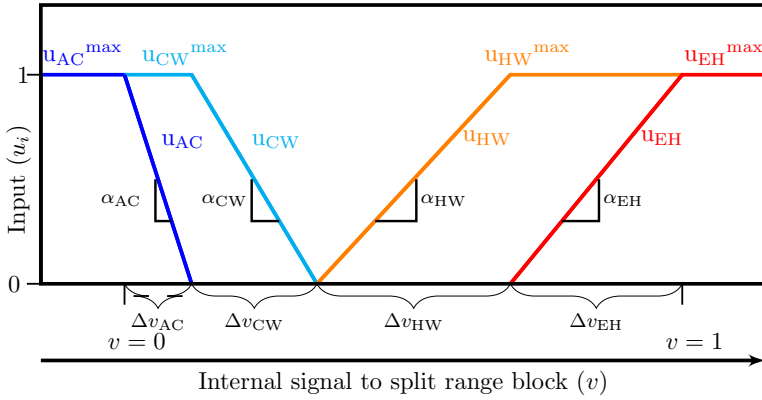


Figure 6.8: Standard split range block for room temperature control with air conditioning (AC), cooling water (CW), hot water (HW), and electric heating (EH); SR block in Fig. 6.7.

### 6.4.3 Generalized implementation of split range control

Fig. 6.9 shows the block diagram for the new proposed generalized split range control structure. We use PI controllers for each input and tune each loop "tightly", according to the SIMC tuning rules (Skogestad, 2003). This is achieved by selecting the closed-loop time constant for each input equal to the time delay ( $\tau_{c,i} = \theta_i$ ). Table 6.2 gives the PI tuning parameters for each  $C_i(s)$ .

Table 6.2: Tuning parameters in room temperature control.

$u_i$	$\tau_{c,i}(min)$	$K_{C,i}$	$\tau_{I,i}(min)$
$u_{AC}$	$\theta_{AC}$	-0.4000	8
$u_{CW}$	$\theta_{CW}$	-0.2500	15
$u_{HW}$	$\theta_{HW}$	0.1389	10
$u_{EH}$	$\theta_{EH}$	0.3125	5

We next design the generalized split range control structure according to the procedure in Section 6.3.

#### Sequencing of outputs

D1 The inputs are normalized, and the operating range for every input is  $u_i = [0, 1]$ .

D2.1 The most economical operating point is when  $T^{amb} = T^{sp}$ , and we can have all inputs fully closed ( $u_i = 0$ ).

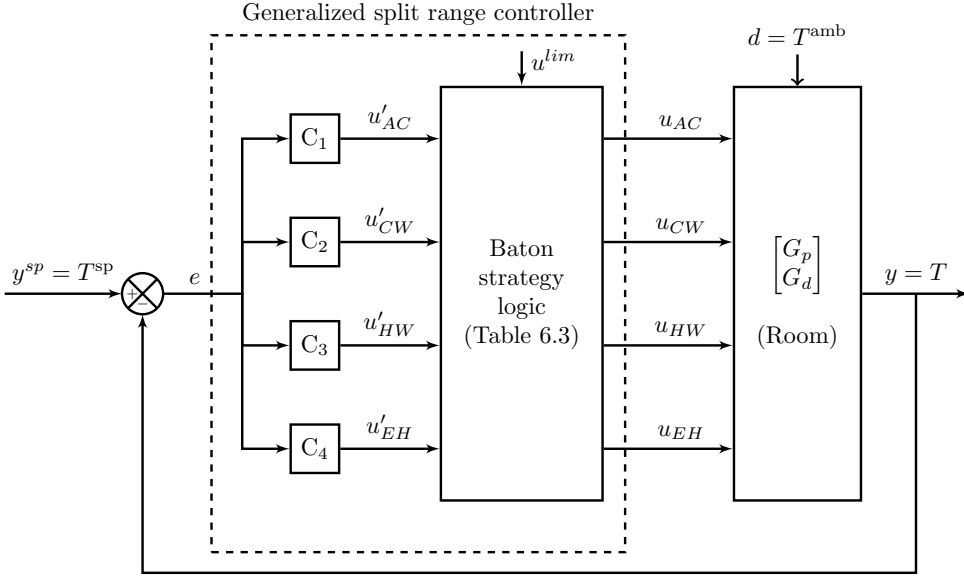


Figure 6.9: Generalized split range control solution for room temperature control.

D2.2 To maintain  $T = T^{sp}$ , we need to cool the room if  $T^{amb} > T^{sp}$ , and to heat the room if  $T^{amb} < T^{sp}$ . We can group the inputs according to their effect on the room temperature into:

- (a) Inputs for which  $y = T$  *increases* when we open them (move away from the desired operating condition, fully closed). These are the two heating sources: HW and EH.
- (b) Inputs for which  $y = T$  *decreases* when we open them (move away from the desired operating condition, fully closed). These are the two cooling sources: CW and AC.

D2.3 As CW is less expensive than AC, we prioritize the use of CW over AC for decreasing room temperature. Likewise, we prioritize the use of HW over EH.

D2.4 The final sequence can be summarized in the split range block in Fig. 6.8. However, note that when using the generalized control structure the values of the slopes ( $\alpha_i$ ) have no significance except for the sign, which determines whether we start from  $u_i^{min}$  or  $u_i^{max}$ .

### Design of the baton strategy.

We consider the block diagram in Fig. 6.9 and use Fig. 6.8 to define the sequence and the choice of bias. The proposed baton strategy logic in steps B.1 to B.3 is written out in detail in Table 6.3.

Table 6.3: Baton strategy logic for case study.

Value of $u'_k$	Active input (input with <i>baton</i> , $u_k$ )			
	$u_1 = u_{AC}$	$u_2 = u_{CW}$	$u_3 = u_{HW}$	$u_4 = u_{EH}$
$u_k^{min} < u'_k < u_k^{max}$	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 $u_3$ $u_3^0 = u_3^{max}$

When an input *receives the baton*, the integrator of its corresponding PI controller ( $C_k(s)$ ) is reset to zero, according to Eq. (6.1). Thus, the initial value for  $u_k$  (at time  $t = t_b$ ) will be the proportional term plus  $u_k^0$

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

Note here that the bias,  $u_k^0$ , is equal to  $u_k^{max}$  or  $u_k^{min}$ , depending on from which side the baton is coming (see Table 6.3). Note that when  $u_1 = u_{AC}$  or  $u_4 = u_{EH}$  reach their corresponding  $u_i^{max}$ , we reach the limit of the range within which we can control  $y = T$ . As there is no other input to *pass the baton*, they remain the "active" input. In those cases, we lose control of  $T$  because all inputs are constrained.

## Simulations

The standard and the generalized split range control schemes are tested for rejection of disturbances in  $T^{amb}$ , which is nominally  $18^\circ C$ .  $T^{sp}$  is kept constant at  $18^\circ C$ . At  $t = 10 \text{ min}$ ,  $T^{amb}$  increases to  $20^\circ C$  and at  $t = 80 \text{ min}$  to  $29^\circ C$ . Then, at  $t = 140 \text{ min}$ ,  $T^{amb}$  decreases to  $24^\circ C$  and at  $t = 180 \text{ min}$  to  $-1^\circ C$ .  $T^{amb}$  then increases to  $17^\circ C$  at  $t = 280 \text{ min}$ , and finally to  $22^\circ C$  at  $t = 350 \text{ min}$ .

From Fig. 6.10, we observe that both the standard and the generalized implementation maintain  $T = T^{sp}$  at steady-state, but the generalized structure is better as it reaches steady-state much faster, except for the disturbances at  $t = 10 \text{ min}$  and  $t = 140 \text{ min}$  when CW (cooling water) is the active input. This is expected because the integral time for the common controller for standard split range control is  $\tau_I = 15 \text{ min}$ , which is the same as for CW with generalized SRC (see Table 6.2). For the other inputs, the integral time for generalized SRC is smaller (8, 10, and 5 min), resulting in a faster return to the setpoint.

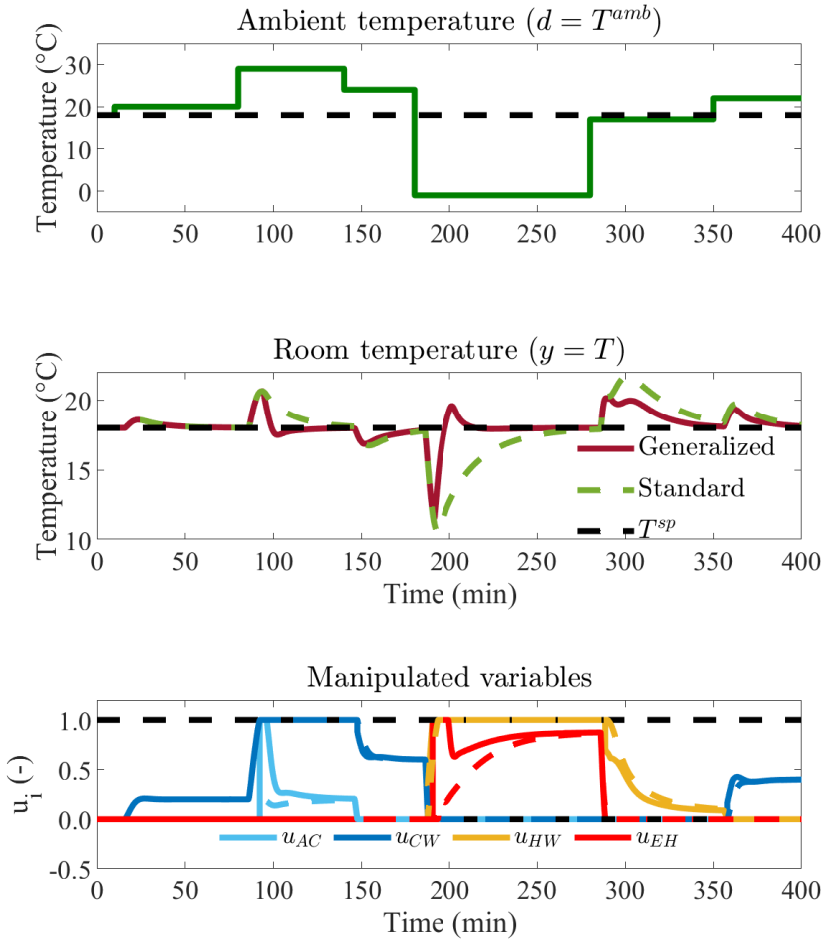


Figure 6.10: Comparison of standard and generalized split range controller (SRC) for case study. The standard SRC structure is shown in Fig. 6.7 and the generalized structure is shown in Fig. 6.9.

## 6.5 Discussion

### 6.5.1 Alternative implementations of generalized split range control

In standard split range control, we can use the slopes in the split range block to adjust the controller gain for each input, but we have to use the same value for the other controller settings, like the integral or derivative times. By “generalized split range control” we mean an implementation where the controllers for each input can be tuned independently. Various statements on using independent controllers have appeared in the literature (Åström and Hägglund, 1995; Hägglund, 1997; Marlin, 2000) but we did not find any details on how it should be implemented or whether it had been used in practice.

During the work with this project, we tried several alternative implementations. Our first attempt was to use a common integrator and put the dynamics after the split range block in Fig. 6.1. For example, to change the PI-tunings from the original set 1 (in  $C$ ) to the set 2, we may add a block

$$K_{C,2}/K_{C,1}(1 + 1/\tau_2 s)/(1 + 1/\tau_1 s)$$

on the signal  $u_2$  exiting the split range block. However, the signal  $u_2$  is a physical signal, which already includes its maximum or minimum value, and adding dynamics to the signal creates non-uniqueness in the switching.

Our next attempt was to have one controller  $C(s)$ , as in Fig. 6.1, and use different sets of parameters in  $C(s)$  based on the output from the split range block, which tells which input is active. Åström and Hägglund (1995) and Hägglund (1997) refer to this idea as a special type of gain scheduling. However, the term gain scheduling is generally used for the case where the inputs and outputs are fixed and we change the controller parameters depending on the operating parameters, for example, the setpoint ( $y^{sp}$ ) or the disturbance ( $d$ ). On the other hand, split range control is used to extend the steady-state range of  $y$  by using a sequence of different inputs. In any case, we encountered problems with implementing this approach. This is because when we change the controller parameters for  $C(s)$ , the signal  $v$  from  $C(s)$  changes, which may cause the selector block to change the active input, resulting in cycling and non-uniqueness in the switching.

We therefore decided to use independent controllers. However, only one controller should be active at the time, and to select which one, we introduced the baton strategy. The baton strategy has the advantage that the selection of the active input is not centralized. Each active controller only needs to know which two controllers it should give the baton to if it reaches its maximum or minimum value, respectively.

### 6.5.2 Comparison with multiple controllers with different setpoints

As mentioned in the introduction, an alternative to split range control is to use multiple controllers with different setpoints (see Chapter 7). In this case, all controllers are active at any given time (although some inputs may be saturated), so to avoid undesired switching and fighting, one has to separate the setpoints.

Our new generalized split range controller may be viewed as an extension of this, which avoids the use of different setpoints (Fig. 6.2). The use of different setpoints has the advantage of avoiding the logic block in Fig. 6.5, as the sequence of the inputs is indirectly given by the value of the setpoints. For example, for our room temperature case study, we could have used four controllers with setpoints  $20^\circ\text{C}$  for AC,  $19^\circ\text{C}$  for CW,  $18^\circ\text{C}$  for HW and  $17^\circ\text{C}$  for EH. We are here assuming that we can have tight temperature control so that a setpoint difference of  $1^\circ\text{C}$  is enough to avoid undesired switching.

### 6.5.3 Comparison of split range control with model predictive control

One obvious design approach to handle MISO systems with input constraints is model predictive control (MPC) (Qin and Badgwell, 2003). The standard approach in MPC is to use the weights in the objective function to assign the priorities for the control objectives. To assure that the controller uses the right input, we need to introduce penalties on deviations in the inputs ( $u_i$ ) from the desired value and the values of the weights should be higher for more costly inputs. As there is no systematic way of choosing the weights or tuning rules for MPC, we used trial and error. However, depending on the selected weights, this scheme may not always bring the output ( $y$ ) to zero offset and dynamically it may use more than one input simultaneously, which is not necessarily the desired strategy.

The simulation in Fig. 6.11 compares the proposed generalized SRC with MPC. The generalized SRC is the same as the one studied earlier (Fig. 6.9 and 6.10) and the details of MPC are given in Appendix D.2. The MPC weight for setpoint deviation is ten times higher than the weight for the use of the expensive inputs ( $u_1 = u_{AC}$  and  $u_4 = u_{EH}$ ) and the weight for the expensive inputs is five times higher than the weight for the less expensive inputs ( $u_2 = u_{CW}$  and  $u_3 = u_{HW}$ ). In general, we see that MPC has better initial response, because it uses several inputs at the same time, but the settling towards the steady-state is slower than with generalized SRC. This is also seen from the values of the integrated absolute error in Table 6.4.



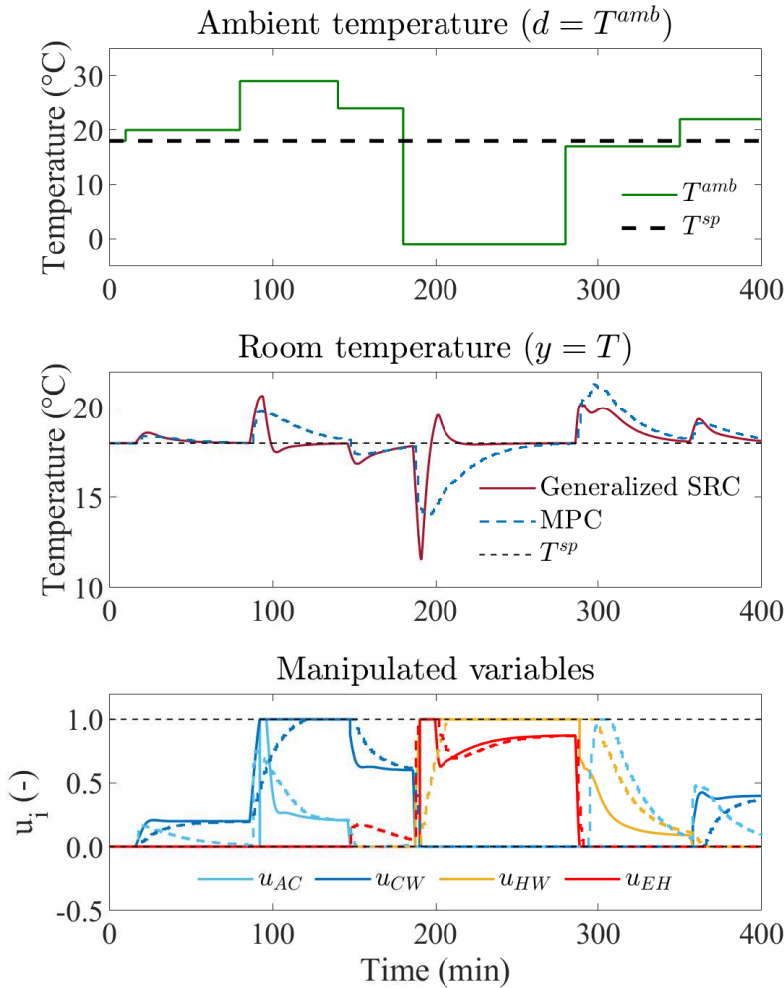


Figure 6.11: Comparison of standard MPC with generalized split range control (SRC) for room temperature. The dashed lines correspond to MPC and the solid lines correspond to the strategy proposed in this chapter, which is also depicted in Fig. 6.10.

For example, consider the response at  $t = 280 \text{ min}$ , when  $d = T^{amb}$  increases from  $-1^\circ\text{C}$  to  $17^\circ\text{C}$  and the room requires much less heating than before but still no cooling. Indeed, split range control handles this disturbance by only limiting the heating. It first turns off the electrical heating ( $u_{EH}$ ) and then controls the temperature by reducing the hot water ( $u_{HW}$ ). MPC also turns off  $u_{EH}$  initially,

but then it starts using the cooling water ( $u_{CW}$ ) while at the same time reducing the hot water ( $u_{HW}$ ). MPC uses cooling to speed up the initial response, but this is not beneficial on a longer time scale as seen from the simulations. Moreover, the input usage is also higher. It is not only dynamically that MPC may use more than one input; it also happens at steady state, at least with quadratic input weights, as in our case study.

Note that the sampling time for the MPC is  $\Delta t = 1 \text{ min}$ , whereas split range control is continuous. This partly explains why SRC is faster than MPC for the disturbance at  $t = 280 \text{ min}$ . The actual performance of MPC will depend on the tuning. Nevertheless, the main disadvantages with MPC compared to SRC are that it requires a detailed dynamic model and that it will increase the input cost because it uses several inputs at the same time. For example, as we observed, it may use cooling to avoid a sudden temperature increase, although the disturbance could be handled without cooling.

Table 6.4: Integral absolute error (IAE) for the case study with alternative controllers.

Controller	IAE
Standard split range control (Fig. 6.7): Fig. 6.10	448.6
Generalized split range control (Fig. 6.9), with integrator resetting (Eq. (6.1)): Figures 6.10, 6.11, and 6.13	202.4
Generalized split range control (Fig. 6.9), with back-calculation tracking (Fig. 6.12): Fig. 6.12	235.7
MPC (Eq. (D.3)): Fig. 6.11	327.9

#### 6.5.4 Anti-windup for generalized split range control

In the proposed generalized structure for split range control there are multiple controllers for the same output. In the case study, windup is overcome by having only one controller active at any time and resetting the integrator term to zero when a controller becomes active (see Eq. (6.1)). However, the proportional and derivative terms of the controller may potentially cause large output changes when the switch occurs. This may be partly seen by the value for  $u_{AC}$  in Fig. 6.10 at  $t = 100 \text{ min}$ , which jumps from 0 to 1 for a short time, before settling at about 0.2. Thus, we do not have bumpless transfer, which actually may be an advantage because it may give a faster response.

Windup can be avoided by implementing other anti-windup schemes, such as input tracking with *back-calculation* (Åström and Hägglund, 2006). Fig. 6.12 shows how input tracking with back-calculation can be implemented for each input ( $u_i$ ) with the generalized split range control structure. In the block diagram in Fig. 6.12, the tracking constant,  $K_T$ , (Matlab, 2019) is used to reset the integrator dynamically (Åström and Hägglund, 2006). With  $K_T = 0$ , tracking is turned off and with a large value for  $K_T$ , tracking is fast. If we implement this anti-windup scheme in combination with the generalized split range controller proposed in this chapter, all

desired inputs ( $u'_i$ ) are calculated at any time, this is, we do not reset the integrator of the input that becomes active (*receives the baton*). Otherwise, the switching logic to *transfer the baton* remains the same.

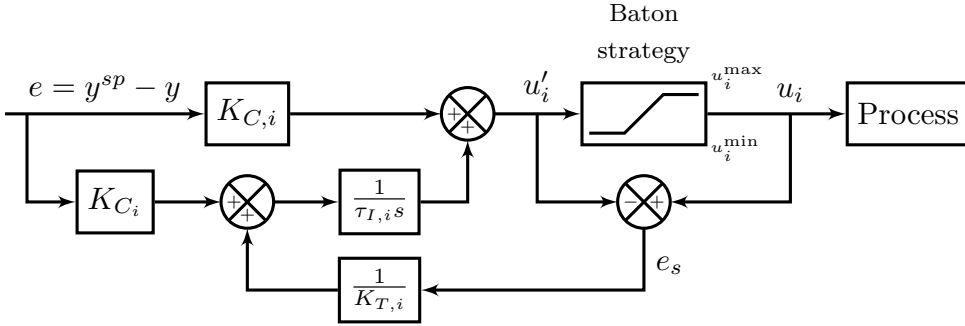


Figure 6.12: Antiwindup with bumpless transfer: input tracking with back-calculation for input  $u_i$ .

Fig. 6.13 compares back-calculation (dashed lines) with the strategy of integral resetting in Eq. (6.1) (solid lines). We implemented the back-calculation tracking scheme in Fig. 6.12, with  $K_T = 1$  for all inputs and with the same PI-tunings (Table 6.2) and switching logic (Table 6.3) as before. The differences are quite small, but as expected, we observe a somewhat less aggressive initial response to the disturbances when we use back-calculation. For example, at  $t = 100 \text{ min}$ ,  $u_{AC}$  does not jump from 0 to 1 as it does with integral resetting. On the other hand, the integrated absolute error (IAE) with back-calculation is somewhat higher than with integral resetting, although it is still significantly lower than with standard split range control (see Table 6.4).

### 6.5.5 Stability for controllers extending the operating range

All the structures considered in this chapter (Figures 6.1, 6.2, 6.3 and 6.5) involve switching between different active controllers. During normal operation, when there is no switching, we achieve robustness by using the SIMC PI tuning with  $\tau_c = \theta$  (Table 6.2), which guarantees a gain margin of about 3 and a delay margin of about 2.5 to 3 (Skogestad, 2003; Grimholt and Skogestad, 2018).

In general, switching may result in oscillations, and indeed, we encountered such problems with some of the other structures we tried (see Section 6.5.1). There exist no general analysis results for switched systems, for say analyzing whether undesired switching will occur. In practice, undesired oscillations may be overcome by introducing something on top of the switching, like adding a delay (Marlin, 2000).

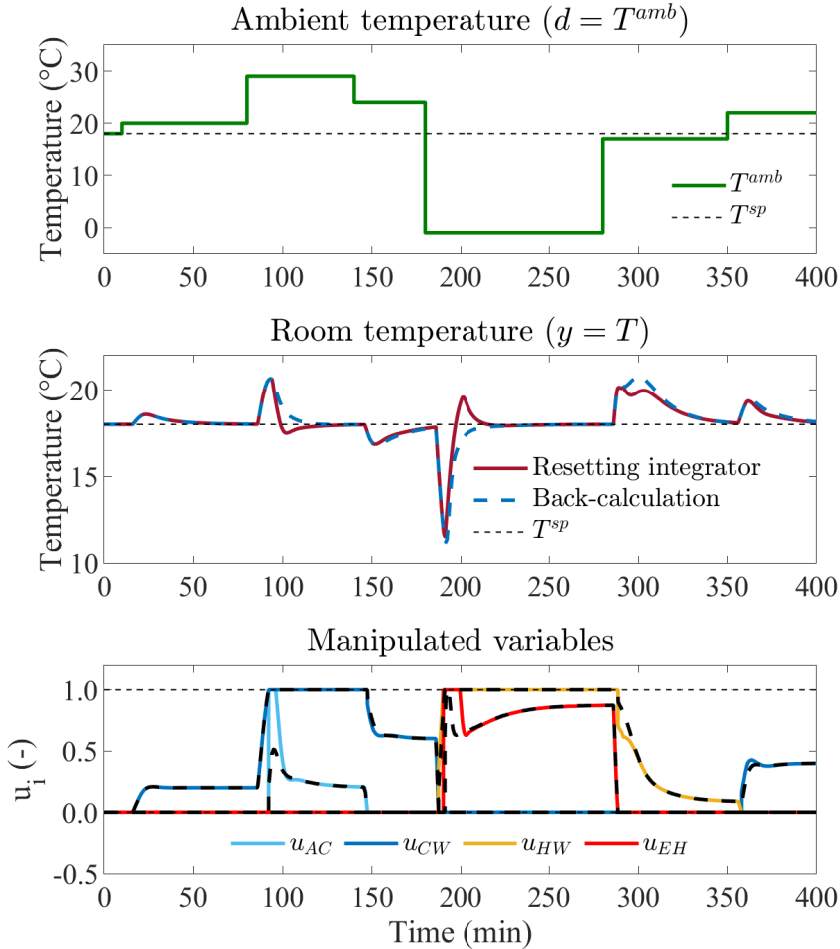


Figure 6.13: Comparison of two anti-windup strategies for the generalized split range control structure in Fig. 6.9. The dashed lines correspond to back-calculation (Fig. 6.12) and the solid lines correspond to the strategy of resetting the integrator (Eq. (6.1)), which is also depicted in Fig. 6.10.

Another option is to use a two-step approach with a compensation (auxiliary) loop to avoid undesired switching. For example, Garelli et al. (2011) propose sliding mode reference conditioning (SMRC), based on variable structure analysis (VSS) theory and sliding mode (SM) related concepts, to improve robustness by shaping the reference signal. In this method, the reference signal is shaped by including a switching block and a first-order low-pass filter in the auxiliary loop. This strategy

can be implemented to avoid bumpy transfers when switching between different controllers.

Regarding MPC, there exist a number of stability results, although an important assumption is that all states are measured or can be perfectly estimated, which is not realistic in most process control problems and does not apply to systems with time delay as in the case study in this chapter. In addition, traditional MPC does not allow for logic variables and therefore does not allow for switching such that only one input is used at the time. However, Bemporad and Morari (1999) developed an MPC strategy which allows for logic variables with closed-loop stability guarantees (again under the assumption that all states are measured), but this assumes the control system has to be designed using the approach proposed in the paper, which involves solving a mixed-integer quadratic program (MIQP), for which there is no guarantee of convergence to a unique solution.

## 6.6 Conclusions

Split range control is widely used in industry, but it has not been studied much in academia. In this work, we introduce a new generalized control structure using a baton strategy that allows for using individual controllers for each available input without a centralized supervisor. The proposed baton strategy is illustrated in Fig. 6.5 and Table 6.3. We demonstrated the feasibility of implementing this structure in a case study with four available inputs and one controlled variable. This new generalized structure has better dynamic performance than the standard split range controller, and also outperforms MPC for our case study.



# Multiple-input single-output control for extending the steady-state operating range: use of controllers with different setpoints

This chapter is published as:

Reyes-Lúa, A. and Skogestad, S. (2019b). Multiple-Input Single-Output Control for Extending the Steady-State Operating Range—Use of Controllers with Different Setpoints. *Processes*, 7(12):941.

This chapter deals with the case when multiple inputs are needed to cover the steady-state operating range. The most common implementation is to use split range control with a single controller. However, this approach has some limitations, discussed in previous chapters. In this chapter, we use multiple controllers with different setpoints and demonstrate that this structure can be optimal in some cases when the cost of the input can be traded off against the penalty of deviating from the desired setpoint. We describe a procedure to find the optimal setpoint deviations. We illustrate our procedure in a case in which three inputs (cooling and two sources of heating) are used to control the temperature of a room with a PID-based control structure and without the need of online optimization.

## 7.1 Introduction

The use of more than one input for one output to extend the steady-state range of the output has been a common practice for more than 75 years (e.g. Eckman (1945); Young (1955))<sup>1</sup>. Split range control is the classical control structure commonly used for this. However, using a single controller has some limitations with respect to

---

<sup>1</sup>In this chapter, we use *input* ( $u$ ) as a synonym of *manipulated variable* (MV) and *output* ( $y$ ) as a synonym of *controlled variable* (CV).

tuning. For example, for split range with PI control, the integral times must be the same for all inputs. This is addressed in Chapter 6.

An alternative to extend the steady-state range of the output is to use *one controller for each input* with independent tunings and different setpoints. This structure is often regarded as "sub-optimal" because the setpoints must be different to avoid undesired switching of the controllers. In this chapter, we argue that having different setpoints can be optimal in some cases because it allows us to consider the trade-off between the cost of using the input against the cost of deviating from the desired setpoint. For example, for room temperature control we may use different setpoints in the winter than in the summer to save on heating and cooling, respectively.

This chapter is organized as follows. In Section 7.2 we describe the classical control structures used to extend the steady-state range and maintain control of the output when there is more than one available input. In Section 7.3 we introduce our proposed procedure to obtain optimal setpoints. In Section 7.4 we implement our proposed procedure to find optimal setpoints for controlling room temperature. In Section 7.5, we discuss the validity and the applicability of our method with objective functions different than the one in Sections 7.3 and 7.4. We give some final remarks in Section 7.6.

## 7.2 Classical advanced control structures for more than one input for one output

When we need more than one input ( $u_i$ , manipulated variable, MV) to cover the whole steady-state range for one output ( $y$ , controlled variable, CV), we can use three alternative classical control structures:

1. Split range control
2. Input (valve) position control
3. One controller for each input, each with a different set point for the output

*Split range control* has been in use for more than 75 years (Eckman, 1945; Fink, 1945), and it is still commonly implemented in industry (Sun et al., 2015). Fig. 7.1 shows the block diagram of a split range controller (SRC) with two inputs ( $u_1$  and  $u_2$ ) for one output ( $y$ ). Here, there is a common controller (C) that produces an internal signal in deviation variables ( $v$ ) that is the input to the split range (SR) block, which calculates the values for  $u_i$  (in physical variables).



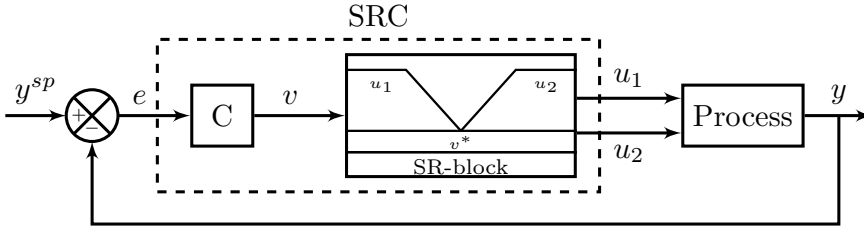


Figure 7.1: Classical split range control (SRC) with two inputs and one output. A typical split range block (SR-block) is shown in Fig. 7.2.

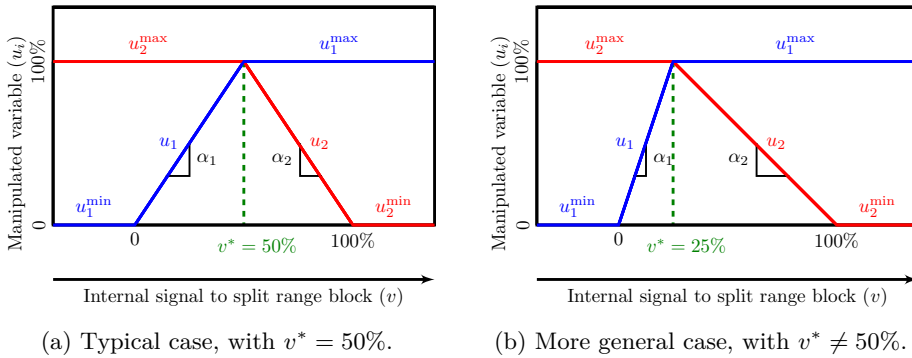


Figure 7.2: Split range block (SR-block) in Fig. 7.1 for a case in which the two inputs have different steady-state gain.

Fig. 7.2 shows a typical split range block. When the internal control signal ( $v$ ) is below the split value ( $v^*$ ),  $u_1$  is used to control  $y$ , while  $u_2$  is saturated; when  $v$  is above  $v^*$ ,  $u_1$  is used to control  $y$ . The split point ( $v^*$ ), or equivalently, the corresponding slopes ( $\alpha_i$ ) in Fig. 7.2, can be used as degrees of freedom to counteract the differences in the effects of the various inputs ( $u_i$ ). The approach introduced in Reyes-Lúa et al. (2019b) (Chapter 5) considers not only the static effect, but also the dynamics. Nevertheless, there are limitations in terms of tuning, as only the controller gains can be adjusted using the slopes; for example, the integral bias needs to be the same for all inputs.

Alternative 2, shown in Fig. 7.3, is *input (valve) position control* (Shinkskey, 1978, 1981). Valve position control is often used to improve the dynamic performance by allowing  $u_1$  to take care of the fast control and  $u_2$  of the long-term control. However, if implemented as shown in Fig. 7.3, it extends the steady-state range. In this scheme, the primary input ( $u_1$ ) always controls the output ( $y$ ). If  $u_1$  approaches its limit ( $u_1^{lim}$ ), then  $u_2$  is used to control  $u_1$  at a setpoint  $u_1^{sp} = u_1^{lim} + \Delta u_1$ , preventing  $u_1$  from saturating. Note that  $u_2$  is only controlling  $u_1$  when needed, so normally it

will be kept at its desired value. We need to have a back-off,  $\Delta u_1 \neq 0$  from the limit ( $u_1^{lim}$ ) to ensure that  $u_1$  always has some range to control  $y$ . Thus, one cannot utilize the full steady-state range of  $u_1$  with this scheme.

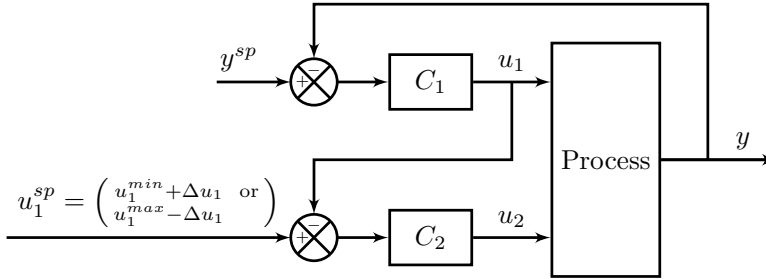


Figure 7.3: Block diagram for input (valve) positioning control for extending the steady-state range.

Finally, Fig. 7.4 shows alternative 3, studied in this chapter, with *one controller for each input*. In Fig. 7.4, the setpoint for the controller using  $u_1$  ( $C_1$ ) is  $y^{sp,1}$ , and the setpoint for the controller using  $u_2$  ( $C_2$ ) is  $y^{sp,1} + \Delta y^{sp}$ . Here,  $\Delta y^{sp}$  should be large enough to guarantee that only one controller is active at a given time, while the other inputs are at their limits (Smith, 2010).

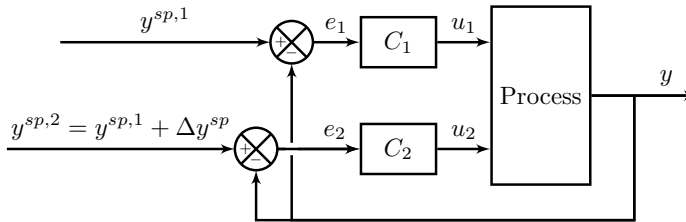


Figure 7.4: Block diagram for two controllers with different setpoints. This is the control structure studied in this chapter.

In Figs. 7.1, 7.3 and 7.4 we show the case with two inputs ( $u_1$  and  $u_2$ ), but all three alternatives are easily generalized to any number of inputs. For all three alternatives, the idea is that only one input ( $u_i$ ) is controlling the output ( $y$ ) at a time. In Fig. 7.1 this is achieved by the split range block. In Fig. 7.3, input  $u_2$  is only used when  $u_1$  reaches its limit. In Fig. 7.4, this is achieved by having sufficiently different setpoints.

In this chapter, we study in detail *one controller for each input* (Fig. 7.4) and we compare this structure with *split range control* (Fig. 7.1).

### 7.3 Optimal setpoint for each input

In this section we consider the cases when there is a trade-off between the cost of input usage ( $u_i$ ) and the cost of deviation from the setpoint ( $\Delta y^{sp}$ ).

As only one input is being used at a time, the cost function (economic objective function) can be written as

$$J(u_k, \Delta y^{sp}), \quad (7.1)$$

where  $u_k$  is the input usage for the active input and  $\Delta y^{sp}$  is the deviation from the desired setpoint. We assume here that the cost is linear in  $u$  and we assume a quadratic penalty for the setpoint deviation. Then the cost function, which we want to minimize, becomes

$$J = p_{u_k} u_k + p_y (y - y^{sp})^2 + c \quad (7.2)$$

where  $p_{u_k}$  is the price for input usage,  $p_y$  represents the price for deviating from the desired setpoint, and  $c$  represents the cost related to keeping the other inputs ( $u_i, \forall i, i \neq k$ ) at their maximum or minimum values (not used to control  $y$ ).

The output ( $y$ ) is a function of the inputs ( $u$ ). We consider the steady-state when we have

$$y = f(u). \quad (7.3)$$

If we consider the case where the relationship in Eq. (7.3) is linear for all inputs ( $u_i$ ), we then have that all inputs can be written as a linear function of  $y$ . Thus,

$$u_i = k_i y + u_{i,0}. \quad (7.4)$$

The cost, when using  $u_k$  as the input, then becomes

$$J = p_{u_k} k_k y + p_y (y - y^{sp})^2 + c_k + p_{u_k} u_{k,0} \quad (7.5)$$

The optimal value of  $y$ , which minimizes the cost  $J$  when using input  $u_k$  is then given by

$$\frac{dJ}{dy} = 0 \quad \Rightarrow \quad p_{u_k} k_k + 2p_y (y - y_{des}^{sp}) = 0 \quad (7.6)$$

We find that the optimal setpoint deviation is

$$\Delta y^{sp*} = y^* - y^{sp} = -\frac{p_{u_k} k_k}{2p_y} \quad (7.7)$$

Thus, in this case, it is optimal with a constant setpoint deviation, independent of any other disturbances. Of course, this will not be the case if we have a different cost function than Eq. (7.2) or a model which is not linear like Eq. (7.4).

An example of a problem that satisfies our assumptions of a linear model is the heating or cooling of a room. The energy balance is

$$0 = \alpha(T^{amb} - T) + Q_h - Q_c \quad (7.8)$$

$y = T$  is the room temperature,  $Q_h$  represents the net heating and  $Q_c$  the net cooling. The term  $\alpha(T^{amb} - T)$  represents the net heat loss to the environment. Eq. (7.8) can be written on the form in Eq. (7.4) with  $y = T$ ,  $u_0 = \alpha T^{amb}$  and

$$k = \begin{cases} \alpha & \text{if } u_k = Q_h \quad (\text{heating}) \\ -\alpha & \text{if } u_k = Q_c \quad (\text{cooling}) \end{cases} \quad (7.9)$$

In general, the optimal setpoint deviation will not be independent of disturbances, as it is in Eq. (7.7). It only holds when Eq. (7.3) and (7.4) are valid.

## 7.4 Case study

Here we will analyze temperature control for the room in Fig. 7.5, which can be described by Eq. 7.8. The detailed model and the parameters are found in Appendix E. The desired (ideal) temperature in the room is  $T^{sp} = 21^\circ C$ . The main disturbance is ambient temperature ( $d = T^{amb}$ ) and there are three available manipulated variables ( $u_i$ ):

- $u_1 = Q_{AC}$ : cooling using air conditioning
- $u_2 = Q_{HW}$ : hot water, through floor heating ( $Q_{FL}$ )
- $u_3 = Q_{EH}$ : electrical heating

We select the nominal operating point as  $T = T^{amb} = 21^\circ C$ . We use air conditioning ( $u_1 = Q_{AC}$ ) to lower the temperature when  $T^{amb} > 21^\circ C$ . When  $T^{amb} < 21^\circ C$  and the room requires heating, we first use hot water ( $u_2 = Q_{HW}$ ), and when it reaches its maximum, we use electric heating ( $u_3 = Q_{EH}$ ). Therefore:

$$\begin{aligned} u &= [Q_{AC}, Q_{HW}, Q_{EH}]^T \\ d &= T^{amb} \\ y &= T \end{aligned}$$

The nominal values and ranges for the inputs ( $u_i$ ) are shown in Table 7.1.

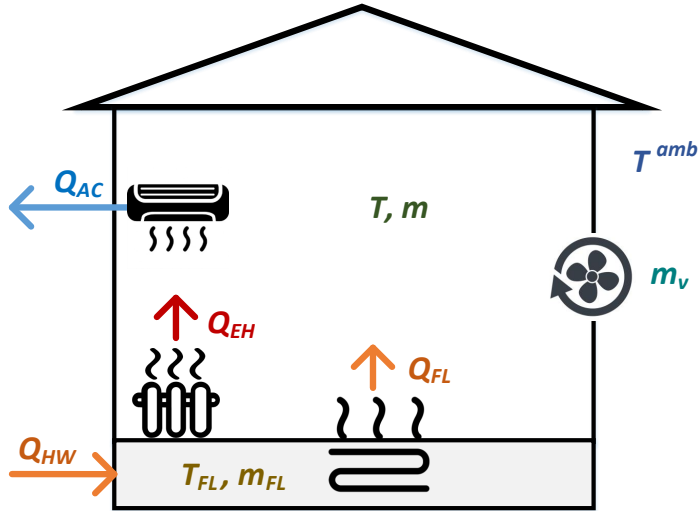


Figure 7.5: Room with three available (independent) inputs ( $u_i = Q_i$ ) for temperature control ( $y = T$ ).

Table 7.1: Ranges for available inputs ( $u_i$ ).

Input ( $u_i$ )	Description	Nominal	Min	Max	Units
$u_1 = Q_{AC}$	air conditioning	0	0	4.5	$kW$
$u_2 = Q_{HW}$	heating water	0	0	3.0	$kW$
$u_3 = Q_{EH}$	electrical heating	0	0	4.0	$kW$

### 7.4.1 Optimal operation for temperature control

We define a scalar cost function which takes into account the cost of energy as well as a quadratic penalty cost for deviating from the temperature setpoint.

$$J = \underbrace{p_{AC} Q_{AC}}_{p_1 u_1} + \underbrace{p_{HW} Q_{HW}}_{p_2 u_2} + \underbrace{p_{EH} Q_{EH}}_{p_3 u_3} + \underbrace{p_T (T - T^{sp})^2}_{p_y (y - y^{sp})^2} \quad [\$ / s] \quad (7.11)$$

where  $p_{EH}$ ,  $p_{HW}$ , and  $p_{AC}$  are the energy prices for electric heating ( $Q_{EH}$ ), heating water ( $Q_{HW}$ ), and air conditioning ( $Q_{AC}$ ).  $p_T$  is a "comfort" penalty for the deviation of the actual room temperature ( $T$ ) from the desired room temperature ( $T^{sp}$ ). The values for these prices are in Table 7.2. Note that Eq. (7.11) has the same form as Eq. (7.5) when only one input is active.

Table 7.2: Parameters for cost function for optimization.

Parameter	Description	Price
$p_{EH}$	price for electrical heating	1.20 \$/kWh
$p_{HW}$	price for heating water	0.80 \$/kWh
$p_{AC}$	price for air conditioning	0.40 \$/kWh
$p_T$	comfort penalty	0.24 \$/°C <sup>2</sup> h

With the prices in Table 7.2 one hour of use of maximum heating water (3 kW) and maximum electricity (4 kW) costs

$$(1.2 \text{ \$/kWh})(4 \text{ kWh}) + (0.8 \text{ \$/kWh})(3 \text{ kWh}) = 7.2 \text{ \$}$$

whereas one hour with a 1°C deviation costs 0.24 \$.

#### 7.4.2 Optimal setpoints for room temperature

We want to find the optimum steady-state value for the room temperature, considering economics and deviation from the desired room temperature (Eq. (7.11)). To this end, we analyze the effect of varying the temperature setpoint when we use different inputs ( $T_i^{sp}$ ) on the economic optimum of the system. At steady-state, the energy balance for the room becomes:

$$0 = \alpha(T^{amb} - T) + Q_{HW} + Q_{EH} - Q_{AC} \text{ [W]} \quad (7.12)$$

Eq. (7.12) is similar to Eq. (7.8), with more inputs.

For illustration purposes, we consider the case when  $Q_{AC}$  is the active input, while  $Q_{HW} = Q_{HW}^{min}$  and  $Q_{EH} = Q_{EH}^{min}$ . Then, Eq. (7.12) becomes:

$$0 = \alpha(T^{amb} - T) + Q_{HW}^{min} + Q_{EH}^{min} - Q_{AC} \quad (7.13)$$

Note that with  $Q_{HW}^{min} = 0$  and  $Q_{EH}^{min} = 0$ , the steady-state room temperature is

$$T = T^{amb} - \frac{Q_{AC}}{\alpha} \quad (7.14)$$

Considering Eq. (7.11) and Eq. (7.13).

$$\left. \frac{dJ}{dT} \right|_{Q_{EH}, Q_{HW}} = -\alpha p_{AC} + 2p_T(T - T^{sp}) \quad (7.15)$$

We find the optimal temperature from Eq. (7.15),  $\frac{dJ}{dT}|_{Q_{EH}, Q_{HW}} = 0$ , and we choose this as our setpoint when we use air conditioning (AC).

$$T_{AC}^{sp} = T^{sp} + \frac{\alpha p_{AC}}{2p_T} \quad (7.16)$$

This same analysis is valid for the case in which  $Q_{HW}$  or  $Q_{EH}$  are the active inputs. This result corresponds to Eq. (7.7) and Eq. (7.9). Thus, the optimal setpoint deviations, when using only one input at a time are:

$$\Delta y^{sp,1} = T_{AC}^{sp} - T^{sp} = +\frac{\alpha p_{ac}}{2p_T} \quad (7.17a)$$

$$\Delta y^{sp,2} = T_{HW}^{sp} - T^{sp} = -\frac{\alpha p_{hw}}{2p_T} \quad (7.17b)$$

$$\Delta y^{sp,3} = T_{EH}^{sp} - T^{sp} = -\frac{\alpha p_{el}}{2p_T} \quad (7.17c)$$

With  $p_T > 0$ , the deviation of  $T$  from  $T^{sp}$  is always penalized. If the comfort penalty ( $p_T$ ) is very high,  $T_i^{sp} \approx T^{sp}$  in equations (7.17c), (7.17b) and (7.17a).

For example, consider  $i = 1$  (AC). Then, from Eq. (7.17a) and with the prices from Table 7.2 and data from Table E.1

$$\Delta y^{sp,1} = \Delta T_{AC}^{sp} = \frac{400 \frac{W}{^\circ C} 0.04 \frac{\$}{kWh}}{(2)(0.24 \frac{\$}{^\circ C^2 h})} = 0.33^\circ C$$

The results for all the inputs are in Table 7.3. The results are also shown graphically in Fig. 7.6 as a function of  $p_i/p_T$  for the case with  $\alpha = 400 W/^\circ C$ .

Table 7.3: Optimal deviations for the three available inputs.

Input	$\Delta y^{sp,i} [^\circ C]$
$u_1 = Q_{AC}$	+0.33
$u_2 = Q_{HW}$	-0.67
$u_3 = Q_{EH}$	-1.00

### 7.4.3 Three controllers with different setpoints

We can implement the results in Section 7.4.2 using a controller for each input each with a different setpoint, as shown in Fig. 7.7, with  $y = T$  and  $u_i = Q_i$ . The tuning procedure for the PI controllers is described in Appendix E.2.

Fig. 7.8 shows the simulation results using large steps in  $d = T^{amb}$  to show the performance of the control structure in the whole range. All controllers have anti-windup (clamping) implemented. We use the optimal setpoints in Table 7.3.

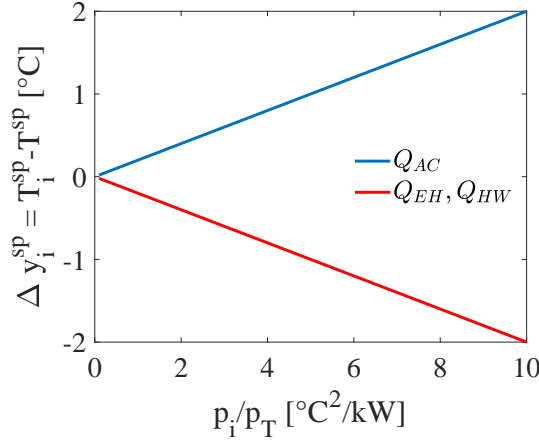


Figure 7.6: Effect of comfort penalty ( $p_T$ ) and input usage ( $p_i$ ) on optimal setpoint deviation ( $\Delta y^{sp,i}$ ).

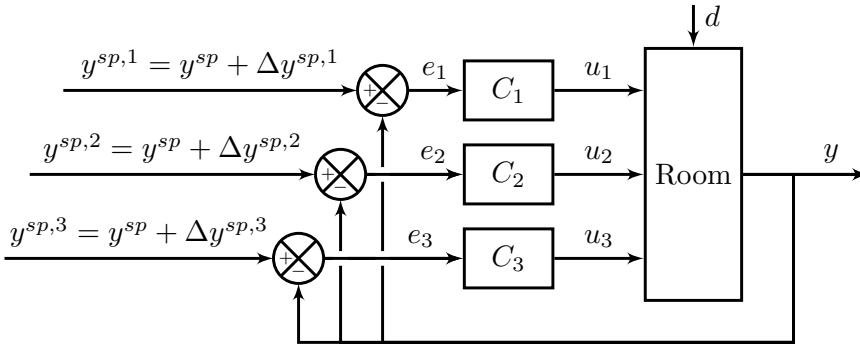


Figure 7.7: Block diagram for three controllers with different setpoints, one for each input;  $y = T$ ,  $d = T^{amb}$  and  $u_i = Q_i$ , where 1 = AC, 2 = HW, 3 = EH.

The simulation starts at the nominal point, with  $T = T^{amb} = 21^\circ C$ . At  $t = 0.5$  h,  $T^{amb}$  increases by  $+10^\circ C$ , and we need air conditioning ( $Q_{AC}$ ) to cool down the room. We observe that  $T$  reaches  $T_{AC}^{sp} = 21.33^\circ C$  at steady-state. At  $t = 3.5$  h,  $T^{amb}$  is decreased by  $-5^\circ C$  to  $20^\circ C$ , and we keep using  $Q_{AC}$  as input, reaching again  $T_{AC}^{sp} = 21.33^\circ C$  at steady-state. Then, at  $t = 7$  h,  $T^{amb}$  is decreased by  $-8^\circ C$  to  $18^\circ C$ , and we now use  $Q_{HW}$  as input and we reach  $T = T_{HW}^{sp} = 20.33^\circ C$  at steady-state. At  $t = 10$  h,  $T^{amb}$  is further decreased by  $-13^\circ C$  to  $5^\circ C$ .  $Q_{HW}$  reaches  $Q_{HW}^{max}$ , such that  $Q_{EH}$  becomes the active input and  $T = T_{EH}^{sp} = 20.33^\circ C$  at steady-state.



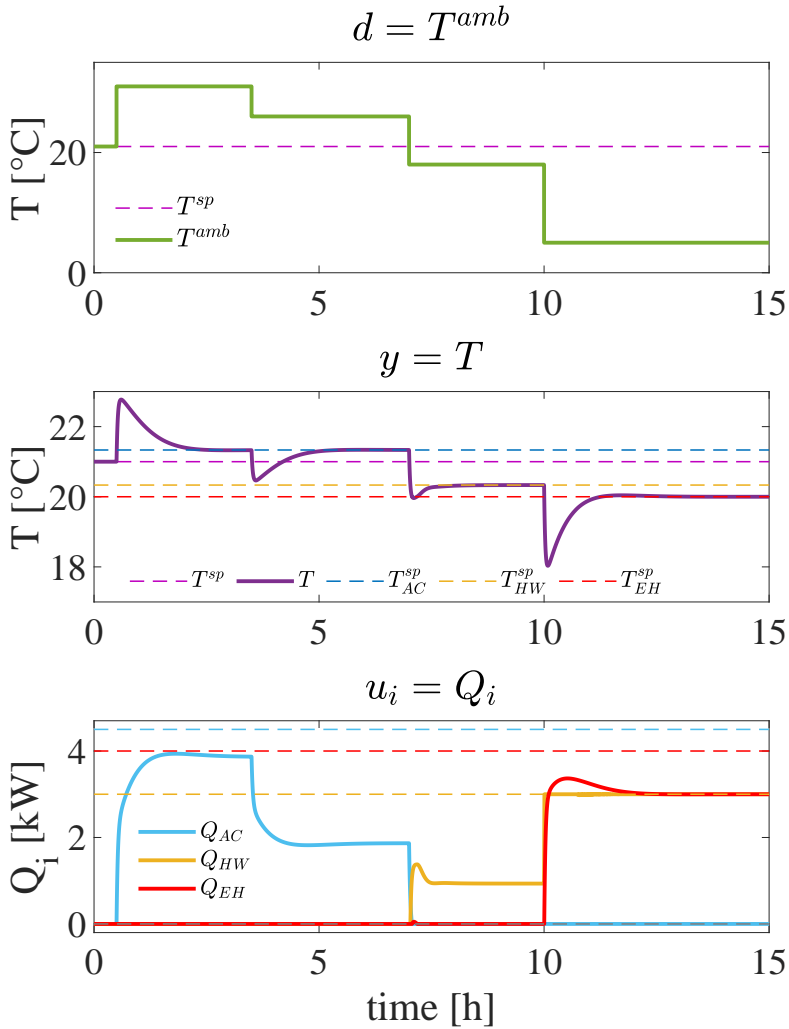


Figure 7.8: Simulation results using three different controllers (Fig.7.7), with the optimal setpoint deviations in Table 7.3.

#### 7.4.4 Comparison with split range control

We implement a classical split range controller as shown in Fig. 7.9. We use the procedure proposed by Reyes-Lúa et al. (2019b) (Chapter 5) to find the tuning parameters for the common controller and the slopes in the split range block, which are available in Appendix E.3.

The common PI-controller has a proportional gain  $K_C = 0.1277 \text{ kW}/^\circ\text{C}$  and integral time constant  $\tau_I = 1200 \text{ s}$ . For all inputs, the setpoint is always fixed at  $y^{sp} = T^{sp} = 21^\circ\text{C}$ , which would correspond to having a huge penalty for setpoint deviation ( $p_T \rightarrow \infty$ ).

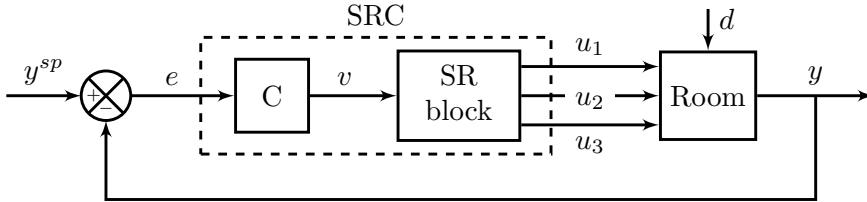


Figure 7.9: Block diagram for split range control (SRC) with three inputs and one output. The SR block is shown in Fig. E.1;  $y = T$ ,  $d = T^{amb}$  and  $u_i = Q_i$ , where 1 = AC, 2 = HW, 3 = EH.

Fig. 7.10 compares the results of split range control with the previous simulation using three controllers with different setpoints. The changes in  $T^{amb}$  are the same as in Fig. 7.8. We observe that, as expected, that the input (energy) usage is higher with split range control as it has a fixed setpoint.

Fig. 7.11 shows the accumulated  $J_{energy}$  with both control structures. At the end of the simulation period,  $J_{energy}^{tot} = 43.15 \text{ [\$]}$  with a constant setpoint policy (split range control), and  $J^{tot} = 39.84 \text{ [\$]}$  when using optimal setpoints. This corresponds to saving 7.66% by slightly modifying  $T^{sp}$ .

## 7.5 Discussion

The optimal ventilation rate ( $\dot{m}_v$ ) may be obtained considering outdoor air quality and indoor air quality requirements. Finding appropriate models to define the optimal ventilation rate is an ongoing area or research (Wang, 2010; Ganesh et al., 2019).

In our case study,  $\dot{m}_v$  is constant, considering typical ventilation rates (Osborn, 1985). As it directly affects indoor temperature, it could also be used as an additional input to extend the control range. The usage of  $\dot{m}_v$  can thus be included in the objective function  $J(u, d)$  as the additional term:

$$p_v(\dot{m}_v^{sp} - \dot{m}_v)^2 \quad (7.18)$$

where  $p_v$  is the penalization for deviating from the desired ventilation rate ( $\dot{m}_v^{sp}$ ). The optimal setpoint for room temperature, using  $\dot{m}_v$  as input, can be found with the procedure described in this work.

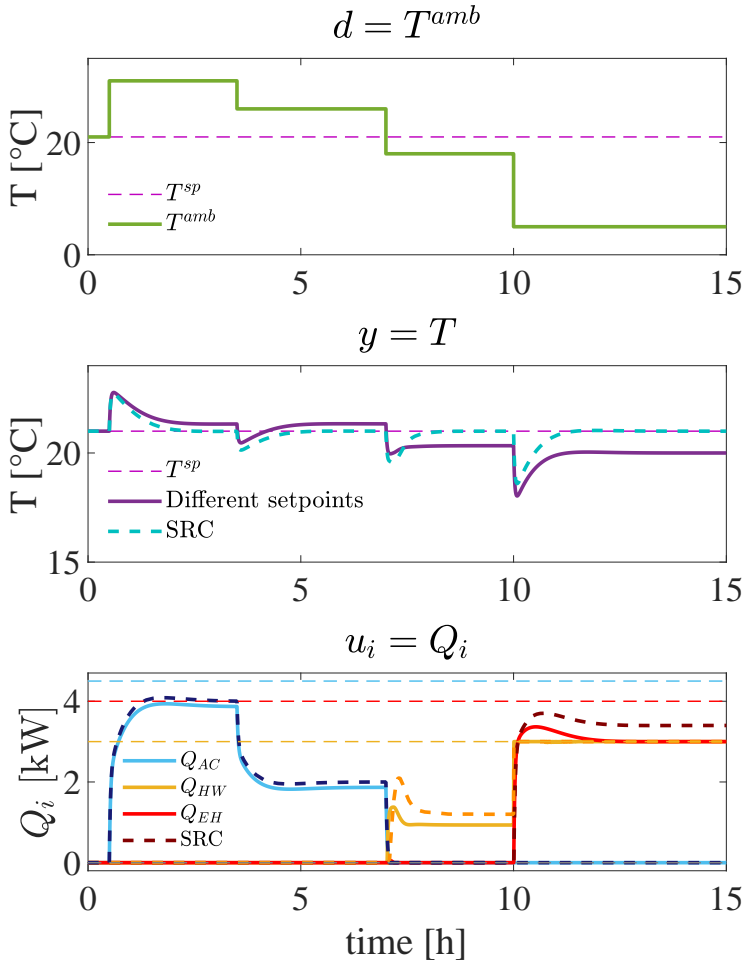


Figure 7.10: Comparison of simulation results with three controllers with different setpoints (Fig. 7.7) and split range control (SRC, Fig. 7.9). The simulation with three controllers is shown with solid lines and the simulation with SRC is shown with dashed lines.

With this approach  $\dot{m}_v$  may be used simultaneously with the rest of the inputs ( $Q_i$ ). For example, if the ambient air temperature is higher than the temperature inside the room ( $T^{amb} > T$ ), the ventilation should be at its minimum position to introduce only the necessary fresh air and reduce the energy consumed by air conditioning (Wang, 2010). Thus,  $\dot{m}_v$  and  $Q_{AC}$  would be used at the same time, and there would always be a deviation from  $\dot{m}_v^{sp}$ .

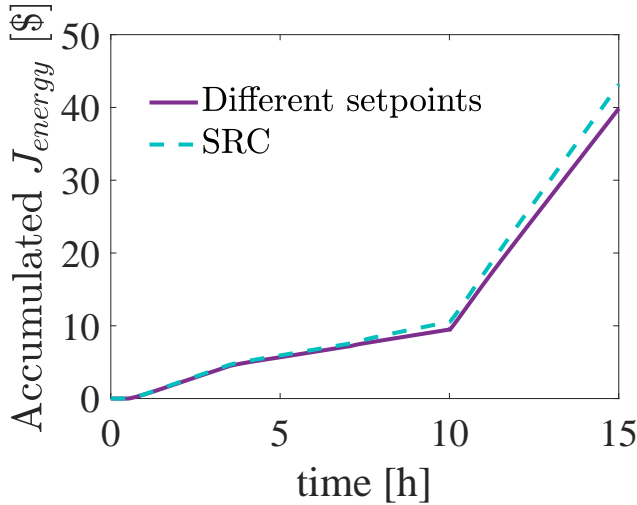


Figure 7.11: Comparison of accumulated cost of energy ( $J_{energy}$ ) using three controllers with different setpoints (Fig. 7.7) and split range control (SRC, Fig. 7.9).

Alternatively, we can implement a two-step solution. In the first step, we use the procedure illustrated in the case study, only using  $Q_i$  (cooling or heating) as inputs, and keeping the ventilation at its desired setpoint,  $\dot{m}_v = \dot{m}_v^{sp}$ . Once that every  $\dot{Q}_i$  is saturated, we proceed to the second step, in which we use the objective function becomes

$$J = k_v(\dot{m}_v^{sp} - \dot{m}_v)^2 + k_T(T - T^{sp})^2 \quad (7.19)$$

Substituting  $\dot{m}_v$  with the steady-state mass balance, we can derive  $T^{opt}$  and construct a plot similar to the one in Fig. 7.6, with  $\Delta T = (\Delta T^{sp})$  as function of  $p_v/p_T$  and  $\dot{m}_v^{sp}$ .

A third option would be to implement control structure in which we combine input (valve) positioning control with either split range control or multiple controllers. In that case, we could use ventilation as the secondary input ( $u_2$  in Fig. 7.3) to prevent  $Q_{AC}$  or  $Q_{EH}$  from saturating<sup>2</sup>. In this case, "air quality" requirements may not be always satisfied.

## 7.6 Concluding remarks

We proposed a procedure to find optimal setpoints when there is more than one available input for one output. This setpoints can be used to achieve optimal steady-state operation using multiple (PID) controllers, one for each input.

<sup>2</sup>This solution is similar to the control structure in Fig. 3.4 in Chapter 3.

The results are valid for problems that can be described with a linear model and in which there is a trade-off between a linear cost for input usage and a quadratic penalty for setpoint deviation.

Using our results, we found optimal setpoints for the control of room temperature using three available inputs. In a simulation case study, we demonstrated that optimal steady-state operation, considering economics and deviation from the desired value, can be reached by using one PI controller for each input, each with a different setpoint. Comparing this implementation with a constant setpoint policy (classical split range control), we obtained a reduction in the energy cost of 7.66% with only a small setpoint deviation. The benefit of this approach is that optimal steady-state operation can be achieved with negligible computational cost and using PID-control. The ideas discussed in this chapter can also be applied to other similar problems and using different types of controllers.



# Chapter 8

## Improved PI control for a surge tank satisfying level constraints

This chapter is published as:

Reyes-Lúa, A., Backi, C. J., and Skogestad, S. (2018a). Improved PI control for a surge tank satisfying level constraints. In *3rd IFAC Conference on Advances in Proportional-Integral-Derivative Control (PID18)*, volume 51, pages 835–840, Ghent, Belgium. IFAC-Papers OnLine

This chapter considers the case of averaging level control, where the main objective is to reduce flow variations by using varying liquid levels. However, to avoid overfilling or emptying the tank, the liquid level needs to satisfy safety-related constraints. In the simplest case, a P-controller can be used, but may not give acceptable averaging of the flow, especially if the surge tank is relatively small. In addition, the P-controller does not allow the level setpoint to be adjusted.

We propose a simple scheme with a PI-controller for normal operation and two high-gain P-controllers to avoid the liquid level constraints. This type of structure allows for an improved control of non-linear processes by utilizing conventional P and PI controllers in a "split-range scheme". We then compare our proposed structure with a benchmark MPC strategy. We demonstrate that the proposed method has similar performance, but with less modeling effort, less computational time and simpler tuning.

### 8.1 Introduction

Liquid level control can have two purposes (Shinskey, 1988; Faanes and Skogestad, 2003): to tightly control the level (setpoint tracking) or to dampen flow disturbances. The latter, where the tank acts as a surge tank, is also known as averaging level control and is the focus in this chapter. The controller tuning for the two cases are completely different, because for tight level control we need a high controller gain,

whereas for averaging level control we want a low controller gain. For a surge tank, the actual value of the level may not be important as long as it is kept within its allowable safety limits (Shinskey, 1988; Åström and Hägglund, 1995); that is, to avoid overfilling or emptying the tank.

Fields of applications for setpoint tracking and safety control for levels in tanks are as diverse as drum boilers in power plants, where both, dry-running and complete filling should be avoided (Åström and Bell, 2000), gravity separators in the mining as well as the oil- and gas industry, where setpoint tracking and avoidance of complete filling are the most important control tasks (Backi and Skogestad, 2017), and wastewater sumps in the chemical industry and surge tanks (Åström and Hägglund, 2001). Especially for the latter, minimization of the change in the outflow is highly desired, since the incoming surge should be distributed further with reduced amplitude. In recent years, not only PI(D) controllers were designed for level control of tanks, but also fuzzy control approaches (Tani et al., 1996; Petrov et al., 2002), as well as optimal averaging strategies (McDonald et al., 1986; Campo and Morari, 1989; Rosander et al., 2012).

In this work, we propose a PI-based control structure that efficiently allows for setpoint tracking with low usage of the manipulated variable (MV) and safety-related constraint satisfaction. Model Predictive Control (MPC) is well known for its capability of following a setpoint while following constraints and limiting rate of change of MVs. For this reason, we compare the performance of the proposed structure with model predictive control (MPC).

The rest of this chapter is structured as follows: Section 8.2 introduces the problem, while the proposed control structure is presented in Section 8.4. Section 8.5 introduces the MPC formulation and simulation results are presented in Section 8.6. A performance comparison is shown Section 8.7, while the chapter is concluded in Section 8.8.

## 8.2 Problem formulation

The control task is to dampen flow disturbances in a simple tank system, modeled with the following differential equation

$$\frac{dh}{dt} = \frac{1}{a} (q_{in} - q_{out}), \quad (8.1)$$

where  $h$  is the level (controlled variable - CV),  $a$  denotes the cross-sectional area of the liquid (here  $a = 1 \text{ m}^2$ ),  $q_{in}$  denotes the volumetric inflow (disturbance variable - DV), and  $q_{out}$  is the volumetric outflow. The nominal residence of the tank is  $\tau = V/q = 1 \text{ m}^3/0.5 \text{ m}^3 \text{ min}^{-1} = 2 \text{ min}$ .



We assume that we have implemented a lower-layer flow controller so that  $q_{out}$  is the MV<sup>1</sup>. The inflow and outflow are assumed to be limited within  $q_{min} \leq q \leq q_{max}$ . With  $q_{min} = 0 \text{ m}^3 \text{ min}^{-1}$  and  $q_{max} = 1 \text{ m}^3 \text{ min}^{-1}$  The tank is at its maximum level when  $h = 1 \text{ m}$ , while an empty tank corresponds to  $h = 0 \text{ m}$ . Actually, to be on the safe side, the level should stay within  $0.1 \leq h \leq 0.9 \text{ m}$ . So  $h_{min} = 0.1 \text{ m}$  and  $h_{max} = 0.9 \text{ m}$ . These limits are shown by the yellow dotted lines in the figures.

Two types of inflow disturbances are assumed to act upon the process (Eq.(8.1)); namely, step-changes and sinusoidal variations. The period of the sinusoids is 6.28 min which is quite long compared to the nominal residence time of 2 min, which means that it will be difficult to dampen large sinusoidal disturbances without violating the level constraints. Furthermore, the level measurement can be noisy.

### 8.3 Simple controller schemes for surge tanks

For a surge tank, the actual value of the level may not be important as long as it is kept within its allowable safety limits (Shinskey, 1988). Therefore, Shinskey argued that integral action should not be used in some cases, and proposed to use a P-only controller in the form,

$$q_{out} = K_c \cdot h, \quad (8.2)$$

$$K_c = \frac{q_{max}}{h_{max}}. \quad (8.3)$$

This controller gives  $q_{out} = 0$  when  $h = 0$  and  $q_{out} = q_{max}$  when  $h = h_{max}$ . Note that there is no level setpoint. Rewritten in terms of deviation variables there will be a "setpoint", but it has no practical significance as it is not well tracked (Rosander et al., 2012). For averaging level control, where we want a low controller gain, this is the P-controller with the lowest controller gain that satisfies the safety constraints. However, one problem is that the gain  $K_c = q_{max}/h_{max}$  may still be too large when the process is operating at normal conditions, resulting in too large variations in  $q_{out}$  (MV) when there are smaller inflow disturbances.

This has led many authors to consider nonlinear controllers and MPC. The simplest nonlinear controller is a P-controller with a varying gain, that is, the gain is larger when the level approaches its safety limits. A simple implementation is to use three gain values as shown in Figure 8.1. The low gain works as an averaging controller when the flow changes are small (normal operation), and the two high gains track each boundary (Åström and Hägglund, 2006).

---

<sup>1</sup>Here, we assume that we have level control in the direction of flow, so that the inflow is the DV and the outflow is the MV, but in other cases it may be opposite. It will not affect the results in this chapter.

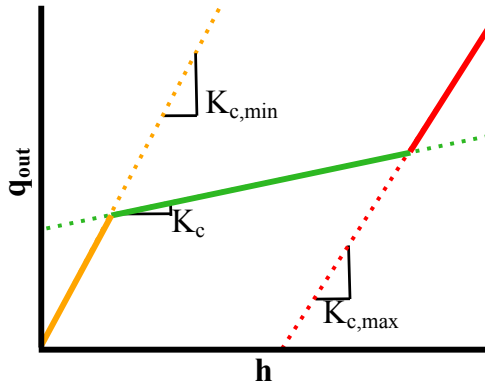


Figure 8.1: Nonlinear relationship (solid lines) between level and outflow for case with three P-controllers.

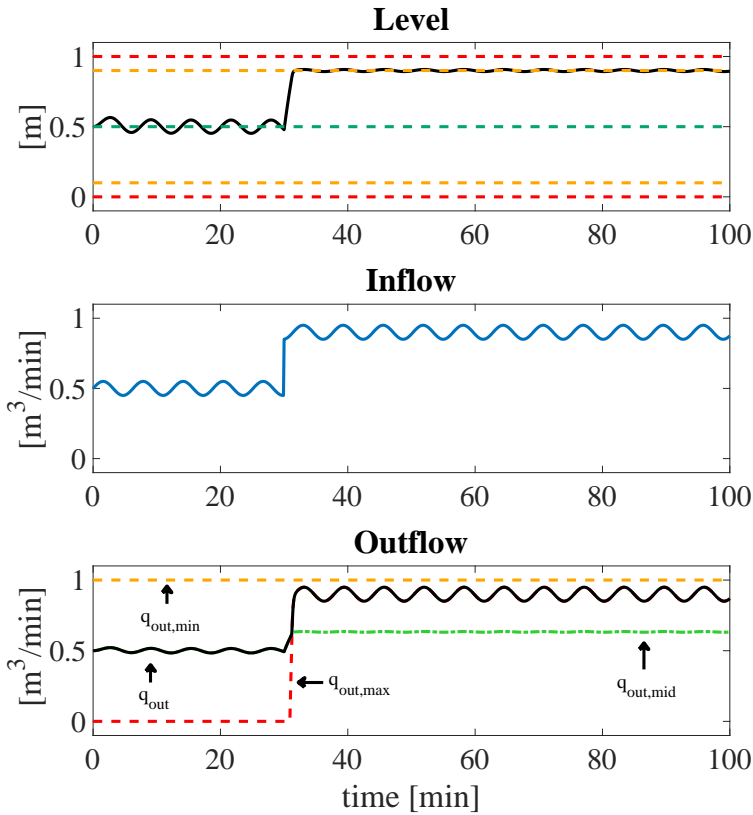


Figure 8.2: Level control with three P-controllers.  $K_c = 0.33$ ,  $K_{c,max} = K_{c,min} = 6.67$ .

During normal operation, inflow disturbances are dampened by the low gain P-controller. Then, when the level approaches the upper limit, the P-controller with high gain takes over, avoiding overflow with a fast response. Similarly, the other high-gain P-controller takes over when the level approaches the lower limit. The scheme may be implemented with three P-controllers and a mid-selector which selects the middle controller output as the MV.

The main drawback of the three P-controller scheme is that the normal range (with low controller gain) can be quite narrow in terms of flow rates, as illustrated in Figure 8.1. Once we get out of the normal range and one of the high-gain P-controllers takes over, it remains controlling the level tightly at the high or low limit and dampening of inflow disturbances is lost, as shown in Fig. 8.2.

Another problem with P-only control is that there is no level setpoint which the operator or a higher-level master controller can manipulate. For example, the operator or master controller may want to set the level temporarily to a low value to prepare the systems for an expected large increase in the inflow, or reduce loss due to disturbances in utilities (Lindholm et al., 2012).

We therefore propose to use a modified three-controller scheme with the slow (normal) P-controller being replaced by a PI-controller, as discussed in the next section. However, before looking at this, let us consider the response with a single linear PI-controller.

Figure 8.3 depicts the response with a slow and a fast PI-controller to step and sinusoidal inflow changes. Both are tuned using the SIMC rules (Skogestad, 2003), in which the tuning parameter,  $\tau_c$ , corresponds to the closed loop time constant. Anti-windup with back-calculation is also implemented.

The fast PI-controller (green lines), with a short closed loop time constant,  $\tau_c = 0.5$  min ( $K_c = 2$ ,  $\tau_I = 2$  min), keeps  $h$  within the safety constraints, but fails to dampen the sinusoidal input during normal operation ( $q_{out} \approx q_{in}$ ). On the other hand, the slow PI-controller (blue lines), with  $\tau_c = 3$  min ( $K_c = 0.33$ ,  $\tau_I = 12$  min), performs well during normal operation, dampening the sinusoidal signal on  $q_{out}$ . However, the response is too slow when  $q_{in}$  has sudden changes close to its limits, and safety and physical constraints on  $h$  are clearly violated.

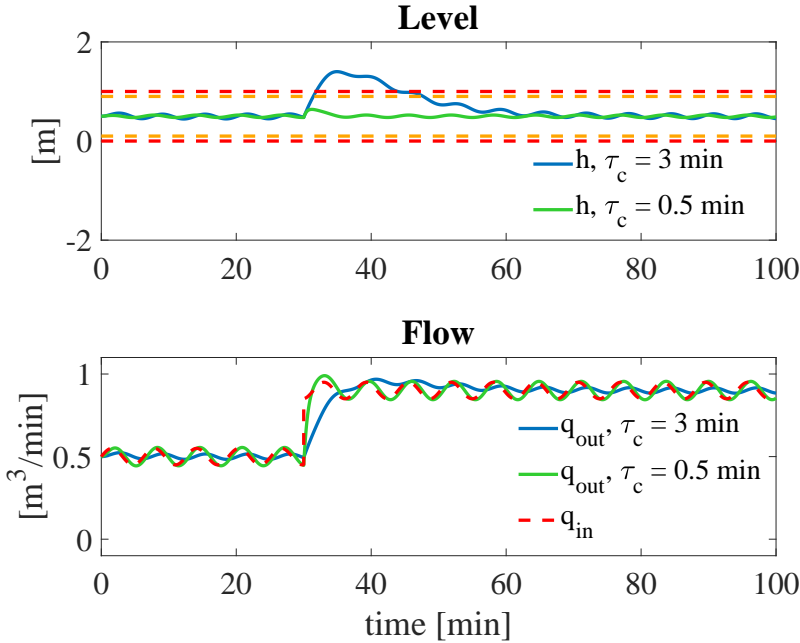


Figure 8.3: Level control with PI-controller.

#### 8.4 Proposed control structure for improved liquid level control

The purpose of this study is to develop a simple, yet efficient control structure for averaging level control based on easy-to-tune P and PI algorithms. We propose a non-linear control scheme in which the selection is done based on the output of three controllers. The overall structure of the proposed controller is demonstrated in Fig. 8.4.

Three different controllers calculate  $q_{out}$  (PIPP control strategy):

- $c_{mid}$ : PI-controller that tracks the actual desired value for the level,  $h_{sp}$ . This is a "slow" controller with a low gain  $K_c$ , designed to dampen the response for disturbances in  $q_{in}$  during normal operation.
- $c_{max}$ : P-controller with a large gain,  $|K_{c,max}| \gg |K_c|$ , which avoids violation of the maximum liquid level.
- $c_{min}$ : P-controller with a large gain,  $|K_{c,min}| \gg |K_c|$ , which avoids violation of the minimum liquid level.

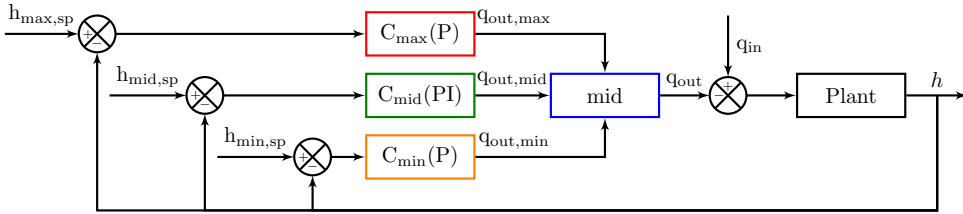


Figure 8.4: Proposed PIPP control structure with one PI-controller and two P-controllers to track safety limits.

The core of the proposed scheme is a mid-selector, based on the output of the three controllers. The proportional parts of the controllers behaves in a similar fashion as the non-linear three P-controllers described in Section 8.3 (Fig. 8.1). During normal operation, the output of the PI-controller,  $q_{out,mid}$ , will be the mid-value. When the level approaches the upper limit, the controller  $c_{max}$  will give an output signal  $q_{out,max} > 0$ , which becomes the middle value, increasing the outflow to avoid overflow. Accordingly, when the level decreases close to the lower limit,  $c_{min}$  will take over, preventing the tank from emptying.

Contrary to the scheme with three P-controllers presented in Section 8.3 (Fig. 8.2), the "slow" PI-controller will always take over after some time due to integral action, which should not be limited by anti windup. It will bring the level back to normal operation and dampen oscillations.

Similar piecewise linear controllers have been proposed previously. Cheung and Luyben (1980) proposed to have a P controller acting within the normal error band and a PI controller coming into action when the error goes outside the normal error band; all controllers with the same gain. They claim that for large disturbances, the integral action would force the error to return within the desired range quickly. However, the level will return to within the band, but not necessarily to the set point. Shunta and Fehervari (1976) described a piecewise linear controller using a combination of P and PI controllers, similar to what we are proposing, with an override system based on high and low selectors.

### 8.4.1 Tuning

To tune the  $c_{mid}$  PI-controller for normal operation,

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

we recommend to use the SIMC tuning rules (Skogestad, 2003), with the following parameters for integral processes:

$$K_c = \frac{1}{k'(\tau_c + \theta)} \text{ and } \tau_I = 4(\tau_c + \theta), \quad (8.4)$$

where  $\theta$  is the process time delay, and  $k'$  is the slope of the integral process ( $\Delta y/(\Delta t \cdot \Delta u)$ ). In our case study  $\theta = 0$  and  $k' = 1$ . The only tuning variable for the PI-controller is the desired closed-loop time constant,  $\tau_c$ , which should be selected long enough to dampen the response for inflow disturbances. Instead of selecting  $\tau_c$ , one can select the controller gain and from this get  $\tau_c$ . As a starting point for the controller gain one may use the value  $K_c = q_{max}/h_{max} \approx 1/1 = 1$  for the slowest single P-controller, see (8.3). Here, we reduce it by a factor 3, because we want to have smaller MV (outflow) variations. Thus, we select  $\tau_c = 3$  min which gives  $K_c = 0.33$ ,  $\tau_I = 12$  min. For the two P-controllers,

$$q_{out,max} = K_{c,max}(h - h_{max,sp}) + q_{out,bias} \quad (8.5a)$$

$$q_{out,min} = K_{c,min}(h - h_{min,sp}) + q_{out,bias} \quad (8.5b)$$

In order to have a wide operation range for the PI-controller (dampening effect), we select a large controller gain for the P-controllers,  $K_{c,max} = K_{c,min} = 20 K_c \approx 6.7$ . We use (8.5) to find  $h_{max,sp}$  and  $h_{min,sp}$ , such that we have a fully open valve ( $q_{out} = q_{out,max} = 1 \text{ m}^3 \text{ min}^{-1}$ ) when the level is at the upper limit ( $h = h_{max} = 0.9 \text{ m}$ ), and a fully closed valve ( $q_{out} = q_{out,min} = 0 \text{ m}^3 \text{ min}^{-1}$ ) when the level is at the lower limit ( $h = h_{min} = 0.1 \text{ m}$ ). We use the nominal value for  $q_{out}$  as the bias,  $q_{out,bias} = 0.5 \text{ m}^3 \text{ min}^{-1}$ . For example,

$$\begin{aligned} h_{max,sp} &= h_{max} - (q_{out,max} - q_{out,bias})/K_{c,max}, \\ h_{max,sp} &= 0.9 \text{ m} - \frac{(1 - 0.5) \text{ m}^3 \text{ min}^{-1}}{6.7 \text{ m}^2 \text{ min}^{-1}} = 0.825 \text{ m} \end{aligned}$$

### 8.4.2 Simulation

Fig. 8.5 shows the response of the proposed PIPP control structure when the process is subject to a sinusoidal disturbance and a large step change. The process starts at steady state, with  $h = h_{sp} = 0.5 \text{ m}$ , which represents normal operation. Hence, the selected MV-signal is  $q_{out,mid}$ , the output of  $c_{mid}$ .

The output of the P-controller  $c_{max}$  is a closed valve,  $q_{out,max} = 0 \text{ m}^3 \text{ min}^{-1}$ , while the output of  $c_{min}$  is a fully open valve, corresponding to  $q_{out,min} = 1 \text{ m}^3 \text{ min}^{-1}$ . At  $t = 30 \text{ min}$ ,  $q_{in}$  increases to an average of  $0.9 \text{ m}^3 \text{ min}^{-1}$ . Then, P-controller  $c_{max}$  takes over as  $q_{out,max}$  increases and becomes the middle value. Eventually, at  $t \approx 55 \text{ min}$ , the output from the PI-controller,  $q_{out,mid}$ , again becomes the middle

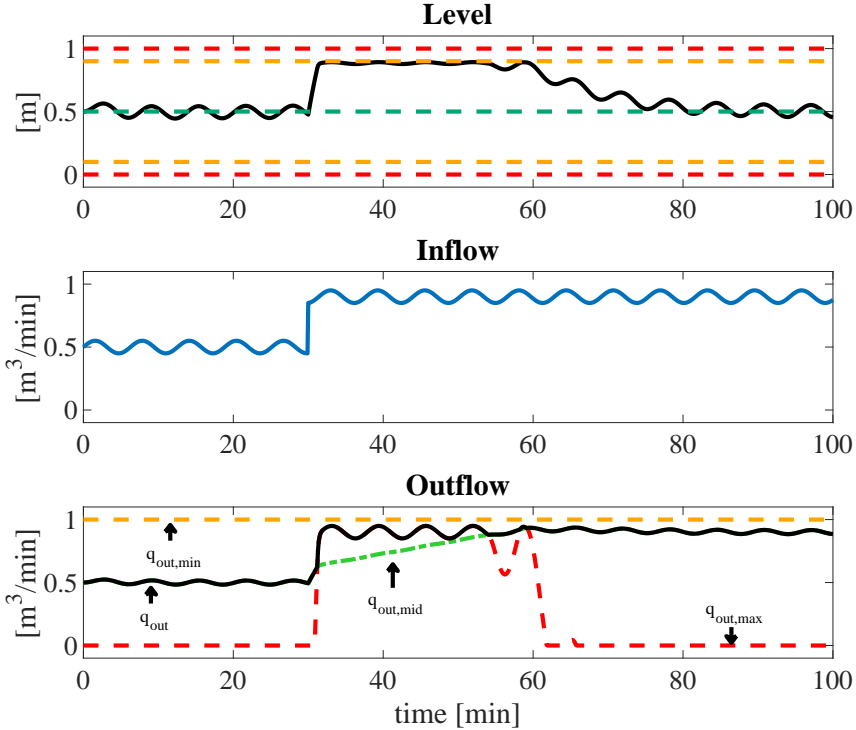


Figure 8.5: Simulation of proposed PIPP control structure.

value and brings the level back to its nominal setpoint. When this happens the variations in the outflow again become much reduced.

## 8.5 MPC Implementation

In order to have a benchmark to compare our simple PIPP scheme, we design a standard MPC controller. The optimal control problem is first discretized into a finite dimensional optimization problem divided into  $N$  elements, which represents the length of the prediction horizon. Hence, each interval is in  $[t_k, t_{k+1}]$  for all  $k \in \{1, \dots, N\}$ , where we use a third order direct collocation Radau scheme for the polynomial approximation of the system trajectories for each time interval  $[t_k, t_{k+1}]$ . The resulting discretized system model is represented as:

$$h_{k+1} = f(h_k, q_{in,k}, q_{out,k}), \quad (8.6)$$

where  $h_k$  represents the differential state from (8.1),  $q_{in,k}$  is the DV (inflow) and  $q_{out,k}$  denotes the MV (outflow), all at time step  $k$ .

Once the system is discretized, the MPC problem can be formulated as:

$$\begin{aligned} \min \quad & \sum_{k=1}^N \omega_1 \|(h_k - h_{sp})\|^2 + \sum_{k=1}^N \omega_2 \|(q_{out,k} - q_{out,k-1})\|^2 \\ \text{s.t.} \quad & (8.6) \tag{8.7a} \\ & h_{min} \leq h_k \leq h_{max} \tag{8.7b} \\ & q_{out,min} \leq q_{out,k} \leq q_{out,max} \tag{8.7c} \\ & h_0 = h_{init} \tag{8.7d} \\ & q_{out,0} = q_{out,init} \tag{8.7e} \end{aligned}$$

with  $h_{min} = 0.1$  m,  $h_{max} = 0.9$  m,  $q_{out,min} = 0$  m<sup>3</sup> min<sup>-1</sup> and  $q_{out,max} = 1$  m<sup>3</sup> min<sup>-1</sup>. The objective function comprises of a term for level setpoint tracking as well as a term penalizing changes in the manipulated variable  $q_{out}$  between time steps  $k - 1$  and  $k$ .

Constraint (8.7a) defines the model dynamics, whereas constraint (8.7b) enforces the level to remain between the bounds,  $h_{min}$  and  $h_{max}$ , respectively. Upper and lower bounds are also enforced for the manipulated variable as  $q_{out,min}$  and  $q_{out,max}$  in (8.7c). We assume that the level is measured. At each iteration, the initial conditions for the states are enforced in (8.7d) and (8.7e).

The dynamic optimization problem is setup as a QP problem in CasADi v3.1.0 (Andersson, 2013), which is then solved using qpOASES (Ferreau et al., 2014). The plant simulator is solved with an ode15s solver. We simulate 2000 MPC iterations with a sample time of  $\Delta t = 0.1$  min. The prediction horizon of the MPC controller is set to 5 min resulting in  $N = 50$  prediction steps.

## 8.6 Comparision of simple PIPP scheme with MPC

In this section we present simulation results for four different cases. For every case, we compare the response of our proposed structure with the aforementioned MPC implementation. The inflow,  $q_{in}$ , is the disturbance and the analyzed cases are:

Case 1: Step changes in  $q_{in}$

Case 2: Step changes in  $q_{in}$  and measurement noise

Case 3: Step changes in sinusoidal  $q_{in}$

Case 4: Step changes in higher frequency sinusoidal  $q_{in}$



In all simulations, the level setpoint is  $h_{sp} = 0.5$  m, and the plant is subject to the same step changes of  $q_{in}$ :  $+0.2$  m at  $t = 50$  min,  $+0.2$  m at  $t = 100$  min, and  $+0.05$  m at  $t = 150$  min, with an initial value of  $q_{in} = 0.5$  m<sup>3</sup> min<sup>-1</sup>. In case 3, the amplitude is 0.05 m<sup>3</sup> min<sup>-1</sup> and the frequency is 1 rad min<sup>-1</sup>. In case 4, the frequency is increased to 2 rad min<sup>-1</sup>.

The parameters for the plant model Eq. (8.1) are  $k' = 1$  and  $\theta = 0$  min. For every case, the SIMC tuning parameter,  $\tau_c$ , was set to 3 min. Then,  $K_c = \frac{1}{3}$ ,  $\tau_I = 12$  min, and  $K_{c,max} = K_{c,min} = 20 K_c$ . The MPC tunings were the same for all the cases with  $\omega_1 = 1$  and  $\omega_2 = 130$ .

### 8.6.1 Case 1: steps in the inflow

As seen in Fig. 8.6, the constraints on the level and the output are satisfied and overall tracking performance is satisfactory for both controllers in this simple tracking case without disturbances or added noise.

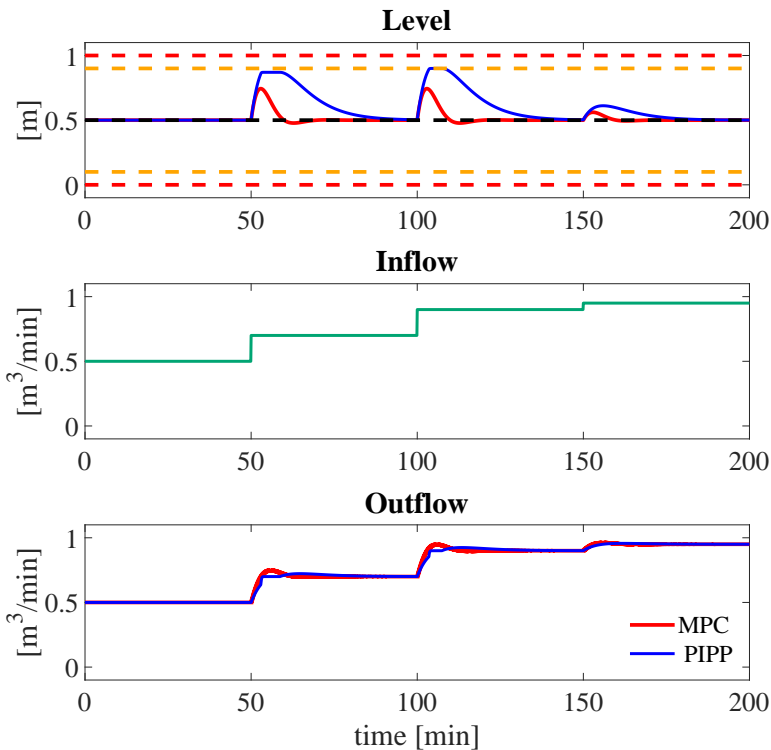


Figure 8.6: Case 1: Response of PIPP structure and MPC with step changes in inflow ( $q_{in}$ ).

Note that in the case of the proposed PIPP controller, the PI-controller effectively dampens the oscillations in the beginning, and  $q_{out} = q_{mid,out}$ . When the disturbance is large and the level approaches the upper limit at  $t \approx 50$  min,  $c_{max}$  takes over and  $q_{out} = q_{max,out}$ . This avoids overflow of the tank. Then, at  $t \approx 60$  min,  $c_{mid}$  takes over again. We observe a similar behavior at  $t \approx 110$  min.

### 8.6.2 Case 2: steps in the inflow plus noisy measurement

Fig. 8.7 shows the effect of the added measurement noise for the level in both controllers. The level can still be maintained around the nominal value  $h_{sp} = 0.5$  m and all constraints are satisfied. A drawback of using a high gain for the P-controllers is that measurement noise is magnified in  $q_{out}$  when the  $h$  is close to the limits.

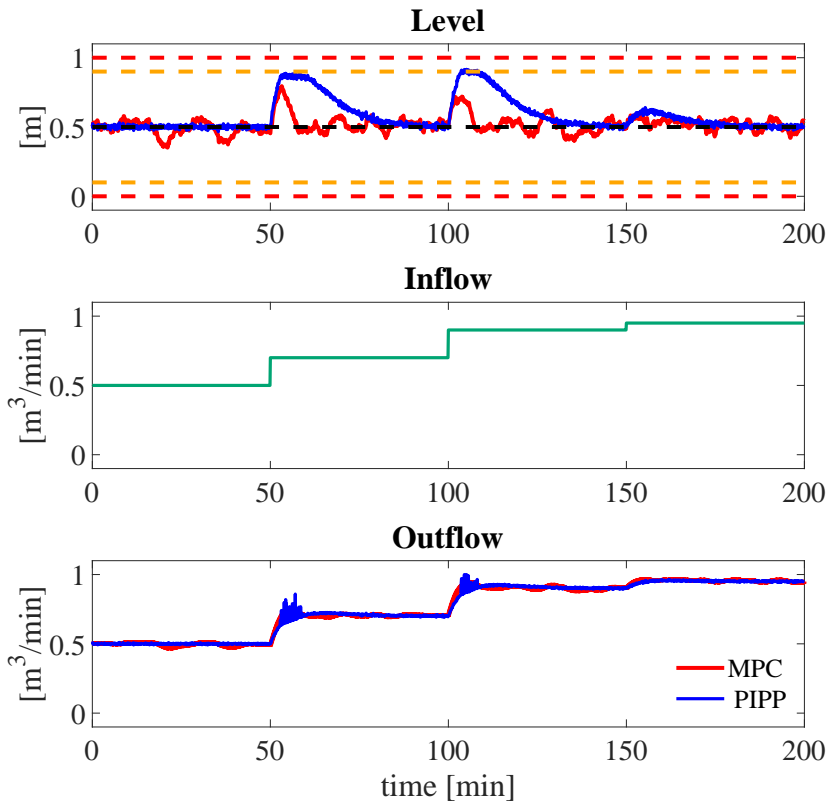


Figure 8.7: Case 2: Response of PIPP structure and MPC with step changes in inflow ( $q_{in}$ ) and measurement noise.

### 8.6.3 Case 3: sinusoidal inflow

In this case we aim for the minimization of the change in  $q_{out}$ . Fig. 8.8 shows the effect of the different gains of the proposed controller on the dampening of the sinusoidal  $q_{in}$ . It can be seen that  $q_{out}$  is heavily reduced in amplitude compared to  $q_{in}$  and that the level constraints are satisfied. For the MPC, we penalize the difference in two subsequent values for  $q_{out}$  more heavily than deviations from the level setpoint  $h_{sp} = 0.5$  m.

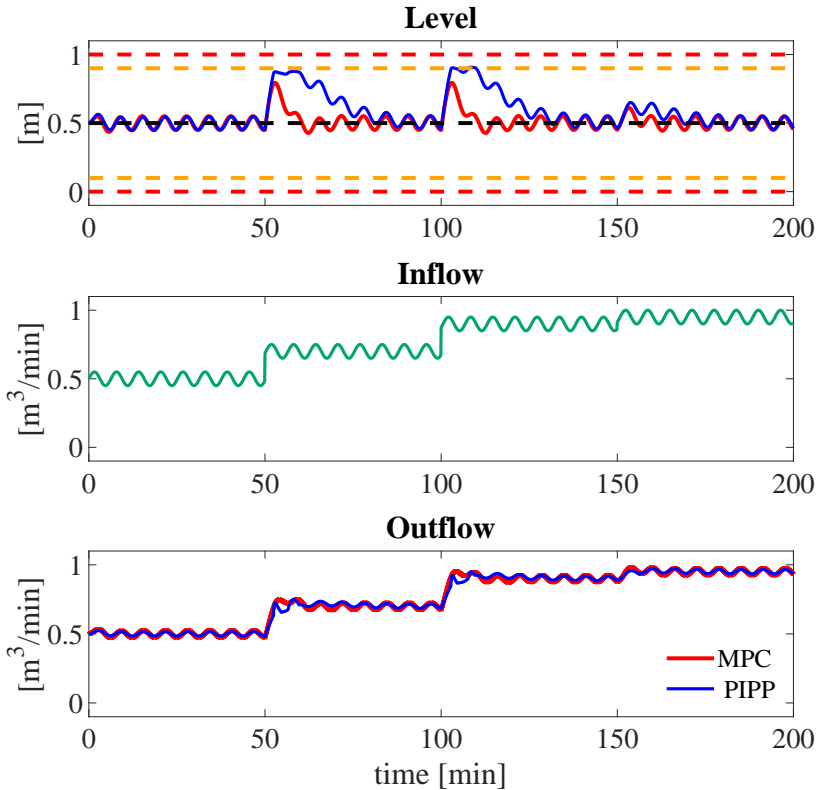


Figure 8.8: Case 3: Response of proposed PIPP control structure and MPC with sinusoidal inflow ( $q_{in}$ ).

### 8.6.4 Case 4: higher frequency sinusoidal inflow

Fig. 8.9 shows the results with a higher frequency sinusoidal disturbance ( $2 \text{ rad min}^{-1}$ ). The faster sinusoid is easier to handle and by comparing Fig. 8.9 with Fig. 8.8, we observe that  $q_{out}$  is smoother. Level constraints are also satisfied in this case. We note that the outflow variations are smaller with PIPP than with MPC in this case.

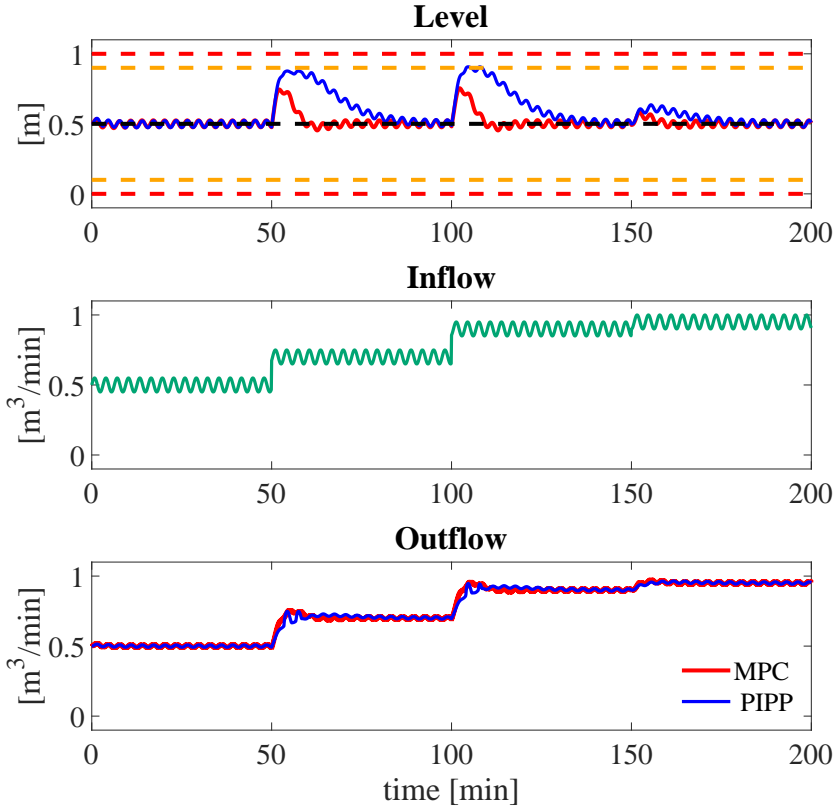


Figure 8.9: Case 4: Response of proposed PIPP control structure and MPC with higher frequency sinusoidal inflow ( $q_{in}$ ).

## 8.7 Comparison of performance of proposed PIPP structure and MPC

Table 8.1 shows the Integral Absolute Error (IAE) for deviations from the level setpoint for each of the previously presented cases, both for the proposed PIPP control structure and MPC.

Table 8.2 presents the IAE for deviations from the outflow to the steady inflow reference without added sinusoidal (compare Fig. 8.6 and 8.8). Furthermore, deviations from the steady inflow to the inflow that is used in the respective cases (without and with added sinusoidal) is shown as 'inflow deviation'. A clear reduction in deviations from the outflows compared to the respective inflows can be seen for cases 3 and 4, which are the cases with added sinusoidal disturbances.

Table 8.1: Deviation of the level from its setpoint.

Case	Proposed PIPP structure	MPC
1	16.19	3.33
2	16.77	9.04
3	19.15	8.76
4	17.47	5.79

We can also pinpoint that the proposed controller performs better with high frequency disturbances, as the deviation is lower in case 4 (high frequency) compared to case 3 (low frequency).

Table 8.2: Deviation of the outflow from the steady inflow setpoint.

Case	Proposed PIPP structure	MPC	Inflow deviation
1	1.75	1.19	0
2	2.24	2.46	0
3	3.24	4.06	6.37
4	2.56	2.42	6.35

Another performance index that could be used to quantify how the outflow is smoothed is the "total variation" or integrated absolute variation of the MV, corresponding to the sum of all "moves" of the MV:

$$\sum_{k=0}^{N-1} \left| \frac{q_{out,k} - q_{out,k-1}}{t_k - t_{k-1}} \right| \quad (8.8)$$

As we desire to smoothen  $q_{out}$ , this value should be as small as possible. Table 8.3 shows this performance index. It can be observed that the best performance of the proposed PIPP structure is when there is no measurement noise (cases 1, 3 and 4). In these cases, the PIPP performance is better than the presented MPC implementation. This can partly be explained because in the presence of noise (case 2), when the level (CV) is close to the limits, the high gain P-controllers take over and  $q_{out}$  (MV) is correspondingly moved aggressively, see Fig. 8.7.

For all simulation cases, simulation times for the proposed structure were in the range  $0.9 \pm 0.04$  s, whereas the runtime for the MPC was in the range of 88.4 – 177.6 s, depending on the case. The long runtime for the MPC was mostly due to the relatively large horizon of  $N = 50$ .

Table 8.3: Total outflow variation.

Case	Proposed PI structure	MPC
1	5.56	6.88
2	123.38	13.40
3	27.35	37.15
4	28.23	34.89

## 8.8 Concluding remarks

In this chapter, we presented a simple, yet efficient level control structure for set point tracking and safety-related lower and upper constraint satisfaction in industrial tanks. The proposed PIPP control algorithm relies in simple and easy to tune P and PI controllers. The proposed method performs much better than standard PI controllers and has a performance comparable to standard MPC in the the exact same simulation cases. These cases include the investigation of sinusoidal and step disturbances for the inflow and white noise added to the level measurement, respectively.

The proposed controller is not only able to effectively smoothen the use of the controlled variable, it is furthermore able to avoid violation of the safety constraints on upper and lower limits. Additionally, it gives the possibility to track the desired level set point in the presence of disturbances and noise. When compared to standard MPC, the proposed structure has the advantage that implementation of PI structures is simpler and computational times are consistently and substantially shorter. Additionally, tuning of PI controllers using the SIMC rule is fast and uncomplicated compared to tuning of MPC. The presented approach is particularly convenient for surge tanks with relatively small volumes, where it is difficult to get dampening of flow disturbances without violating liquid level constraints.

A similar control structure had been proposed by Shunta and Fehervari (1976). Compared to that work, our contributions are:

- Proposing a structure using a mid-selector, based on the understanding of the behavior of the gains with respect to the level.
- Suggesting the set-points for the P-controllers that avoid overflowing or emptying the tank.
- Insights regarding tuning (Section 8.4.1)
- Simulation of different disturbances (step and sinusoidal), including noise.
- Comparison with MPC.

# Chapter 9

## Conclusions and future work

This chapter will give summarize the main conclusions of this Ph.D. work, and give some insights about its applicability and possible future research directions.

### 9.1 Conclusions

In this thesis we showed that in some cases steady-state optimal operation can be easily achieved using single-loop PID-based control structures, also when the set of active constraints changes. We also developed tools to systematically design these structures. The main advantage of this approach, compared to model-based optimization is that it requires little modeling effort. Contrary to MPC, there are well-known tuning rules and methods for PID controllers (see Skogestad (2003); Åström and Hägglund (2004); Shamsuzzoha and Skogestad (2010); Hägglund (2019)). In addition, PID-based control is simple to implement and has a reduced computational load (Skogestad and Postlethwaite, 2005) (e.g. see Chapter 8).

A five-step systematic procedure to design constraint switching schemes, using PID-based classical controllers and logics (selectors), was introduced in Chapter 2. An important step to design control structures for optimal operation is to identify the constraints and organize them in a *priority list*<sup>1</sup>. Using this list, constraints are given up systematically and feasibility is assured when the set of active constraints changes<sup>2</sup>. Three types of active constraint switches and the advanced control structures to use in each case were identified. The proposed procedure was used to design the control structure of three different processes; a mixing process and a distillation column in Chapter 2 and a refrigeration cycle with heat recovery in Chapter 3.

In Chapter 4, we compared the three MV to CV switching strategies, which had been identified in Chapter 2; namely, split range control, input (valve position)

---

<sup>1</sup>The use of priority lists for this purpose was proposed in Reyes-Lúa et al. (2018d).

<sup>2</sup>This approach is used for MPC (e.g. Aske et al. (2005)), but to the knowledge of the author it had not been suggested for PID-based control structures.

and different controllers with different setpoints, all in combination with a selector. As case study, we used a cooler in which optimality is defined as keeping the CV (temperature) at its setpoint and maximizing the use of the MV (throughput). In this case, split range control is the only structure that reached optimality, as the others required back-off, either from the setpoint or from the MV limit. When comparing split range control to MPC, both follow the priority list of constraints that had been defined and reach operation at steady-state. In this case study we see that both, advanced control structures and MPC, have advantages and disadvantages. For example, PID-based control structures do not require an explicit model and tuning can be done in a systematic way, while MPC handles constraints, interactions and optimality by design. The designer of the control structure should be aware of this to choose the most appropriate approach.

Split range control is widely used in industry, but it had not been studied much in academia. A systematic method to design standard split range control, considering also the dynamic effect of the MVs on the CV, was missing. In Chapter 5 we proposed a method to design split range controllers by using the slopes ( $\alpha_i$ ) in the split range block, or equivalently the split values ( $v^*$ ), as parameters to get the desired controller for each  $MV_i$ . This method is used throughout the thesis with good results<sup>3</sup>. However, standard split range control has limitations with respect to tuning and in Chapter 6 we proposed generalized split range control structure using the *baton strategy* and demonstrated the feasibility of implementing this structure in a case study<sup>4</sup>.

In cases such as the cooler studied in Chapter 4, having different controllers with different setpoints when there is more than one available MV for one CV implies a back-off from optimality. However, in Chapter 7 we showed that in other cases it can be optimal to have different setpoints, and we proposed a procedure to find optimal setpoint deviations. In a case study, we demonstrated the economic convenience of using different setpoints when there is a trade-off between a linear cost for input usage and a quadratic penalty for setpoint deviation.

Finally, in Chapter 8 we presented a simple, yet efficient level PID-based control structure that uses a *mid-selector* that can be used for set point tracking and safety-related lower and upper constraint satisfaction in industrial tanks.

---

<sup>3</sup>All the split range controllers in this thesis, except the one in Chapter 4, were designed using the method proposed in Chapter 5.

<sup>4</sup>The proposed generalized split range control structure is implemented in a second case study in Appendix F.



## 9.2 Future work

The findings in this Ph.D. and the general framework introduced for active constraint switches using advanced control structures can be extended in the following directions:

- Study the effects of other decisions (steps) of the plantwide control procedure (Skogestad, 2000) on the use of advanced control structures to design the supervisory layer and vice-versa. An example of this work can be further analysis of the effect of changes in the location of the throughput manipulator (TPM) in the design of the supervisory layer using advanced control structures.
- Application of the generalized procedure in different processes, and demonstration of its effectiveness and limitations.
- Specifying the limitations for active constraint switching using PID-based control structures<sup>5</sup>.
- Developing a method to quantify the complexity of using advanced control structures in the supervisory layer in comparison to using MPC for different processes. This can help the designer of the control system to choose between the two options.
- Incorporating dynamic considerations into the design of the supervisory layer using advanced control structures. For example, as discussed in Section 2.5.2, it can be that with the pairing with a better dynamic behavior we have an MV that is likely to saturate controlling an important CV. Then, it is possible to use some of the control structures for MV to CV switching. However, in Chapter 2 we do not discuss how to handle MV saturation when there is no available MV with a convenient dynamic behavior (e.g. relative gain) for the important CV<sup>6</sup>.

Different control structures can be studied or developed using the findings in this work. For example, in Chapter 7 we identified that implementing different controllers with different setpoints can be optimal in some cases. We assumed that  $\Delta y^{sp}$  would be large enough such that only one input would be active at a time. There is still no defined criterion to find the theoretical minimum setpoint difference ( $\Delta y^{sp,min}$ ) and defining such criterion would be interesting. In practice, to assure that only one input is active at a time, a *baton strategy block* such as the one introduced in Chapter 6 (Fig. 6.5), could be used when implementing different controllers with different setpoints (Fig. 7.4).

---

<sup>5</sup>Some work in this direction, specifically for CV-CV switching, has been done by Krishnamoorthy and Skogestad (2019).

<sup>6</sup>This type of analysis might fall into design issues, but these limitations should be discussed.

When implementing in practice the control structures analyzed in this thesis, it may be necessary to back off from the optimal value of CV constraints; for example, when these are difficult to measure or there are poor dynamics (Narraway et al., 1991). For example, Govatsmark and Skogestad (2005) analyzed back-off policies (flexible and robust) and their effect on feasibility. Incorporating previous understanding in this regard with the work of this thesis will support the appropriate implementation of control structures such that the expected performance is obtained.

# Bibliography

- Allison, B. J. and Isaksson, A. J. (1998). Design and performance of mid-ranging controllers. *Journal of Process Control*, 8(5-6):469–474.
- Allison, B. J. and 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.
- Alsop, N. (2016). Implementing Mid Ranging in a DCS Environment. *IFAC-PapersOnLine*, 49(7):550–555.
- Alstad, V., Skogestad, S., and Hori, E. S. (2009). Optimal measurement combinations as controlled variables. *Journal of Process Control*, 19(1):138–148.
- Amezquita-Brooks, L., Liceaga-Castro, J. U., Liceaga-Castro, E., Martínez-Vázquez, D., and García-Salazar, O. (2017). Fundamental Analysis of the Perturbation Rejection Characteristics of Single-Input-Multiple-Output Systems Subject to Multiple Perturbations. *Mathematical Problems in Engineering*.
- Andersson, J. (2013). *A General Purpose Software Framework for Dynamic Optimization*. Ph.d. thesis, Arenberg Doctoral School, KU Leuven.
- Aske, E. M. B. and Skogestad, S. (2009). Dynamic Degrees of Freedom for Tighter Bottleneck Control. In de Brito Alves, R. M., do Nascimento, C. A. O., and Biscoia, E. C., editors, *Computer Aided Chemical Engineering*, volume 27, pages 1275–1280. Elsevier.
- Aske, E. M. B., Skogestad, S., and Strand, S. (2007). Throughput maximization by improved bottleneck control. *IFAC Proceedings Volumes*, 40(5):63–68.
- Aske, E. M. B., Strand, S., and Skogestad, S. (2005). Implementation of MPC on a deethanizer at Kårstø gas plant. *IFAC Proceedings Volumes*, 38(1):437–442.
- Åström, K. and Hägglund, T. (2004). Revisiting the Ziegler–Nichols step response method for PID control. *Journal of Process Control*, 14(6):635–650.
- Åström, K. J. (1999). Automatic Control - The Hidden Technology. In Frank, P. M., editor, *Advances in Control - Highlights of ECC99*, chapter 1, pages 1–28. Springer, Duisburg, 1st edition.

- Åström, K. J. and Bell, R. D. (2000). Drum-boiler dynamics. *Automatica*, 36:2000.
- Åström, K. J. and Hägglund, T. (1995). *PID Controllers - Theory, Design, and Tuning*. ISA, 2nd edition.
- Åström, K. J. and Hägglund, T. (2001). The future of PID control. *Control Engineering Practice*, 9:2001.
- Åström, K. J. and Hägglund, T. (2006). *Advanced PID Control*. ISA.
- Backi, C. J. and Skogestad, S. (2017). A simple dynamic gravity separator model for separation efficiency evaluation incorporating level and pressure control. In *Proceedings of the 2017 American Control Conference*, Seattle, USA.
- Bahadori, A. (2017). Control Valves. In *Oil and Gas Pipelines and Piping Systems*, pages 483–571. Elsevier.
- Balchen, J. G. and Mummé, K. I. (1988). *Process Control Structures and Applications*. Van Nostrand Reinhold Company Inc.
- Bastani, F. B. and Chen, I.-R. (1988). The role of artificial intelligence in fault-tolerant process-control systems. In *Proceedings of the first international conference on Industrial and engineering applications of artificial intelligence and expert systems - IEA/AIE '88*, volume 2, pages 1049–1058, New York, New York, USA. ACM Press.
- Bemporad, A. and Morari, M. (1999). Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35(3):407–427.
- Bequette, B. W. (2002). *Process Control: Modeling, Design, and Simulation*. Prentice-Hall.
- Blevins, T. L. (2011). *Control loop foundation: batch and continuous processes*. International Society of Automation (ISA).
- Bristol, E. (1980). After DDC idiomatic control. *Chemical Engineering Progress*, 76(11):84–89.
- Buckley, P. S. (1964). *Techniques of Process Control*. John Wiley & Sons, Inc., Delaware, USA, 1st edition.
- Campo, P. J. and Morari, M. (1989). Model Predictive Optimal Averaging Level Control. *AIChE Journal*, 35(4):579–591.
- Cheung, T.-F. and Luyben, W. L. (1980). Nonlinear and Nonconventional Liquid Level Controllers. *Industrial & Engineering Chemistry Fundamentals*, 19(1):93–98.
- Eckman, D. (1945). *Principles of industrial control*, pp.204–207. John Wiley & Sons, New York.
- Faanes, A. and Skogestad, S. (2003). Buffer Tank Design for Acceptable Control Performance. *Industrial & Engineering Chemistry Research*, 42(10):2198–2208.

- Fatani, S. T., Patwardhan, R. S., and Andreu, M. A. L. (2017). A novel single-input two-output (SITO) strategy for split range control. In *2017 6th International Symposium on Advanced Control of Industrial Processes (AdCONIP)*, pages 348–353. IEEE.
- Ferreau, H. J., Kirches, C., Potschka, A., Bock, H. G., and Diehl, M. (2014). qpOASES: a parametric active-set algorithm for quadratic programming. *Mathematical Programming Computation*, 6(4):327–363.
- Fink, E. D. (1945). *Instruments and Process Control*, pp. 120,149. Delmar Publishers, Inc., Albany, NY.
- Fonseca, R. R., Schmitz, J. E., Fileti, A. M. F., and da Silva, F. V. (2013). A fuzzy-split range control system applied to a fermentation process. *Bioresource Technology*, 142:475–482.
- Forbes, M. G., Patwardhan, R. S., Hamadah, H., and Gopaluni, R. B. (2015). Model Predictive Control in Industry: Challenges and Opportunities. *IFAC-PapersOnLine*, 48(8):531–538.
- Foss, A. S. (1973). Critique of chemical process control theory. *AIChE Journal*, 19(2):209–214.
- Freudenberg, J. and Middleton, R. (1999). Properties of single input, two output feedback systems. *International Journal of Control*, 72(16):1446–1465.
- Ganesh, H. S., Fritz, H. E., Edgar, T. F., Novoselac, A., and Baldea, M. (2019). A model-based dynamic optimization strategy for control of indoor air pollutants. *Energy and Buildings*, 195:168–179.
- Garelli, F., Mantz, Ricardo, J., and De Battista, H. (2011). *Advanced Control for Constrained Processes and Systems*. The Institution of Engineering and Technology. IET Control Engineering Series 75.
- Glemmestad, B. (1997). *Optimal Operation of Integrated Processes - Studies on Heat Recovery Systems*, Ch. 5. PhD thesis, Norwegian University of Science and Technology (NTNU), Norway.
- Govatsmark, M. S. and Skogestad, S. (2005). Selection of Controlled Variables and Robust Setpoints. *Industrial & Engineering Chemistry Research*, 44(7):2207–2217.
- Govind, R. and Powers, G. J. (1978). Synthesis of Process Control Systems. *IEEE Transactions on Systems, Man, and Cybernetics*, 8(11):792–795.
- Govind, R. and Powers, G. J. (1982). Control system synthesis strategies. *AIChE Journal*, 28(1):60–73.
- Grimholt, C. (2018). *Optimal tuning of PID controllers*. PhD thesis, Norwegian University of Science and Technology (NTNU).

- Grimholt, C. and Skogestad, S. (2016). Optimal PID control of double integrating processes. *IFAC-PapersOnLine*, 49(7):127–132.
- Grimholt, C. and Skogestad, S. (2018). 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.
- Gullo, P., Tsamos, K. M., Hafner, A., Banasiak, K., Ge, Y. T., and Tassou, S. A. (2018). Crossing  $CO_2$  equator with the aid of multi-ejector concept: A comprehensive energy and environmental comparative study. *Energy*, 164:236–263.
- Gupta, P., Rana, K., Kumar, V., and Mishra, P. (2015). Split-range control of a Jacketed CSTR using self-tuning fuzzy PI controller. In *2015 International Conference on Advances in Computer Engineering and Applications*, pages 527–533. IEEE.
- Hägglund, T. (1997). *Praktisk processreglering (in Swedish)*. Studentlitteratur, Lund, Sweden, 2nd edition.
- Hägglund, T. (2017). The Tracking Ratio Station. *Control Engineering Practice*, 69:122–130.
- Hägglund, T. (2019). The one-third rule for pi controller tuning. *Computers and Chemical Engineering*, 127:25–30.
- Hägglund, T. and Guzmán, J. L. (2018). Development of Basic Process Control Structures. *IFAC-PapersOnLine*, 51(4):775–780.
- Haugwitz, S., Karlson, M., Velut, S., and Hagander, P. (2005). Anti-windup in mid-ranging control. In *44th IEEE Conference on Decision and Control*, Seville, Spain.
- Jacobsen, M. G. and Skogestad, S. (2011). Active Constraint Regions for Optimal Operation of Chemical Processes. *Industrial & Engineering Chemistry Research*, 50(19):11226–11236.
- Jacobsen, M. G. and Skogestad, S. (2012). Active Constraint Regions for Optimal Operation of Distillation Columns. *Industrial & Engineering Chemistry Research*, 51(7):2963–2973.
- Jensen, J. B. (2008). *Optimal operation of refrigeration cycles*. Phd thesis, NTNU, Norway.
- Jordal, K., Reyes-Lúa, A., and Langørgen, Ø. (2015). The potential benefit of using CLC in industrial boilers. In *The 8th Trondheim CCS Conference*, Trondheim, Norway.
- Kim, M.-H., Pettersen, J., and Bullard, C. W. (2004). Fundamental process and system design issues in  $CO_2$  vapor compression systems. *Progress in Energy and Combustion Science*, 30(2):119–174.

- Krishnamoorthy, D. (2019). *Novel Approaches to Online Process Optimization under Uncertainty*. Phd thesis, Norwegian University of Science and Technology.
- Krishnamoorthy, D. and Skogestad, S. (2019). Online Process Optimization with Active Constraint Set Changes using Simple Control Structures. *Industrial & Engineering Chemistry Research*, 58(30):13555–13567.
- Liao, S., Zhao, T., and Jakobsen, A. (2000). A correlation of optimal heat rejection pressures in transcritical carbon dioxide cycles. *Applied Thermal Engineering*, 20(9):831–841.
- Lieberman, N. P. (2008.). *Troubleshooting Process Plant Control*. Wiley.
- Lindholm, A., Johnsson, C., Häggglund, T., and Carlsson, H. (2012). Reducing revenue loss due to disturbances in utilities using buffer tanks - a case study at perstorp. In *Proceedings of the 17th Nordic Process Control Workshop*, pages 199–200. Technical University of Denmark DTU Informatics.
- Lipták, B. G. (1985). Control and on-off valves. In *Instrument Engineers' Handbook*, chapter 4, pages 410–412. Chilton Book Company, 2nd edition.
- Lu, J. Z. (2003). Challenging control problems and emerging technologies in enterprise optimization. *Control Engineering Practice*, 11(8):847–858.
- Lucia, S. and Engell, S. (2012). Multi-stage and Two-stage Robust Nonlinear Model Predictive Control. *IFAC Proceedings Volumes*, 45(17):181–186.
- Lucia, S., Finkler, T., and Engell, S. (2013). Multi-stage nonlinear model predictive control applied to a semi-batch polymerization reactor under uncertainty. *Journal of Process Control*, 23(9):1306–1319.
- Marlin, T. E. (2000). *Process Control. Designing Processes and Control Systems for Dynamic Performance*. McGraw Hill.
- Matlab (2019). Anti-Windup Control Using a PID Controller.
- Mayne, D. Q. (2014). Model predictive control: Recent developments and future promise. *Automatica*, 50(12):2967–2986.
- McDonald, K. A., McAvoy, T. J., and Tits, A. (1986). Optimal Averaging Level Control. *AIChE Journal*, 32(1):75–86.
- Mercer, S. (1968). Cryogenics: a technological tool for space scientists. *Cryogenics*, 8(2):68–78.
- Minasidis, V., Skogestad, S., and Kaistha, N. (2015). Simple Rules for Economic Plantwide Control. In Gernaey, K. V., Huusom, J. K., and Gani, R., editors, *PSE 2015 and ESCAPE-25*, pages 101–108. Elsevier Science Direct.
- Narraway, L., Perkins, J., and Barton, G. (1991). Interaction between process design and process control: economic analysis of process dynamics. *Journal of Process Control*, 1(5):243–250.

- Nekså, P. (2002).  $CO_2$  heat pump systems. *International Journal of Refrigeration*, 25(4):421–427.
- Osborn, P. D. (1985). Data charts and tables. In *Handbook of Energy Data and Calculations*, pages 1–67. Elsevier.
- Petrov, M., Ganchev, I., and Taneva, A. (2002). Fuzzy PID Control of Nonlinear Plants. In *Proceedings of the 1st International IEEE Symposium on Intelligent Systems*, pages 30–35, Varna, Bulgaria. IEEE.
- Piovoso, M., Dahl, K., and Kosanovich, K. (1995). Control of a Batch Reactor Using a Multivariate Statistical Controller Design. *IFAC Proceedings Volumes*, 28(9):357–361.
- Price, R. M. and Georgakis, C. (1993). Plantwide regulatory control design procedure using a tiered framework. *Industrial & Engineering Chemistry Research*, 32(11):2693–2705.
- Price, R. M., Lyman, P. R., and Georgakis, C. (1994). Throughput manipulation in plantwide control structures. *Industrial & Engineering Chemistry Research*, 33(5):1197–1207.
- Qin, S. and Badgwell, T. A. (2003). A survey of industrial model predictive control technology. *Control Engineering Practice*, 11(7):733–764.
- Reyes-Lúa, A., Andreasen, G., Larsen, L. F. S., Stoustrup, J., and Skogestad, S. (2019a). Optimal operation of a  $CO_2$ -refrigeration system with heat recovery. In *Proceedings of the 29th European Symposium on Computer Aided Process Engineering (ESCAPE)*, Eindhoven, Netherlands. Computer-aided chemical engineering.
- Reyes-Lúa, A., Backi, C. J., and Skogestad, S. (2018a). Improved PI control for a surge tank satisfying level constraints. In *3rd IFAC Conference on Advances in Proportional-Integral-Derivative Control (PID18)*, volume 51, pages 835–840, Ghent, Belgium. IFAC-Papers OnLine.
- Reyes-Lúa, A. and Skogestad, S. (2016). Optimal operation of vapor-compression cycles in off-design conditions. In Kitanovski, A. and Poredoš, A., editors, *29th International Conference on Efficiency, Cost, Optimisation, Simulation and Environmental Impact of Energy Systems (ECOS)*, Portorož, Slovenia.
- Reyes-Lúa, A. and Skogestad, S. (2019a). Multi-input single-output control for extending the operating range: Generalized split range control using the baton strategy. *Journal of Process Control (Under review)*.
- Reyes-Lúa, A. and Skogestad, S. (2019b). Multiple-Input Single-Output Control for Extending the Steady-State Operating Range—Use of Controllers with Different Setpoints. *Processes*, 7(12):941.



- Reyes-Lúa, A. and Skogestad, S. (2019c). Systematic design of active constraint switching using classical advanced control structures. *Industrial & Engineering Chemistry Research*.
- Reyes-Lúa, A. and Skogestad, S. (2020). Active constraint switching with the generalized split range control structure using the baton strategy. In *21st IFAC World Congress (Submitted)*, Berlin, Germany. IFAC Papers Online.
- Reyes-Lúa, A., Solvik, M., and Skogestad, S. (2016). Inclusion of thermodynamic equations for efficient steady-state process optimization. In *Proceedings of the 26th European Symposium on Computer Aided Process Engineering (ESCAPE)*, Portorož, Slovenia. Computer-aided chemical engineering.
- Reyes-Lúa, A., Zotică, C., Das, T., Krishnamoorthy, D., and Skogestad, S. (2018b). Changing between Active Constraint Regions for Optimal Operation: Classical Advanced Control versus Model Predictive Control. In *Proceedings of the 28th European Symposium on Computer Aided Process Engineering (ESCAPE)*, Graz, Austria. Computer-aided chemical engineering.
- Reyes-Lúa, A., Zotică, C., Forsman, K., and Skogestad, S. (2019b). Systematic Design of Split Range Controllers. In *12th IFAC Symposium on Dynamics and Control of Process Systems, including Biosystems (DYCOPS)*, Florianópolis, Brazil. IFAC-Papers OnLine.
- Reyes-Lúa, A., Zotică, C., and Skogestad, S. (2018c). Optimal operation using classical advanced control structures. In *21st Nordic Process Control Workshop (NPCW)*, Turku (Åbo), Finland.
- Reyes-Lúa, A., Zotică, C., and Skogestad, S. (2018d). Optimal Operation with Changing Active Constraint Regions using Classical Advanced Control. In *10th IFAC Symposium on Advanced Control of Chemical Processes (ADCHEM)*, Shenyang, China. IFAC-Papers OnLine.
- Rosander, P., Isaksson, A. J., Löfberg, J., and Forsman, K. (2012). Practical Control of Surge Tanks Suffering from Frequent Inlet Flow Upsets. In Vilanova, R. and Visioli, A., editors, *IFAC Conference on Advances in PID Control*, Brescia, Italy. Elsevier Ltd.
- Sawalha, S. (2013). Investigation of heat recovery in  $CO_2$  trans-critical solution for supermarket refrigeration. *International Journal of Refrigeration*, 36(1):145–156.
- Schmid, C. and Biegler, L. (1994). Quadratic programming methods for reduced hessian SQP. *Computers & Chemical Engineering*, 18(9):817–832.
- Schuermans, J. (2019). Review of override control methods. *Computer Aided Chemical Engineering*, 46:649–654.
- Seborg, D. E. (1999). A perspective on advanced strategies for process control. In Frank, P. M., editor, *Advances in Control - Highlights of ECC991*, chapter 4, pages 103–134. Springer, Duisburg.

- Seborg, D. E., Edgar, T. F., and Mellichamp, D. A. (2003). *Process Dynamics and Control*. John Wiley & Sons, Inc., 2nd edition.
- Shamsuzzoha, M. and Skogestad, S. (2010). The setpoint overshoot method: A simple and fast closed-loop approach for PID tuning. *Journal of Process Control*, 20(10):1220–1234.
- Shen-Hui, D., Gang, Z., and Mei-Rong, H. (2011). Research on regulator signal segment match to control valve in split range control system. In *2011 International Conference on Consumer Electronics, Communications and Networks (CECNet)*, pages 4350–4353. IEEE.
- Shinsky, F. G. (1978). *Energy conservation through control*. Academic Press, New York.
- Shinsky, F. G. (1981). *Controlling multivariable processes*. Instrument Society of America, Massachusetts, U.S.A.
- Shinsky, F. G. and Shunta, J. (1995). Selective, Override and Limit Controls. In Lipták, B. G., editor, *Instrument Engineers' Handbook*, chapter 1.17, pages 112–118. Butterworth Heinemann, 3rd edition.
- Shinsky, F. (1977). The stability of interacting control loops with and without decoupling. *IFAC Proceedings Volumes*, 10(6):21–30.
- Shinsky, F. (1988). *Process Control Systems*. McGraw-Hill, 3rd edition.
- Shunta, J. and Fehervari, W. (1976). Nonlinear control of liquid level. *Instrumentation Technology*, 23:43–48.
- Skjervold, V., Gullberg, R., Langørgen, Ø., Berstad, D., Schandera, C., Reyes-Lúa, A., and Jordal, K. (2016). Evaluation of a natural gas-fired CLC boiler for industrial steam generation. In *13th International Conference on Greenhouse Gas Control Technologies*, Lausanne, Switzerland.
- Skogestad, S. (2000). Plantwide control: the search for the self-optimizing control structure. *Journal of Process Control*, 10(5):487–507.
- Skogestad, S. (2003). Simple analytic rules for model reduction and PID controller tuning. *Journal of Process Control*, 13:291–309.
- Skogestad, S. (2004). Control structure design for complete chemical plants. *Computers & Chemical Engineering*, 28(1-2):219–234.
- Skogestad, S. (2007). The Dos and Don'ts of Distillation Column Control. *Chemical Engineering Research and Design*, 85(1):13–23.
- Skogestad, S. (2012). Economic Plantwide Control. In Kariwala, Vinay Rangaiah, G. P., editor, *Plantwide Control*, chapter 11, pages 229–251. Wiley.

- Skogestad, S. (2015). Control Structure Selection. In *Encyclopedia of Systems and Control*, pages 202–215. Springer London, London.
- Skogestad, S. and Grimholt, C. (2012). The SIMC Method for Smooth PID Controller Tuning. In *PID Control in the Third Millennium*, pages 3–44. Springer.
- Skogestad, S., Lundström, P., and Jacobsen, E. W. (1990). Selecting the best distillation control configuration. *AIChE Journal*, 36(5):753–764.
- Skogestad, S. and Morari, M. (1988). Understanding the dynamic behavior of distillation columns. *Industrial & Engineering Chemistry Research*, 27(10):1848–1862.
- Skogestad, S. and Postlethwaite, I. (2005). *Multivariable Feedback Control*. Wiley, 2. ed. edition.
- Smith, C. L. (2010). *Advanced Process Control. Beyond Single Loop Control*. John Wiley & Sons, New Jersey.
- Smith, C. L. (2014). *Control of Batch Processes*. John Wiley & Sons, Incorporated.
- Stephanopoulos, G. (1983). Synthesis of control systems for chemical plants A challenge for creativity. *Computers & Chemical Engineering*, 7(4):331–365.
- Stephanopoulos, G. (1984). *Chemical Process Control: An Introduction to Theory and Practice*. Prentice-Hall.
- Strand, S. and Saggi, J. R. (2004). MPC in Statoil – Advantages with in-house technology. *IFAC Proceedings Volumes*, 37(1):97–103.
- Sun, B., Skogestad, S., Lu, J., and Zhang, W. (2018). Dual SIMC-PI Controller Design for Cascade Implement of Input Resetting Control with Application. *Industrial & Engineering Chemistry Research*, 57(20):6947–6955.
- Sun, B. X., Shah, A., and Amalraj, J. (2015). A dual split-range control strategy for pressure and flow processes. *Control Engineering*.
- Tani, T., Murakoshi, S., and Umamo, M. (1996). Neuro-Fuzzy Hybrid Control System of Tank Level in Petroleum Plant. *IEEE Transactions on Fuzzy Systems*, 4(3):360–368.
- Vandenbussche, P. (1975). Digital Transposition and Extension of Classical Analogical Control Algorithms. *IFAC Proceedings Volumes*, 8(1):41–44.
- Wächter, A. and Biegler, L. T. (2005). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57.
- Wade, H. L. (2004). *Basic and Advanced Regulatory Control: System Design and Application*. International Society of Automation.

Wang, S. (2010). *Intelligent Buildings and Building Automation*. Spon Press, first edition.

Young, A. (1955). *An introduction to process control system design*. Longmans, Green and Co. Ltd., London.

Zotică, C., Nord, L. O., Kovács, J., and Skogestad, S. (2019). Optimal Operation and Control of Heat-to-Power Cycles: a New Perspective using a Systematic Plantwide Control Approach. pages 1429–1434.

# Supporting Information



## Appendix

# Model for mixing of air and MeOH in Chapter 2 and Appendix G

Eq. (A.1) and (A.2) describe the steady-state mass and molar balance for this system.

$$\dot{m}_{tot} = \dot{m}_{air} + \dot{m}_{MeOH} \quad (\text{A.1})$$

which corresponds to  $y_1 = u_1 + u_2$ .

$$x_{MeOH} = \frac{\dot{m}_{air}/MW_{air}}{\dot{m}_{air}/MW_{air} + \dot{m}_{MeOH}/MW_{MeOH}} \quad (\text{A.2})$$

where  $\dot{m}_i$  are the air and MeOH inlet mass flow rates (kg/h),  $\dot{m}_{tot}$  is the outlet total mass flow rate (kg/h), and  $x_{MeOH}$  is the methanol molar concentration (kmol/kmol), and  $MW_i$  are the methanol and air (average) molecular weights (kg/kmol).

Taking into account the dynamics of the actuators and the measurements, the dynamic responses can be approximated to the first order transfer functions with time delay,

$$G_i = \frac{K_{p,i}e^{-\theta_i s}}{\tau_i s + 1},$$

in Table A.1, which were identified using step-tests for each possible pairing.

Table A.1: First order transfer functions for mixing of MeOH and air.

$MV_i$	$CV_i$	$K_{p,i}$	$\tau_i$ (s)	$\theta_i$ (s)
$\dot{m}_{air}$	$x_{MeOH}$	-3.43E-05	2.83	0.37
$\dot{m}_{air}$	$\dot{m}_{tot}$	1.1229	2.90	0.56
$\dot{m}_{MeOH}$	$x_{MeOH}$	2.94E-04	1.26	0.20
$\dot{m}_{MeOH}$	$\dot{m}_{tot}$	9.14	3.80	1.15





## Optimization results for distillation column in Chapter 2

These results were found by solving optimization problem in Eq. 7 and the data in Table 2. The optimization was done using NLP solver `fmincon`, active-set, in Matlab and the model for Column A described in Skogestad and Postlethwaite (2005). The active constraint regions in Fig. 15 were found by evaluating the Lagrangian multipliers of the constraints (Jacobsen and Skogestad, 2011). Fig. B.1 shows the contour plot for the cost function in optimization problem (2.7). As expected, the profit is higher with lower price for boil-up ( $p_V$ ) and higher production rate ( $F$ ).

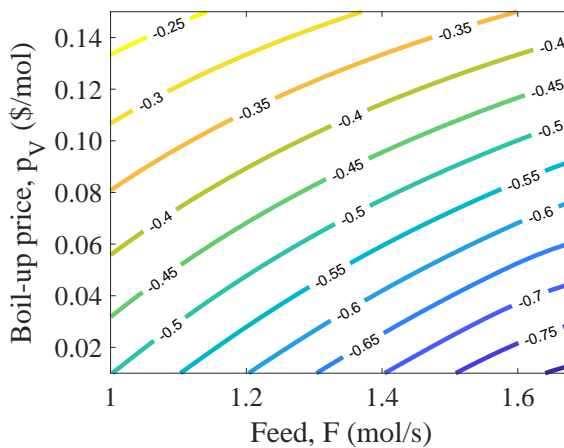


Figure B.1: Contour plot for objective function  $J^{opt}$  for distillation column.

Fig. B.2 shows the contour plot for  $x_D^{opt}$ . We plot this to get some insight on the optimal set point for  $x_D$ , which is the only part of this procedure in which we need an actual model. It is very clear that the behavior of  $x_D^{opt}$  depends on the active constraint region. In region I, with lower  $p_V$ , it is convenient to overpurify. As  $p_V$  increases,  $x_D^{opt}$  is reduced until we reach  $x_D^{min} = 0.95$  and enter region III. Starting

from region I, and  $F$  increases, we reach  $V = V^{max}$  and enter region III. Here,  $x_D^{opt}$  is not longer inversely proportional to  $p_V$ , but to  $F$ . At a higher  $F$ , we reach again  $x_D^{min}$  at the bottleneck ( $F = 1.68$  mol/s).

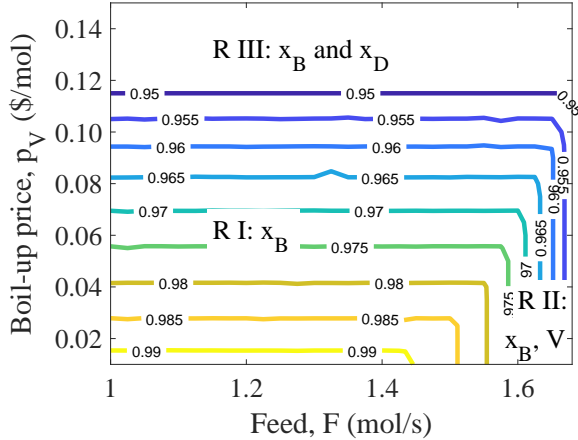


Figure B.2: Contour plot for  $x_D^{opt}$  for distillation column, with active constraint regions indicated.

In the proposed control structure, while operating in region II, we give up controlling  $x_D$  and while operating on region III,  $x_D = x_D^{min}$ . Therefore, we only need to find an appropriate set point for  $x_D$  for region I. Fig. B.3 shows that  $x_D^{opt}$  can be very well fitted as a linear function of  $p_V$  within region I. This is used as  $x_D^{sp}$ .

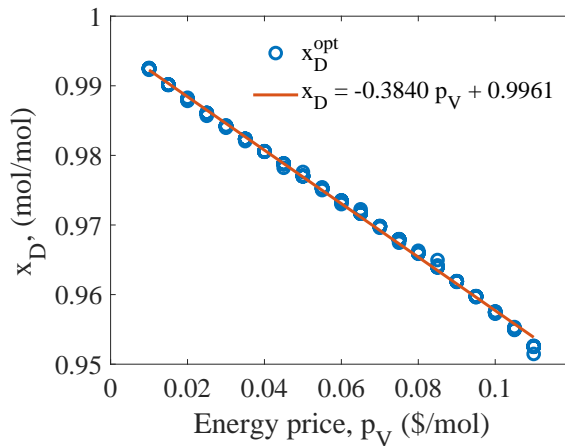


Figure B.3: Linear approximation of  $x_D^{opt}$  as a function of  $p_V$  in Region I.

## Appendix

# Supporting information for Chapter 4 - Saturation of manipulated variables

### C.1 Cooler model

We consider a countercurrent cooler, represented by the dynamic lumped model in Eq. (C.1). The cooler is discretized in space into a series of  $n = 10$  cells, as depicted in Fig. C.1. Incompressible fluids and constant heat capacities are assumed. The boundary conditions are:  $T_{H_0} = T_{H_{in}}$  for cell  $i = 1$  (inlet), and  $T_{C_{11}} = T_{C_{in}}$  for cells  $i = 10$  (outlet). The energy balance for cell  $i = 1 \dots n$  is:

$$\frac{dT_{C_i}}{dt} = \frac{F_C}{\rho_C V_{C_i}} (T_{C_{i+1}} - T_{C_i}) + \frac{U A_i (T_{H_i} - T_{C_i})}{\rho_C V_{C_i} c_{pC}} \quad (\text{C.1a})$$

$$\frac{dT_{H_i}}{dt} = \frac{F_H}{\rho_H V_{H_i}} (T_{H_{i-1}} - T_{H_i}) + \frac{U A_i (T_{H_i} - T_{C_i})}{\rho_H V_{H_i} c_{pH}} \quad (\text{C.1b})$$

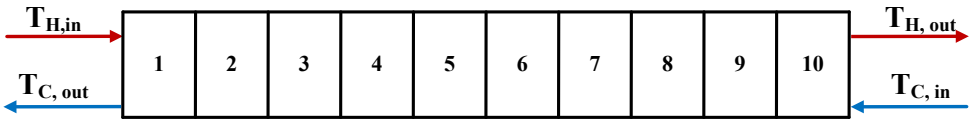


Figure C.1: Lumped model for cooler.

### C.2 Controller tuning

The tuning parameters  $K_C$  and  $\tau_I$  for the PI-controllers in Table C.1 were determined by fitting a first order with time delay (FOTD) model ( $K$ ,  $\tau$ ,  $\theta$ ) obtained from open-loop step responses of the process described in C.1, and applying the SIMC tuning rules (Skogestad, 2003).

The open-loop responses in  $T_H$  for a step in the MVs ( $F_C$  and  $F_H$ ) are depicted in Fig. C.2. The tuning for temperature controller  $TC$ , which manipulates  $F_C$ ,

is the same for the three evaluated structures. The closed-loop time constant is  $\tau_c = 2\theta = 88$  s. For *TC2*, which manipulates  $F_H$ ,  $\tau_c = 10\theta = 70$  s.

In this case study we do not use the improved design procedure for SRC introduced in Chapter 5. Here, the split range block is designed such that the split value is at  $u = 0$ , for simple implementation. To account for the different gains that  $F_C$  (negative gain) and  $F_H$  (positive gain) have on  $T_H$ , the output of the controller is respectively multiplied by 1 and  $-2$ . The integral term is the same for both MVs.

Table C.1: Tuning parameters for the cooler case study.

Parameter	TC	VPC	TC2
$K_C$	-0.055	3	0.080
$\tau_I$ (s)	74	77	86
$\tau_c$ (s)	88	12	70

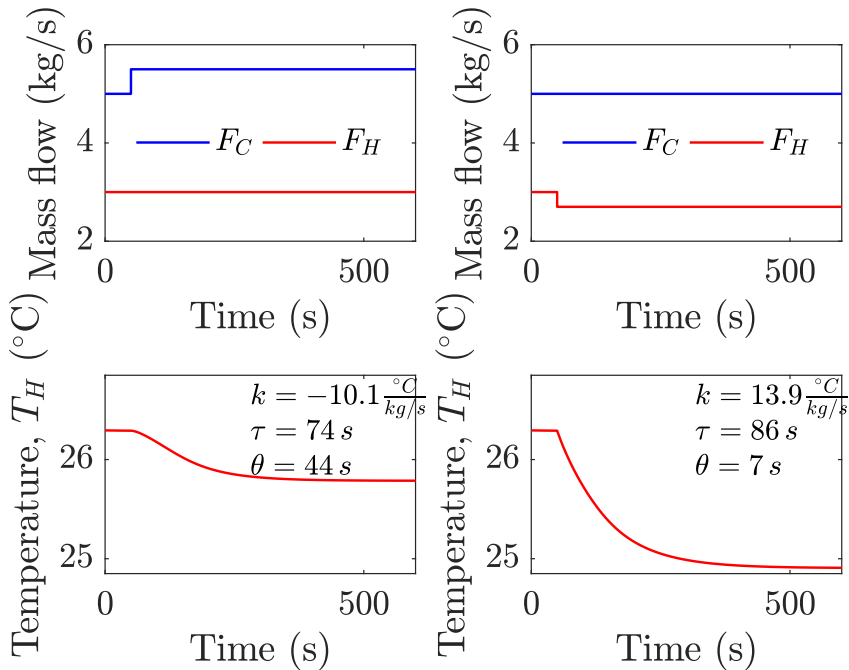


Figure C.2: Open-loop response in  $T_H$  for a step in  $F_C$  and  $F_H$ .

Fig. C.3 shows the response of  $F_C$  to a step in  $F_H$ , keeping  $T_H = T_H^{sp}$ . This closed-loop response is required for tuning the VPC, which was tuned for tight control, i.e.  $\tau_c = \theta = 12$  s.

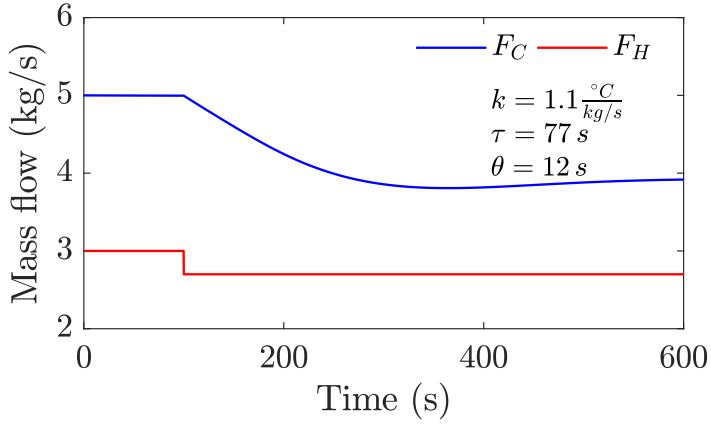


Figure C.3: Response in  $F_C$  for a step in  $F_H$ ; used to tune the valve position controller.



# Supporting information for Chapter 6 - Generalized split range control

## D.1 Parameters for standard split range controller for case study

Table D.1 summarizes the information that describes the standard split range block in Fig. 6.8, where  $u_i^0$  corresponds to the bias, the slopes are  $\alpha_i$  and  $\Delta v_i$  is the range of the internal variable for each input.

Table D.1: Values for  $\alpha_i$ ,  $\Delta v_i$  and  $u_{i,0}$ .

	AC	CW	HW	EH
$\alpha_i$	-6.7600	-4.2250	2.3472	5.2813
$\Delta v_i$	0.1479	0.2367	0.4260	0.1893
$u_i^0$	1.0000	1.6250	-0.9028	-4.2813

## D.2 MPC implementation

To implement MPC for the system described in the case study, the dynamic optimization problem is set up using Matlab (?). The transfer function model, Eq. (5.10), relating the inputs with the output, is converted to the discrete-time linear time-invariant (LTI) system described by:

$$x_{k+1} = Ax_k + Bu_k \tag{D.1a}$$

$$y_k = Cx_k + Du_k \tag{D.1b}$$

Where  $x \in \mathbb{R}^n$  is the state vector,  $u \in \mathbb{R}^m$  the input vector,  $y \in \mathbb{R}^1$  the output vector, and  $A \in \mathbb{R}^{n \times n}$ ,  $B \in \mathbb{R}^{n \times m}$ ,  $C \in \mathbb{R}^{1 \times n}$ ,  $D \in \mathbb{R}^{1 \times m}$  are constant matrices.

With a sampling time of  $\Delta t = 1$  min:

$$A = \begin{bmatrix} 0.8825 & 0 & 0 & 0 & 0 \\ 0 & 0.9355 & 0 & 0 & 0 \\ 0 & 0 & 0.9048 & 0 & 0 \\ 0 & 0 & 0 & 0.8187 & 0 \\ 0 & 0 & 0 & 0 & 0.9355 \end{bmatrix} \quad (\text{D.2a})$$

$$B = \begin{bmatrix} 0.94 & 0 & 0 & 0 & 0 \\ 0 & 0.9674 & 0 & 0 & 0 \\ 0 & 0 & 0.9516 & 0 & 0 \\ 0 & 0 & 0 & 0.9063 & 0 \\ 0 & 0 & 0 & 0 & 0.2418 \end{bmatrix} \quad (\text{D.2b})$$

$$C = [-0.625 \quad -0.67 \quad 1.20 \quad 1.60 \quad 0.27] \quad (\text{D.2c})$$

$$D = [0 \quad 0 \quad 0 \quad 0 \quad 0] \quad (\text{D.2d})$$

Here,  $n = 5$ ,  $m = 5$ ; the states ( $x$ ) do not have a physical meaning and the input  $u_k$  in Eq. (D.1) contains the manipulated variables (Eq. 6.3b) and the disturbance. In practice, only the room temperature is measured, but for simplicity we assume that we have a perfect estimator so that we can have full state feedback.

Once that the system is discretized, the MPC problem can be formulated as:

$$\min \sum_{k=1}^N w (T_k - T_k^{sp})^2 + \sum_{k=1}^N Qu^2 \quad (\text{D.3a})$$

$$\text{s.t. discretized model, Eq. (D.1) and (D.2)} \quad (\text{D.3b})$$

$$u^{min} = 0 \leq u_k \leq u^{max} = 1 \quad \forall k \in \{1, \dots, N\} \quad (\text{D.3c})$$

The objective function includes a term for temperature ( $y = T$ ) setpoint tracking as well as a term penalizing the use of the manipulated variables ( $u$ ). The following weights were selected:  $w = 50$ ,  $Q = \text{diag}(5, 1, 1, 5)$ .

The prediction horizon is set to 100 min, and the control horizon to 5 min. The problem is solved using the KWIK algorithm (Schmid and Biegler, 1994).



# Supporting information for Chapter 7 - Controllers with different setpoints

## E.1 DAE model for room temperature

The room temperature can be described with the following differential-algebraic system of equations:

$$mC_p \frac{dT}{dt} = \alpha(T^{amb} - T) + \dot{Q} \quad (\text{E.1a})$$

$$m_{fl}C_{p,fl} \frac{dT_{fl}}{dt} = \dot{Q}_{HW} - \dot{Q}_{fl} \quad (\text{E.1b})$$

$$\dot{Q} = \dot{Q}_{fl} + \dot{Q}_{EH} - \dot{Q}_{AC} \quad (\text{E.1c})$$

$$\dot{Q}_{fl} = U_{fl}A_{fl}(T_{fl} - T) \quad (\text{E.1d})$$

$$\alpha = \dot{m}_v C_p + U_w A_w \quad (\text{E.1e})$$

The two states in this model are the room temperature ( $T$ ) and the floor temperature ( $T_{fl}$ ).  $T^{amb}$  is the ambient temperature (outside the room), which is the main disturbance. The main assumptions are: heat losses through the walls ( $U_w A_w (T^{amb} - T)$ ), constant ventilation flow which gives a heat loss ( $\dot{m}_v C_p (T^{amb} - T)$ ), constant heat capacities ( $C_{p,i}$ ), constant air mass inside the room ( $m$ ) and perfect mixing.

The size of the room is  $5m \times 10m$  (floor), with a height of  $3.33m$ , and with  $m/\dot{m}_v = 900s = 15min$  there are 4 changes of air per hour, which is within requirements for buildings (Osborn, 1985). Table E.1 shows the parameters for Eq. (E.1).

## E.2 Tuning parameters for each input

We use the SIMC tuning rules (Skogestad, 2003) to systematically tune the desired PI controllers for each input. We first identify a first-order plus time delay model for each input

$$G_i(s) = \frac{K_{p,i}}{\tau_i s + 1} e^{-\theta_i s} \quad \forall i \in \{1, 2, 3\} \quad (\text{E.2})$$

Table E.1: Parameters for room model.

Parameter	Description	Value	Units
$U_{fl}$	floor heat transfer coefficient	10	$W/(m^2 \text{ } ^\circ C)$
$A_{fl}$	floor area	50	$m^2$
$m_{fl}$	floor mass	600	$kg$
$C_{p,fl}$	floor heat capacity	1000	$J/(kg^\circ C)$
$C_p$	air heat capacity	1000	$J/(kg^\circ C)$
$m$	mass air in the room	180	$kg$
$\dot{m}_v$	ventilation flow rate	0.2	$kg/s$
$A_w$	wall area	100	$m^2$
$U_w$	wall heat transfer coefficient	2	$W/(m^2 \text{ } ^\circ C)$
$\alpha$	Eq. (E.1e)	400	$W/^\circ C$

Here,  $u_1 = \dot{Q}_{AC}$ ,  $u_2 = \dot{Q}_{HW}$  and  $u_3 = \dot{Q}_{EH}$ . Then, we select the desired closed loop time constant ( $\tau_{c,i}$ ) to calculate  $K_{C,i}$  and  $\tau_{I,i}$ :

$$K_{C,i} = \frac{\tau_i}{K_{p,i}(\tau_{c,i} + \theta_i)} \quad \forall i \in \{1, 2, 3\} \quad (\text{E.3a})$$

$$\tau_{I,i} = \min\{\tau_i, 4(\tau_{c,i} + \theta_i)\} \quad \forall i \in \{1, 2, 3\} \quad (\text{E.3b})$$

Skogestad Skogestad (2003) recommends to select  $\tau_{c,i} = \theta_i$  for tight control; but in many cases, slower tunings ( $\tau_{c,i} > \theta_i$ ) are used to reduce input usage and improve robustness (Grimholt, 2018).

We use the half-rule (Skogestad, 2003; Skogestad and Grimholt, 2012) to approximate the responses to first-order processes with time delay (Eq. (E.2)). Table E.2 shows the parameters of the transfer functions used to find the tuning parameters in Table E.3 using Eq. (E.3). Due to the floor dynamics, this is a second-order process. We should note that the magnitude of the gain for all inputs is  $(\alpha)^{-1} = 2.5^\circ C/kW$ . However, the transfer functions for  $u_1 = \dot{Q}_{AC}$  and  $u_3 = \dot{Q}_{EH}$  have positive numerator time constants that need to be approximated. Here we are using Rule T1a in Skogestad and Grimholt (2012) to approximate the transfer function.

Table E.2: Parameters for first-order transfer functions for the available inputs.

Input	$K_{p,i}$ ( $^\circ C/kW$ )	$\tau_{1,i}$ (s)	$\theta_i$ (s)
$u_1 = \dot{Q}_{AC}$	-8	2968	0
$u_2 = \dot{Q}_{HW}$	+2.5	3058	90
$u_3 = \dot{Q}_{EH}$	+8	2968	0

Table E.3: PI tuning parameters for the available inputs.

Input	$\tau_{c,i}$ (s)	$K_{C,i}$	$\tau_{I,i}$ (s)
$u_1 = \dot{Q}_{AC}$	300	-1.2367	1200
$u_2 = \dot{Q}_{HW}$	300	+3.1364	1560
$u_3 = \dot{Q}_{EH}$	300	+1.2367	1200

### E.3 Design of the split range controller

Table E.4 summarizes the information that describes the standard split range block for the system analyzed in the case study, Section 7.4.4. The slopes in Table E.4 are found with the procedure described in Reyes-Lúa et al. (2019b) using the tuning parameters in Table E.3. Here we consider  $v = 0$  (all inputs closed) at the nominal operating point.

Table E.4: Values for the slopes  $\alpha_i$ ,  $\Delta v_i$  and  $u_{i,0}$  in the split range block.

Parameter	$u_{AC}$	$u_{HW}$	$u_{EH}$
$\alpha_i$	-9.6829	24.5575	9.6829
$\Delta v_i$	0.4647	0.1222	0.4131
$u_i^0$	0	0	-1.4316

Note that  $\alpha_i$  in Table E.4 and Fig. E.1 are the slopes for the split range block described in Section 7.2 and are not related to  $\alpha$  in Eq. (E.1e) and Table E.1.

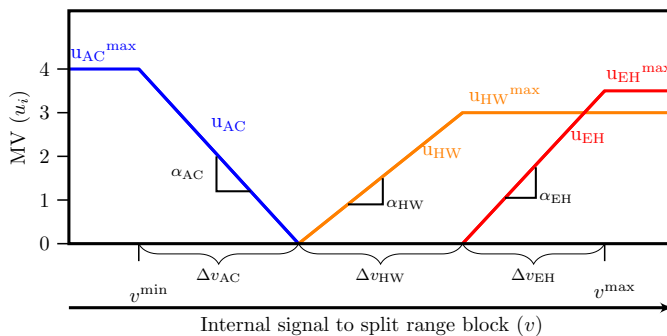


Figure E.1: Split range block for controlling room temperature with one source of cooling (AC) and two sources of heating (HW and EH). This is the SR block in Fig. 7.9.



# Active constraint switching with the generalized split range control structure using the baton strategy

Split range control is used to extend the steady-state operating range for a single output (controlled variable, CV) by using more than one input (manipulated variable, MV). In the context of optimal operation, this advanced control structure can be used for active constraint switching.

The generalized split range control structure introduced in Chapter 6 overcomes the limitations of standard split range control in terms of tuning by using multiple independent controllers with the same setpoint. It avoids undesired switching between the controllers using the baton strategy. In this contribution, we apply this novel control structure to a mixing process in which we must switch the MV used to control an important CV due to MV saturation. This is the same model used in Case Study I in Chapter 2, described in Appendix A.

Therefore, this paper builds upon the contributions of Chapters 2 and 6. It was submitted as:

Reyes-Lúa, A. and Skogestad, S. (2020). Active constraint switching with the generalized split range control structure using the baton strategy. In *21st IFAC World Congress (Submitted)*, Berlin, Germany. IFAC Papers Online

# Active constraint switching with the generalized split range control structure using the baton strategy

Adriana Reyes-Lúa\* Sigurd Skogestad\*

\* Department of Chemical Engineering, Norwegian University of Science and Technology (NTNU), Sem Sælands vei 4, 7491, Trondheim, Norway (e-mail: {adriana.r.lua, sigurd.skogestad}@ntnu.no)

**Abstract:** Split range control is used to extend the steady-state operating range for a single output (controlled variable, CV) by using more than one input (manipulated variable, MV). In the context of optimal operation, this advanced control structure can be used for active constraint switching. The generalized split range control structure analyzed in this paper overcomes the limitations of standard split range control in terms of tuning by using multiple independent controllers with the same setpoint. It avoids undesired switching between the controllers using the baton strategy. In this contribution, we apply this novel control structure to a mixing process in which we must switch the MV used to control an important CV due to MV saturation.

*Keywords:* split range control, control structure, PID, tuning, anti-windup, baton, multiple input, constraint switching, optimal operation

## 1. INTRODUCTION

We can use three alternative classical control structures when we need more than one manipulated variable (MV,  $u_i$ ) to cover the steady-state operating range for a single controlled variable (CV,  $y$ ):

- (1) (Standard) split range control
- (2) One controller for each input, each with a different setpoint for the output
- (3) Input (valve) position control

In the context of optimal operation, these structures can be used for active constraint switching, namely, *MV to MV constraint switching* and *MV to CV constraint switching* (Reyes-Lúa and Skogestad, 2019c). This is further discussed in Section 2.

Each structure has advantages and disadvantages. If we use more than one controller, the difference between setpoints ( $\Delta y^{sp}$ ) should be large enough to assure that only one controller is active at a time. With valve position control we cannot utilize the full steady-state range of the primary input as it requires a back-off. Therefore, split range control is often the chosen alternative<sup>1</sup>.

In this paper we focus on split range control, although the structure that we are using has more than one independent controller. Split range control has been in use for at least 75 years (Eckman, 1945)<sup>2</sup>, and it is still applied in industry (Sun et al., 2015). However, except for descriptions and examples of applications (see Stephanopoulos (1984);

Hägglund (1997); Marlin (2000); Bequette (2002); Seborg et al. (2003); Smith (2010)), there are almost no academic studies.

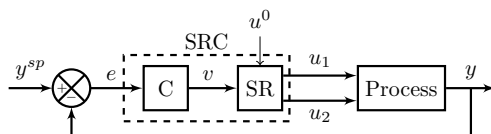


Fig. 1. Standard implementation of split range control (SRC) with two inputs ( $u_i$ ) and one output ( $y$ ). An SR-block is shown in Fig. 2.  $u_0$  contains information about maximum and minimum values for both physical inputs.

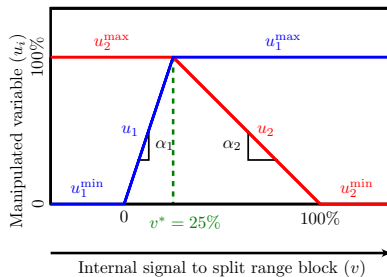


Fig. 2. Typical split range block for Fig.1, with  $v^* \neq 50\%$ .

In standard split range control (Fig. 1) the common controller ( $C(s)$ ) computes the internal signal ( $v$ ) to the

<sup>1</sup> In some problems, these alternative structures may actually be optimal. This is discussed in Reyes-Lúa and Skogestad (2019b)  
<sup>2</sup> Eckman (1945) called it "dual control agent".

split range block (SR), which assigns the value (e.g the valve opening) for each of the inputs ( $u_i$ ). As can be observed from the split range block in Fig. 2, the resulting controller from  $y$  to each  $u_i$  is  $\alpha_i C(s)$ .

In Reyes-Lúa et al. (2019), we proposed a systematic procedure to design a standard (classical) split range controller in which we select  $\alpha_i$  (or equivalently  $v^*$ ) such that  $\alpha_i K_C = K_{C,i}$ . However, there are intrinsic limitations for standard split range control in terms of tuning. Since we only have one design parameter for each MV ( $\alpha_i$ ), we cannot make  $\alpha_i C(s) = C_i(s)$  for all MVs. Thus, we must have a common integral time ( $\tau_I$ ) for all MVs, which is a compromise for dynamic performance.

To overcome this limitations, in Reyes-Lúa and Skogestad (2019a), we proposed a generalized split range control structure. This structure allows using multiple independent controllers with the same setpoint. Undesired switching between the controllers is avoided by using a *baton strategy*, and only one controller (and one MV) is active at a time. This is the structure that we use in this paper.

This paper is structured as follows. In Section 2 we describe in which cases we can use split range control for active constraint switching. In Section 3, we detail the generalized split range control structure recently introduced in Reyes-Lúa and Skogestad (2019a) and used in this paper. In Section 4 we apply the generalized split range control structure in a mixing process that requires active constraint switching for optimal operation. We finalize the paper with some concluding remarks in Section 5.

## 2. SPLIT RANGE CONTROL FOR ACTIVE CONSTRAINT SWITCHING

*Active constraints* are variables that should *optimally* be kept at their limiting value. These can be either *manipulated variable* (MV, input) constraints or *controlled variable* (CV, output) constraints. When a disturbance occurs, the set of active constraints may change and we might need to repair, that is to say, switch active constraints. Split range control is one of the advanced control structures that can be used for active constraint switching (Reyes-Lúa et al., 2018).

*MV to MV constraint switching* refers to the case in which the primary MV saturates and one or more extra MVs are added to cover the whole steady-state range and maintain control of the CV (Reyes-Lúa and Skogestad, 2019c). This is the most extended application of split range control, and corresponds to Fig. 1.

*MV to CV constraint switching* refers to the case in which there are the same number of MVs and CVs (any MV may be used to control any CV) and one of the MVs saturates (we loose one degree of freedom). In this case there are two possibilities:

- (1) The *input saturation pairing rule* (Minasidis et al., 2015) was followed. This means that, compared to other CVs, it is less important to control the CV ( $y_1$ ) paired with the MV that saturates ( $u_1$ ). Then, when we loose one degree of freedom we should give up controlling  $y_1$  (the least important CV).

- (2) The *input saturation pairing rule* was *not* followed. This means that there are other CVs that are less important to control compared to  $y_1$ , and we should not give up controlling  $y_1$ . Thus, when  $u_1$  saturates, we need to find another MV ( $u_2$ ) to control  $y_1$ . In this case, we give up controlling  $y_2$  (the CV previously controlled by  $u_2$ ). To do this, we can implement an *MV to MV switching strategy*, such as split range control, in combination with a *min/max selector* (Reyes-Lúa et al., 2018), as shown in Fig. 3.

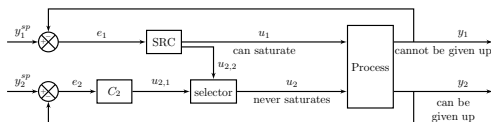


Fig. 3. MV to CV switching for the case when the input saturation rule is *not* followed; so control of  $y_1$  cannot be given up.

In Fig. 3, the SRC block can be either the standard split range structure in Fig. 1 or the generalized split range structure in Fig. 4.

## 3. GENERALIZED SPLIT RANGE CONTROLLER USING THE BATON STRATEGY

Fig. 4 depicts the generalized control structure for split range control structure proposed in Reyes-Lúa and Skogestad (2019a). Here, each input has its own controller  $C_i(s)$ , which can be any type of controller, but it is commonly a PI controller. Each controller produces a signal, which is a deviation variable ( $v_i = \Delta u_i$ ), and the *baton strategy block* selects and computes the physical inputs ( $u_i$ ), based on a predefined sequence.

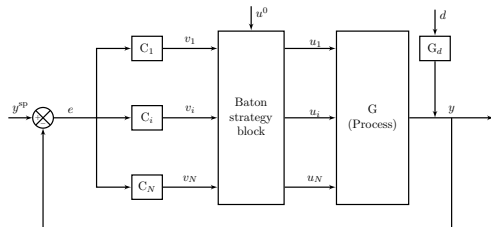


Fig. 4. Generalized split range control structure using the baton strategy. Note that  $v_i$  are deviation variables, whereas  $u_i$  are in physical units. Note here that  $u_i$  contains the bias information (maximum and minimum values for each input).

Importantly, at any given time, *only one input* ( $u_i$ ) is *actively controlling the output* ( $y$ ) and the other inputs are required to be at fixed values ( $u_k^{min}$  or  $u_k^{max}$ ). We call this *baton strategy* because we let the active input decide when to switch to another input (*pass the baton*). The *active input* ( $k$ ) remains active as long as its not saturated ( $u_k^{min} < u_k < u_k^{max}$ ) and will only *pass the baton* to another input once it becomes saturated (reaches  $u_k^{min}$  or  $u_k^{max}$ ).

As with standard split range control (Reyes-Lúa et al., 2019), as a first step to design the control structure, we need to:

- Define the minimum and maximum values for every MV ( $u_i^{\min}$ ,  $u_i^{\max}$ )
- Define the sequence in which we want to use the MVs. This is done considering the effect of each MV on the CV and economics (for MVs with the same effect, we use the least expensive MV first).

This sequence can be illustrated as in Fig. 2, but for the generalized split range controller the slopes have no significance.

### 3.1 Design of the baton strategy block

Consider that input  $k$  is the active input (*has the baton*). The baton strategy proposed in Reyes-Lúa and Skogestad (2019a) is then:

- B.1 Controller  $C_k$  computes  $u'_k = v_k + u_k^0$ , which is the suggested value for the input  $k$ .
- B.2 If  $u_k^{\min} < u'_k < u_k^{\max}$ 
  - (a) keep  $u_k$  active, with  $u_k \leftarrow u'_k$
  - (b) set the remaining inactive inputs at their corresponding constant values ( $u_i^{\min}$  or  $u_i^{\max}$ ).
- B.3 If  $u'_k \leq u_k^{\min}$  or  $u'_k \geq u_k^{\max}$ 
  - (a) Set  $u_k = u_k^{\min}$  or  $u_k = u_k^{\max}$  and *pass the baton* to the new active input  $j$ . The new active input is selected according to the predefined sequence, depending on which bound is reached ( $j = k + 1$  or  $j = k - 1$ ).
  - (b) Set  $k = j$  and go to step B.1. The value of the bias  $u_k^0$  is the input value just before receiving the baton, that is, either  $u_k^{\max}$  or  $u_k^{\min}$ .

We need to decide how to initialize the new active controllers and avoid windup. There are several alternatives. A simple strategy is to set all the states of the non active controllers to zero. For a PI controller (Eq. (1)), this means that the integral action starts at the time of the switching ( $t_k$ ).

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

Note that the integration in Eq. (1) starts from  $t_k$  and not from 0. Another alternative is to implement bumpless transfer (Åström and Hägglund, 2006).

## 4. CASE STUDY: MIXING OF AIR AND METHANOL

In a formaldehyde production process, air and methanol (MeOH) are mixed in a vaporizer. Air is fed using a blower with limited capacity. The main CV is the methanol molar fraction at the outlet of the vaporizer ( $y_1 = x_{MeOH}$ ) which should be kept at 0.10 (desired), and with a minimum value of 0.08 (more important), such that the reaction can take place. Additionally, we want to control the total mass flow ( $y_2 = \dot{m}_{tot}$ ), and in some cases to maximize it. This process is also described in Reyes-Lúa and Skogestad (2019c), and the model can be found in Appendix A.

The controlled variables (CVs) are:

- $y_1 = x_{MeOH}$ : MeOH molar fraction

- $y_2 = \dot{m}_{tot}$ : total mass flow

The two manipulated variables (MVs) are:

- $u_1 = \dot{m}_{air}^{sp}$ : mass flow of air
- $u_2 = \dot{m}_{MeOH}^{sp}$ : mass flow of methanol

Note that the physical MVs are the air blower rotational speed ( $\dot{\omega}_{air}$ ) and the MeOH valve opening ( $z_{MeOH}$ ), but we use a (lower) regulatory control layer with flow controllers which follow  $u_1 = \dot{m}_{air}^{sp}$  and  $u_2 = \dot{m}_{MeOH}^{sp}$ .

Table 1 shows the constraints and nominal operating conditions. Note that the valve for  $u_2 = \dot{m}_{MeOH}$  is not limited, and only  $y_1 = x_{MeOH}$  and  $u_1 = \dot{m}_{air}$  have relevant boundaries.

Table 1. Maximum and nominal values for case study.

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

As the main CV is  $y_1 = x_{MeOH}$ , it has a higher priority to maintain

$$x_{MeOH} = x_{MeOH}^{sp}; \quad \text{setpoint for } y_1 \quad (2)$$

compared to

$$\dot{m}_{tot} = \dot{m}_{tot}^{sp}; \quad \text{setpoint for } y_2 \quad (3)$$

At the nominal operating point (defined in Table 1), we are able to satisfy all the constraints. Due to upstream plant conditions, we pair  $u_1 = \dot{m}_{air}$  with  $y_1 = x_{MeOH}$  and  $u_2 = \dot{m}_{MeOH}$  with  $y_2 = \dot{m}_{tot}$ . As  $u_1 = \dot{m}_{air}$  is likely to saturate, we are *not following the input saturation pairing rule* with this pairing.

When  $u_1 = \dot{m}_{air}$  reaches its maximum value ( $u_1 = u_1^{max}$ ) and we then lose a degree of freedom, we must give up controlling  $y_2$  (constraint (3)) to keep controlling  $y_1$  (constraint (2)). To do this, we must implement an MV to CV switching strategy (Fig. 3). Fig. 5 shows the solution using split range control with a *min* selector in this process.

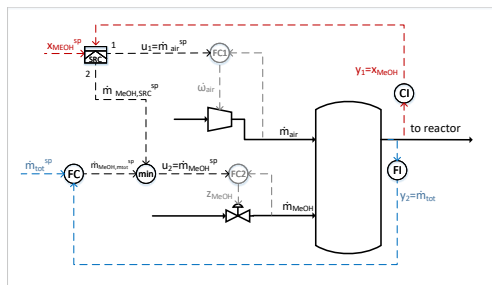


Fig. 5. Control structure for mixing of MeOH and air when *not following the input saturation pairing rule* using split range control (SRC) with a *min* selector.

The split range block (SRC) in Fig. 5 can be a standard split range controller or a generalized split range controller.



For both alternatives, we realize that available inputs have opposite effects on  $y_1 = x_{MeOH}$  (see  $K_{p,i}$  in Table A.1); that is,

- increasing  $u_1 = \dot{m}_{air}^{sp}$  decreases  $x_{MeOH}$ , and
- increasing  $u_2 = \dot{m}_{MeOH}$  increases  $x_{MeOH}$ .

Considering the pairing in Fig. 5, we use first use  $u_1 = \dot{m}_{air}^{sp}$  to control  $y_1 = x_{MeOH}$ . When  $u_1 = \dot{m}_{air}^{max}$ , we start using  $u_2 = \dot{m}_{MeOH}^{sp}$  to control  $y_1 = x_{MeOH}$  and we give up controlling  $y_2 = \dot{m}^{tot}$ .

The value of  $u_2 = \dot{m}_{MeOH}$  at the time of the switch (when  $u_1 = u_1^{max} = \dot{m}_{air}^{max}$ ) is not fixed, and it depends on the setpoint for  $x_{MeOH}$  ( $x_{MeOH}^{sp}$ ). Therefore, to improve the dynamic performance, the bias for  $\dot{m}_{MeOH}$ ,  $u_{MeOH}^0$ , should be updated to the value of  $\dot{m}_{MeOH}$  at the time of the switch. This value can also be obtained from the steady-state mass balance, as described in Appendix B.

The desired PI tuning parameters for both MVs to control  $y_1$  are in Table 2. These are obtained using the transfer functions in Table A.1 and the SIMC tuning rules (Skogestad, 2003), where  $\tau_c$  is the desired closed loop time constant.

Table 2. Desired PI tuning parameters for  $u_1$  and  $u_2$  to control  $y_1$ .

Controller (Fig. 8)	MV ( $u_i$ )	$\tau_{c,i}$ (s)	$K_{C,i}$	$\tau_{I,i}$ (s)
$C_1$	$u_1 = \dot{m}_{air}$	$\theta_{air}$	-74360	2.83
$C_2$	$u_2 = \dot{m}_{MeOH}$	$2\theta_{MeOH}$	10736	1.26

#### 4.1 Standard split range controller

Fig. 6 shows the configuration of the standard split range controller for the SRC in Fig. 5. Using the procedure introduced in Reyes-Lúa et al. (2019), we obtain  $\alpha = [-29874; 4313]$  and the tuning parameters for the common PI controller  $K_C = 2.5$ ,  $\tau_I = 2.83$  s.

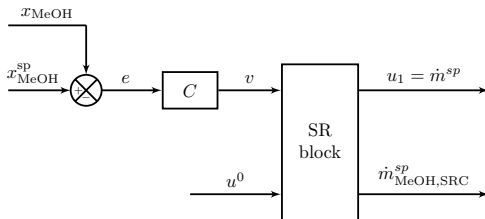


Fig. 6. Standard split range control solution for mixing of air and formaldehyde. This structure can be used in the SRC block in Fig. 5. The SR-block is shown in Fig. 7.

We should note that in this case, we do not need input tracking (anti-windup) for the split range controller ( $C$  in Fig. 6) because  $y_1 = x_{MeOH}$  is always being controlled; that is, the selected signal in the split range controller will always be active. Anti-windup is implemented for the flow controller for  $y_2 = \dot{m}^{tot}$ , as it will wind up during the period in which it is not selected and we give-up controlling  $y_2 = \dot{m}^{tot}$ .

Fig. 7 shows the split range block for the standard split range configuration for mixing of air and MeOH. The MVs are not scaled, and we can see the opposite effects of  $u_1$  and  $u_2$  on  $y_1$ .

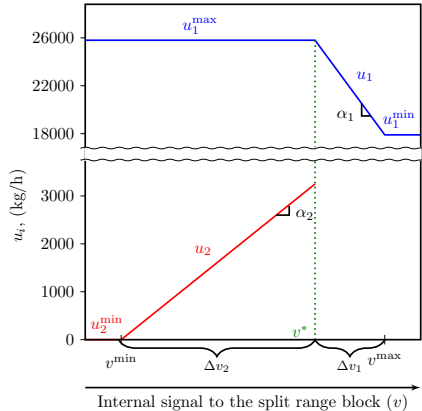


Fig. 7. Split range block for standard implementation of mixing of air ( $u_1$ ) and MeOH ( $u_2$ ). This is the SRC block in Fig. 6.

#### 4.2 Generalized split range controller

Fig. 8 shows the block diagram for the generalized split range controller.  $C_1$  and  $C_2$  are PI controllers (Eq. (1)), and the tunings are in Table 2. We consider Fig. 7 to define the sequence for the inputs. The baton strategy logic is written in Table 3.

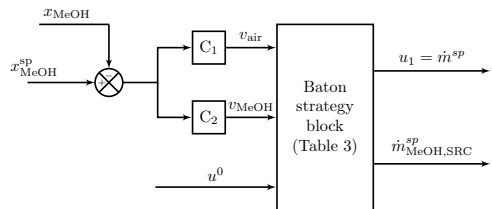


Fig. 8. Generalized split range control solution for mixing of air and MeOH. This structure can be used in the SRC block in Fig. 5.

When the input receives the baton, the integrator of its corresponding PI controller is reset, according to Eq. (1), and the initial value for  $u_i$  (at time  $t = t_k$ ) will be the proportional term plus the bias.

Note that in Table 3, when  $u_2 = \dot{m}_{MeOH}$  receives the baton, the initial value is  $u_2^0 = u_2^{max}$ . This is a generalization. However, as explained before and in Appendix B, this value is actually not fixed, and a better dynamic response is obtained if  $u_2^0$  is set to the value of  $u_2$  at the time of the switch or is calculated using Eq. (B.1).

Table 3. Baton strategy logic for mixing of air and MeOH;  $u_k$  is the active MV (the one that has the *baton*).

Value of $u_k$	$u_k = u_1 = \dot{m}_{air}$	$u_k = u_2 = \dot{m}_{MeOH}$
	baton to $u_2$	baton to $u_1$
$u'_k \geq u_k^{max}$	$u_1 \leftarrow u_1^{max}$ $u_2 \leftarrow u_2^{max}$	$u_1^0 = u_1^{max}$ $u_2 \leftarrow u_2^{max}$
	Keep $u_1$ active	Keep $u_2$ active
$u_k^{min} < u'_k < u_k^{max}$	$u_1 \leftarrow u'_1$ $u_2 \leftarrow u_2^{max}$ (see note below)	$u_2 \leftarrow u'_2$ $u_1 \leftarrow u_1^{max}$

As in standard split range control, anti-windup is implemented for the flow controller for  $y_2 = \dot{m}_{tot}$ .

*Simulations* We test the generalized split range structure and the standard structure with a step change in  $x_{MeOH}^{sp}$  of  $-0.005$  (from 0.1 to 0.095)  $t = 10$ s. at  $t = 30$ s,  $\dot{m}_{tot}^{sp}$  is increased by 10% (from 26860 kg/h to 29546 kg/h). Finally, at  $t = 70$ s  $\dot{m}_{tot}^{sp}$  is brought back to its initial value.

Fig. 9 shows the simulation results for both structures. We observe that both structures bring  $y_1 = x_{MeOH}$  to its set point at steady state. When  $y_1 = \dot{m}_{air}$  saturates and  $y_1 = x_{MeOH}$  is controlled using  $u_2 = \dot{m}_{MeOH}$  and we give up controlling  $y_2 = \dot{m}_{tot}$

However, at  $t = 30$  s, when  $u_1 = \dot{m}_{air}$  saturates, the response of  $y_1 = x_{MeOH}$  is clearly better with the generalized structure, with no overshoot in  $u_2$ . Likewise, when we can use again  $u_1$  to control  $y_1$ , the generalized structure keeps  $y_1$  closer to  $y_1^{sp}$ .

Table 4 shows that, with an improved dynamic response due to better tunings, the integral absolute error (IAE) for the high priority CV  $y_1 = x_{MeOH}$  decreases when using the generalized structure. The IAE for  $y_2 = \dot{m}_{tot}$  is expected to be high, as we give up controlling  $\dot{m}_{tot}$  when  $\dot{m}_{air}$  saturates ( $\dot{m}_{air} = \dot{m}_{air}^{max}$ ).

Table 4. Comparison of IAE with standard and generalized split range control for mixing of air and MeOH.

Case	IAE $x_{MeOH}$ (mol/mol)	IAE $\dot{m}_{tot}$ (kg/h)
Standard SRC	0.1623	59754
Generalized SRC	0.1082	60274

## 5. FINAL REMARKS

The generalized split range control structure using the baton strategy in Figs. 4 and 8 can be used in the same applications as standard split range control (Fig. 1). In this novel structure, each MV has its own controller, but only one MV (the one *with the baton*) is active at a time. This approach has the obvious advantage that once that the *baton strategy logic*, such as the one in Table 3, is implemented one can independently adjust the tunings for each MV and obtain the desired dynamic performance, without affecting the performance of the other the MVs.

In this paper, we implemented this structure in a mixing process that requires an *MV to CV constraint switching strategy* (split range control with *min* selector) to maintain control of an important CV. We compared simulation results with standard split range control, and showed that by having independent tunings for each MV, we can better handle switches in active constraints.

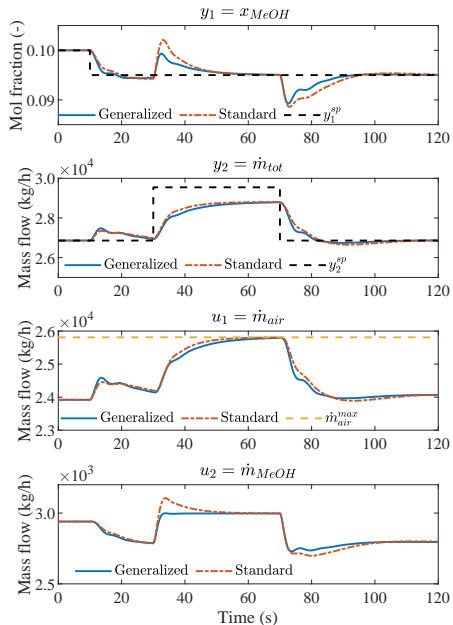


Fig. 9. Comparison of generalized and standard split range control for mixing of MeOH and air.

## REFERENCES

- Åström, K.J. and Hägglund, T. (2006). *Advanced PID Control*. ISA.
- Bequette, B.W. (2002). *Process Control: Modeling, Design, and Simulation*. Prentice-Hall.
- Eckman, D. (1945). *Principles of industrial control*. John Wiley & Sons, New York.
- Hägglund, T. (1997). *Praktisk processreglering*. Studentlitteratur, Lund, Sweden, 2nd edition.
- Marlin, T.E. (2000). *Process Control. Designing Processes and Control Systems for Dynamic Performance*. McGraw Hill.
- Minasidis, V., Skogestad, S., and Kaistha, N. (2015). Simple Rules for Economic Plantwide Control. In K.V. Gernaey, J.K. Huusom, and R. Gani (eds.), *12th PSE and 25th ESCAPE*, 101–108. Elsevier Science Direct. doi:10.1016/B978-0-444-63578-5.50013-X.
- Reyes-Lúa, A. and Skogestad, S. (2019a). Generalized split range control using the baton strategy. *Submitted to Journal of Process Control*.
- Reyes-Lúa, A. and Skogestad, S. (2019b). Multiple controllers with different optimal setpoints as an alternative to split range control. *Submitted to Processes*.
- Reyes-Lúa, A. and Skogestad, S. (2019c). Systematic design of active constraint switching using classical advanced control structures. *Industrial & Engineering Chemistry Research*, acs.iecr.9b04511. doi:10.1021/acs.

iecr.9b04511.

- Reyes-Lúa, A., Zotică, C., Forsman, K., and Skogestad, S. (2019). Systematic Design of Split Range Controllers. In *12th IFAC Symposium on Dynamics and Control of Process Systems, including Biosystems (DYCOPS)*. IFAC-Papers OnLine, Florianópolis, Brazil.
- Reyes-Lúa, A., Zotică, C., and Skogestad, S. (2018). Optimal Operation with Changing Active Constraint Regions using Classical Advanced Control. In *10th IFAC International Symposium on Advanced Control of Chemical Processes (ADCHEM)*, 434–439. IFAC Papers Online, Shenyang, Liaoning, China.
- Seborg, D.E., Edgar, T.F., and Mellichamp, D.A. (2003). *Process Dynamics and Control*. John Wiley & Sons, Inc., 2nd edition.
- Skogestad, S. (2003). Simple analytic rules for model reduction and PID controller tuning. *Journal of Process Control*, 13(4), 291–309. doi:10.1016/S0959-1524(02)00062-8.
- Smith, C.L. (2010). *Advanced Process Control. Beyond Single Loop Control*. John Wiley & Sons, New Jersey.
- Stephanopoulos, G. (1984). *Chemical Process Control: An Introduction to Theory and Practice*. Prentice-Hall.
- Sun, B.X., Shah, A., and Amalraj, J. (2015). A dual split-range control strategy for pressure and flow processes. *Control Engineering*. URL [www.controleng.com/articles/a-dual-split-range-control-strategy-for-pressure-and-flow-processes/](http://www.controleng.com/articles/a-dual-split-range-control-strategy-for-pressure-and-flow-processes/).

#### Appendix A. MODEL FOR MIXING OF METHANOL AND AIR

Eq. (A.1) and (A.2) describe the steady-state mass and molar balance for this system.

$$\dot{m}_{tot} = \dot{m}_{air} + \dot{m}_{MeOH} \quad (\text{A.1})$$

which corresponds to  $y_1 = u_1 + u_2$ .

$$x_{MeOH} = \frac{\dot{m}_{air}/MW_{air}}{\dot{m}_{air}/MW_{air} + \dot{m}_{MeOH}/MW_{MeOH}} \quad (\text{A.2})$$

where  $\dot{m}_i$  are the air and MeOH inlet mass flow rates (kg/h),  $\dot{m}_{tot}$  is the outlet total mass flow rate (kg/h), and  $x_{MeOH}$  is the methanol molar concentration (kmol/kmol), and  $MW_i$  are the methanol and air (average) molecular weights (kg/kmol).

Taking into account the dynamics of the actuators and the measurements, the dynamic responses can be approximated to the first order transfer functions with time delay ( $G_i = \frac{K_{p,i}e^{-\theta_i s}}{\tau_i s + 1}$ ) in Table A.1, which were identified using step-tests for each possible pairing.

Table A.1. First order transfer functions for mixing of MeOH and air.

$MV_i$	$CV_i$	$K_{p,i}$	$\tau_i$ (s)	$\theta_i$ (s)
$\dot{m}_{air}$	$x_{MeOH}$	-3.43E-05	2.83	0.37
$\dot{m}_{air}$	$\dot{m}_{tot}$	1.1229	2.90	0.56
$\dot{m}_{MeOH}$	$x_{MeOH}$	2.94E-04	1.26	0.20
$\dot{m}_{MeOH}$	$\dot{m}_{tot}$	9.14	3.80	1.15

#### Appendix B. BIAS CALCULATION FOR METHANOL FLOW

When  $\dot{m}_{air}^{sp} = \dot{m}_{air}^{max}$ , the value of  $\dot{m}_{MeOH}$  that will be required to get  $x_{MeOH} = x_{MeOH}^{sp}$  is a function of the current value of  $x_{MeOH}^{sp}$ . From the mass balance (Eq. (A.1)), when  $\dot{m}_{air} = \dot{m}_{air}^{max}$ , the  $\dot{m}_{MeOH}$  that satisfies the mass balance is:

$$\dot{m}_{MeOH}^0 = \dot{m}_{air}^{max} \left( \frac{MW_{MeOH}}{MW_{air}} \right) \left( \frac{x_{MeOH}^{sp}}{1 - x_{MeOH}^{sp}} \right) \quad (\text{B.1})$$

Updating  $u_2^0 = \dot{m}_{MeOH}^0$  improves the dynamic response of the system.

The bias update can also be done by setting the bias in the split range controller equal to the current value of  $\dot{m}_{MeOH}$  at the moment when  $u_2 = \dot{m}_{MeOH}$  receives the baton and starts controlling  $y_1 = x_{MeOH}$ .



## Appendix

# Inclusion of thermodynamic equations for efficient steady-state process optimization

This paper is published as:

Reyes-Lúa, A., Solvik, M., and Skogestad, S. (2016). Inclusion of thermodynamic equations for efficient steady-state process optimization. In *Proceedings of the 26th European Symposium on Computer Aided Process Engineering (ESCAPE)*, Portorož, Slovenia. Computer-aided chemical engineering

Figure 3b gives an example of active constraint regions for a different distillation column than the one analyzed in Chapter 2, with different disturbances.

## Inclusion of thermodynamic equations for efficient steady-state process optimization

Adriana Reyes-Lúa<sup>a</sup>, Marie Solvik<sup>a</sup> and Sigurd Skogestad<sup>a\*</sup>

<sup>a</sup>*Norwegian University of Science and Technology, Department of Chemical Engineering, Sem  
Sælands vei 4, 7491 Trondheim, Norway*  
*\*skoge@ntnu.no*

### Abstract

A two-layer approach, with a separate thermodynamic package, is common even for equation-oriented process simulators. However, this approach can be inefficient for optimization because the layer dedicated to the thermodynamic model needs to converge at each optimization step. An alternative is to solve the system in a truly equation-oriented mode by including the thermodynamic equations directly, together with the flowsheet model, to avoid solving for the roots of the thermodynamic model. Thermodynamic parameters that cannot be written explicitly, such as compressibility factors, are added as state variables. The main advantage is that we avoid unnecessary iterations and improve convergence of the optimization algorithm. The proposed approach was used to find the optimal active constraint regions for an industrial distillation column separating  $CO_2$  and ethane.

**Keywords:** optimization, mathematical modeling, equation-oriented

### 1. Introduction

In an equation-oriented (EO) approach for process simulation and optimization all equations describing the system are solved simultaneously. However, even when using an EO approach it is common to include the thermodynamic equations in a separate block. Indeed, using a specialized package for obtaining thermodynamic properties and solving equilibrium equations is robust and can be convenient for simulation and sensitivity studies. A drawback of solving thermodynamic equations separately is that an additional convergence point is required because the thermodynamic model needs to converge at every optimization step of the process model. Additionally, depending on the implementation, accurate gradient information may not be available for the optimization layer, leading to possible inaccuracies and convergence problems.

A way to overcome this issue is to include the thermodynamic equations in the overall model. Kamath et al. (2010) proposed a method to embed cubic equations of state for flowsheet optimization by adding inequality constraints on the first and second derivatives of the cubic equation of state. This approach uses the mathematical properties of cubic equations and assumes no previous knowledge of the equilibrium phases present in the system. Likewise, Skogestad (2008) explains the possibility to solve equilibrium calculations and the model differential equations simultaneously by adding state variables. Our approach is thought to be used for steady-state process optimization. Therefore, we exploit previous knowledge of existing equilibrium phases and solve the optimization problem by adding one equality constraint for each thermodynamic parameter that cannot be written explicitly.

## 2. Proposed Framework

As mentioned before, in most simulation and optimization frameworks, the thermodynamic model is separated from the process model in a two-layer scheme, as shown in Fig. (1a). Thermodynamic properties are usually calculated using a root solver and, in many cases, conditional statements.

We propose to remove the second layer by introducing state variables corresponding to the thermodynamic parameters that cannot be obtained explicitly. Correspondingly, one algebraic equation is added for each additional state. These additional equations are included in the overall model ( $\mathbf{h}(\mathbf{x}') = 0$ ), where  $\mathbf{x}'$  is an extended variable vector that includes states, inputs, and disturbances. Then, process and thermodynamic equations can be solved simultaneously, as shown in Fig. (1b).

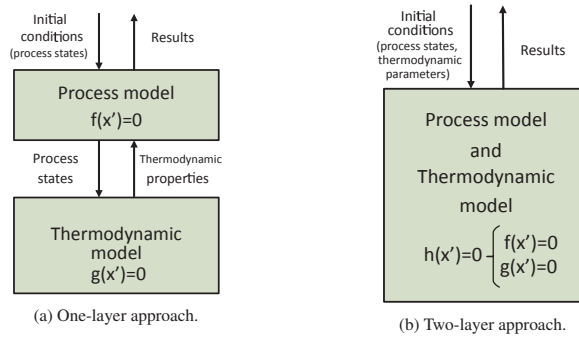


Figure 1: Approaches to solve process model and thermodynamic equations.

This approach requires a feasible or very close to feasible initial point for the process variables and thermodynamic parameters. In any case, if the model contains vapor-liquid equilibrium (VLE) stages, it is frequent to know beforehand which phases are present. Therefore, reasonable initial values for thermodynamic parameters can be provided to the optimization algorithm.

### 2.1. Use of SRK cubic equation of state within the proposed framework

Due to its simplicity and reasonable accuracy, the Soave-Redlich-Kwong (SRK) equation of state (EoS), proposed by Soave (1972), is one of the most popular EoS in simulations and optimizations in which vapor-liquid equilibrium (VLE) properties are required (Valderrama, 2003). The SRK EoS uses cubic Eq. (1a) to describe the compressibility factor  $Z$ , which relates molar volume ( $V_m$ ) with pressure ( $P$ ) and temperature ( $T$ ), as in Eq. (1b).

$$0 = Z^3 - Z^2 + (A - B - B^2)Z - AB \quad (1a)$$

$$Z = \frac{PV_m}{RT} \quad (1b)$$

Coefficients  $A$  and  $B$  in Eq. (1a) depend on critical temperature, critical pressure, reduced temperature, and the acentric factor of the fluid. The mathematical solution of cubic Eq. (1a) may be real or complex, whereas the compressibility factor must be real. Compressibility factors close to 1 correspond to vapor phase, while liquid phase has small compressibility factors. For this reason, it is common to solve Eq. (1a) separately from the process model and include conditional statements to select the appropriate real root depending on the fluid conditions. In our proposed framework, liquid and vapor compressibility factors ( $Z_L$  and  $Z_G$ ) are added as additional states. Correspondingly, Eq. (1a) is added for every phase present at each equilibrium stage.

### 3. Case Study

We analyze the optimization of a 3-component  $CO_2$ -stripper in which the bottom product is purified ethane and the distillate is a mixture of methane, ethane and  $CO_2$ . The SRK EoS is used to calculate the VLE information that is required to solve the model.

#### 3.1. $CO_2$ -stripper model

Fig. (2) shows the  $CO_2$ -stripper where the bottom product ( $B$ ) is purified liquid ethane (do not confuse with the SRK coefficient  $B$  in Eq. (1a)). The distillate flow ( $D$ ), which is burned to produce steam, is a mixture of  $CO_2$ , methane and ethane in vapor phase. The modeled column has 74 trays and 75 equilibrium stages, where the last stage is a partial condenser. Steady state degrees of freedom ( $\mathbf{u}$ ) are: reboiler duty ( $Q_r$ ), condenser duty ( $Q_c$ ) and inlet flow rate ( $F$ ). Typical disturbances ( $\mathbf{d}$ ) are the maximum available feed rate ( $F_{max}$ ) and  $CO_2$  concentration in the feed ( $z_{F,CO_2}$ ).

The DAE model of this column (Eq. 2) is based on the discussion made by Skogestad (1997, 2008). It was developed to investigate the changing dynamics in different operation points. It can also be used to study the steady-state behavior of the process and the effect of disturbances, as done in this paper.

The states ( $\mathbf{x}$ ) for simulation include, for each stage ( $j$ ): total stage molar hold-up ( $n_{T,j}$ ), component hold-up ( $n_{i,j}$ ), total internal energy ( $U_j$ ), temperature ( $T_j$ ), liquid volume ( $V_{L,j}$ ), liquid composition ( $x_{i,j}$ ), vapor composition ( $y_{i,j}$ ), pressure ( $p_j$ ), as well as liquid and vapor compressibility factors ( $Z_{L,j}$ ,  $Z_{G,j}$ ).

For every stage ( $j$ ), Eq. (2a) and Eq. (2b) represent the dynamic overall and component ( $i$ ) mass balance, while Eq. (2c) is the energy balance. Algebraic Eq. (2d) calculates the internal energy in each stage. Constant total volume ( $V_{tot}$ ) with varying molar hold-up is assumed. Then, Eq. (2e) is used to calculate  $V_{L,j}$  and component hold-up in Eq. (2f) is required to calculate  $x_{i,j}$ . Eq. (2g) relates composition in vapor and liquid phase for every stage. Composition information in Eq. (2h) is used to find  $p_j$ . Fugacity coefficients for  $K = \phi_L/\phi_G$  are calculated from the SRK EoS.

Eq. (2b), (2f) and (2g) are solved for  $(NC - 1)$  components at every stage. Additionally, we introduced corrections to the feed stage, reboiler, condenser and top tray.

$$\mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{d})_{\mathbf{j}} = 0 \begin{cases} 0 = \frac{dn_{T,j}}{dt} = V_{(j-1)} + L_{(j+1)} - V_j - L_j & \text{find } n_{T,j} \quad (2a) \\ 0 = \frac{dn_{i,j}}{dt} = V_{j-1}y_{i,(j-1)} + L_{j+1}x_{i,(j+1)} - V_jy_{i,j} - L_jx_{i,j} & \text{find } n_{i,j} \quad (2b) \\ 0 = \frac{dU_j}{dt} = V_{(j-1)}h_{G,(j-1)} + L_{(j+1)}h_{L,(j+1)} - V_jh_{G,j} - L_jh_{L,j} & \text{find } U_j \quad (2c) \\ 0 = U_j + p_jV_{tot,j} - h_{L,j}n_{L,j} - h_{G,j}n_{G,j} & \text{find } T_j \quad (2d) \\ 0 = n_j - n_{L,j} - n_{G,j} & \text{find } V_{L,j} \quad (2e) \\ 0 = n_{i,j} - x_{i,j}n_{L,j} - y_{i,j}n_{G,j} & \text{find } x_{i,j} \quad (2f) \\ 0 = y_{i,j} - K_{i,j}x_{i,j} & \text{find } y_{i,j} \quad (2g) \\ 0 = (1 - y_{n,j}) - K_{n,j}(1 - x_{n,j}) & \text{find } p_j \quad (2h) \end{cases}$$

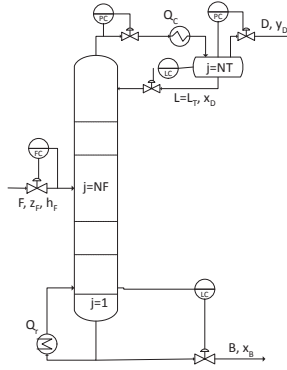


Figure 2: Simplified  $CO_2$  stripper



VLE information is required for both phases in every stage. For this reason, we included Eq. (3a) and (3b) as part of the system of equations, as proposed in this paper.

$$\mathbf{g}(\mathbf{x}, \mathbf{u}, \mathbf{d}) = 0 \begin{cases} 0 = Z_{G,j}^3 + Z_{G,j}^2 + Z_{G,j}(A_{G,j} - B_{G,j} - B_{G,j}^2) - A_{G,j}B_{G,j} & \text{find } Z_{G,j} \quad (3a) \\ 0 = Z_{L,j}^3 + Z_{L,j}^2 + Z_{L,j}(A_{L,j} - B_{L,j} - B_{L,j}^2) - A_{L,j}B_{L,j} & \text{find } Z_{L,j} \quad (3b) \end{cases}$$

Equation sets (2) and (3) are solved for each stage. Additionally, explicit algebraic equations describe the influence of pressure drop and liquid hold-up on column hydraulics. Vapor flow ( $V$ ) is constant through the column, depending on the pressure difference between the top and bottom stage. Liquid flows ( $L$ ) are assumed to be a linear function of the liquid hold-up and vapor flow. Stage efficiency is introduced to account for vapor bypassing. Given pressure, temperature, composition and compressibility, we also use explicit algebraic equations to calculate fugacity coefficients ( $\phi_L, \phi_G$ ), enthalpy ( $h_L, h_G$ ), molar volume, and the SRK coefficients in Eq. (1a).

The most important thing to note is that we have added the compressibilities on each stage ( $Z_{G,j}, Z_{L,j}$ ) as additional state variables. The optimizer will adjust them to satisfy Eq.(3). Thus, no root solver is needed, contrary to the case when the thermodynamic model is solved separately.

### 3.2. Optimization

The nonlinear optimization problem to be solved is:

$$\begin{aligned} \min_{\mathbf{x}, \mathbf{u}, \mathbf{d}} \quad & J(\mathbf{x}, \mathbf{u}, \mathbf{d}) = -P_D x_D D - P_B B + P_F F + P_E E \\ \text{s.t.} \quad & \mathbf{h}(\mathbf{x}, \mathbf{u}, \mathbf{d}) = 0 \\ & \mathbf{lb} \leq \mathbf{c}(\mathbf{x}, \mathbf{u}, \mathbf{d}) \leq \mathbf{ub} \end{aligned} \quad (4)$$

As explained in Fig.(1b),  $\mathbf{h}(\mathbf{x}, \mathbf{u}, \mathbf{d})$  includes  $\mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{d})$  and  $\mathbf{g}(\mathbf{x}, \mathbf{u}, \mathbf{d})$ . In the objective function ( $J(\mathbf{x}, \mathbf{u}, \mathbf{d})$ ), the price ( $P_B$ ) for the bottom flow is higher than for the distillate ( $P_D$ ).  $x_D$  is the ethane concentration in the distillate, which is used for steam production.  $F$  is the feed flow rate, and  $E$  is the required energy to provide the reboiler duty ( $Q_r$ ).

Feed rate, reboiler and condenser duties are the degrees of freedom for operation ( $\mathbf{u}$ ). If the model was to be used for dynamic simulation, these should be given or calculated. To use the model for optimization, they were included as decision variables for the optimization algorithm. Considering  $\mathbf{x}$  and  $\mathbf{u}$ , the optimization problem for the  $CO_2$ -stripper has 980 decision variables and the same number of initial values has to be provided to the optimization algorithm.

Inequality constraints were included to account for quality parameters such as the  $CO_2$  concentration in the bottom flow ( $x_B$ ) and the heating value of the distillate, represented by the Wobbe index ( $WI$ ). Inequality constraints for operational parameters such as: inlet flowrate ( $F$ ), reflux ( $LT$ ), bottom flow ( $B$ ), pressure drop ( $\Delta p$ ), steam production using distillate ( $SP$ ), and reboiler duty ( $Q_r$ ) were also included.

$$\mathbf{lb} \leq \mathbf{c}(\mathbf{x}, \mathbf{u}, \mathbf{d}) \leq \mathbf{ub} \begin{cases} 0 \leq x_B \leq x_{B,max} & (5a) \\ WI_{min} \leq WI \leq WI_{max} & (5b) \\ 0 \leq F \leq F_{max} & (5c) \\ LT_{min} \leq LT \leq LT_{max} & (5d) \\ 0 \leq B \leq B_{max} & (5e) \\ \Delta p_{min} \leq \Delta p \leq \Delta p_{max} & (5f) \\ 0 \leq SP \leq SP_{max} & (5g) \\ Q_{r,min} \leq Q_r \leq Q_{r,max} & (5h) \end{cases}$$

We performed steady-state optimization for 17 values of maximum feed rate ( $F_{max}$ ) and 15 values of  $CO_2$  concentration in the feed ( $z_{F,CO_2}$ ), which are the most relevant disturbances. This gives 255 optimization points. Matlab `fmincon` active-set solver, a medium-scale algorithm for nonlinear programming which solves the Karush-Kuhn-Tucker (KKT) equations, was used for optimization (MathWorks, 2015). Initial conditions for the 980 decision variables for optimization were obtained finding the steady-state solution of the dynamic problem. A tolerance of  $1 \times 10^{-9}$  for the objective and constraints was used. Average CPU time for optimization was 121 seconds using an i7 processor (2.7 GHz), with a mean of 12 iterations per optimization.

### 3.3. Results

In this type of analysis it is useful to identify the effect of disturbances on the cost function and active constraints. Fig. (3a) shows the contour plot of the cost function. Fig. (3b) shows the active constraint regions, identified in Table (1). Cost is minimized when feed rate is increased, until an economic bottleneck is reached (thicker line in Fig. (3b)). In regions D, E, F constraint (5c) ( $F \leq F_{max}$ ) is not active. This means that it is not optimal to process all the available feed, so we have exceeded the economic bottleneck. Regions D, E and F are relevant when the feedrate is given as  $F = F_{max}$ ; that is, when the feedrate is a disturbance rather than a degree of freedom.

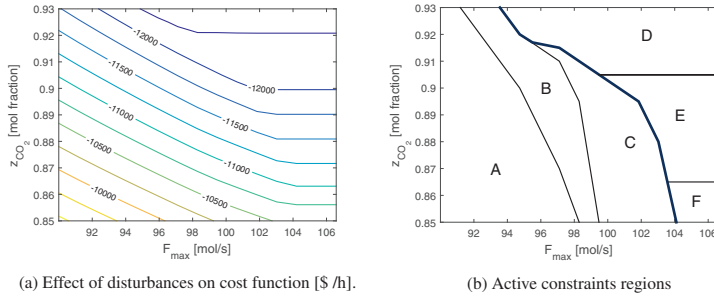


Figure 3: Optimization results for  $CO_2$ -stripper.

As expected, there is always at least one active quality constraint. At low  $F_{max}$ , in regions A and B, Wobbe index ( $WI$ ) is at its minimum. In regions C, E and F, with medium and high  $F_{max}$ , but medium and low  $CO_2$  concentration in the feed,  $x_{B,max}$  is activated. If  $CO_2$  concentration in the bottom product is at its maximum, ethane "give-away" is avoided (overpurifying costs energy). In regions D and E,  $WI$  is at its maximum. In region D, with high  $CO_2$  concentration in the inlet, also the maximum bottom flow becomes active. Therefore, the most valuable product flow cannot be increased. In regions C, E, and F, reboiler duty ( $Q_{r,max}$ ) activates. In region F, the column reaches an operational and a quality maximum, with  $Q_{r,max}$  and  $x_{B,max}$ , while the distillate is used to produce as much steam as possible ( $SP_{max}$ ).

Table 1: Active constraints regions for  $CO_2$ -stripper

Region	Active Constraints
A	$WI_{min}, F_{max}$
B	$WI_{min}, F_{max}, x_{B,max}$
C	$x_{B,max}, F_{max}, Q_{r,max}$
D	$WI_{max}, B_{max}$
E	$x_{B,max}, WI_{max}, Q_{r,max}$
F	$x_{B,max}, SP_{max}, Q_{r,max}$

### 3.4. Discussion

In many models for optimization of process conditions it is necessary to consider rigorous thermodynamics in order to obtain physically sensible results. This adds complexity to the model and thermodynamic equations are commonly handled separately. In this paper we show that if the model is used for optimization and previous knowledge of the existing conditions is available, then it is possible to solve the problem by introducing state variables.

The  $CO_2$ -stripper that we analyze in this paper is one of such processes. We solved the system of equations in one layer and in a truly equation-oriented manner, also eliminating conditional statements. We achieved this by including the liquid and vapor compressibility as additional states, adding 150 states to the original model, two for each equilibrium stage. The resulting model is numerically robust, as it no longer includes the possibility to calculate complex roots. It is worth to mention that we first implemented this approach because this same model had problems converging when using a two-layer approach. We applied this method to avoid using a root solver to calculate compressibility factors, but it should be pointed out that the same approach could be applied for other thermodynamic parameters that cannot be written explicitly.

To solve the resulting sparse system of equations, an appropriate optimization algorithm should be used. The solver used in this study approximates the gradients by finite differences. However, the benefit of having the complete system of equations in one layer could be further exploited if an optimization solver that calculates exact gradients of the system of equations was used.

A downside of the equation-oriented approach for process optimization is that it requires an adequate initial point. However, for steady-state optimization of operation conditions, it is common to have previous knowledge of the process, such as the phases present in each stage. This is particularly true in the case of steady-state process optimization for control purposes.

## 4. Conclusion

Including the thermodynamic model by adding additional state variables has a good potential for improving convergence when optimizing systems described by parameters that cannot be written explicitly, particularly when appropriate initial conditions are available. We have applied this approach satisfactorily to the optimization of a  $CO_2$ -stripper model. The overall description of this column is a sparse system of equations that is successfully optimized using a standard active-set algorithm. Optimization for different disturbances was performed. The simulated behavior of the cost function, as well as the main operating parameters were explained in physical terms. A map of active constraint regions for disturbances was generated using the optimization results.

## References

- Kamath, R. S., Biegler, L. T., Grossmann, I. E., Dec 2010. An equation-oriented approach for handling thermodynamics based on cubic equation of state in process optimization. *Computers & Chemical Engineering* 34 (12), 2085–2096.  
URL <http://dx.doi.org/10.1016/j.compchemeng.2010.07.028>
- MathWorks, 2015. Constrained Nonlinear Optimization Algorithms: fmincon Active Set Algorithm.  
URL <http://se.mathworks.com/help/optim/ug/constrained-nonlinear-optimization-algorithms.html>
- Skogestad, S., Jan 1997. Dynamics and Control of Distillation Columns - A Critical Survey. *Modeling, Identification and Control: A Norwegian Research Bulletin* 18 (3), 177–217.  
URL <http://www.mic-journal.no/ABS/MIC-1997-1997-3.asp>
- Skogestad, S., 2008. Dynamic simulation with examples. *Systems with algebraic equations*. In: *Chemical and Energy Process Engineering*, 1st Edition. CRC Press, Ch. 11, pp. 316–321.  
URL <http://www.nt.ntnu.no/users/skoge/book-cep/>
- Soave, G., Jun. 1972. Equilibrium constants from a modified Redlich-Kwong equation of state. *Chemical Engineering Science* 27 (6), 1197–1203.  
URL [http://dx.doi.org/10.1016/0009-2509\(72\)80096-4](http://dx.doi.org/10.1016/0009-2509(72)80096-4)
- Valderrama, J. O., Apr 2003. The State of the Cubic Equations of State. *Industrial & Engineering Chemistry Research* 42 (8), 1603–1618.  
URL <http://dx.doi.org/10.1021/ie020447b>