



# Multi-input single-output control for extending the operating range: Generalized split range control 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

## ARTICLE INFO

### Article history:

Received 3 November 2019  
Received in revised form 1 May 2020  
Accepted 1 May 2020  
Available online xxxx

### Keywords:

Split range control  
Control structure  
PID  
Tuning  
Anti-windup  
MISO  
MPC

## ABSTRACT

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 paper, 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.

© 2020 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

## 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. [1–3]). 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. 1),
2. One controller for each input, each with a different setpoint for the output (Fig. 2),
3. Input (valve) position control (Fig. 3).

The strategies in Figs. 1–3 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. 1) has been in use for more than 75 years [4,5]. Some other names that have been used for split range control are *dual control agent* [4], *range extending* [6] and

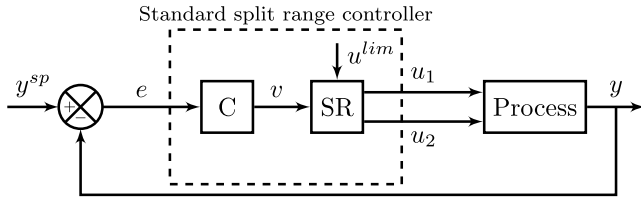
*valve sequencing* [7]. Split range control has been extensively applied in industry [7,8], but except for basic descriptions and examples of applications (see [3,9–16]), we have not found a systematic design procedure, and there are almost no academic studies. Therefore, in a previous paper [17], we proposed a systematic procedure to design the standard (classical) split range controller in Fig. 1. However, as we explain in Section 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. 2). For example, when controlling the temperature in a room ( $y = T$ ), one may use  $y^{sp,1} = 23$  °C as the setpoint for cooling ( $u_1$ ) and  $y^{sp,2} = 21$  °C as the setpoint for heating ( $u_2$ ) [18]. Then, on hot days, we use cooling ( $u_1$ ) and keep the temperature at  $y^{sp,1} = 23$  °C. If we have a disturbance in the outdoor temperature so that it decreases, say below 20 °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  $y^{sp,2} = 21$  °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 [14].

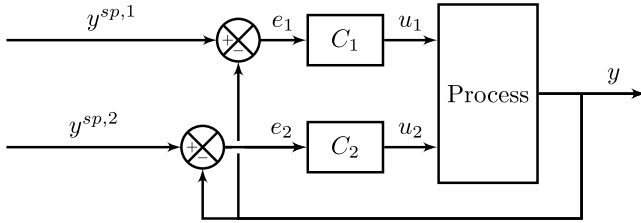
The third classical control structure for extending the steady-state range is shown in Fig. 3. Here, the output ( $y$ ) is always controlled with the same input ( $u_1$ ), but if  $u_1$  approaches its limit ( $u_1^{lim}$ ), then input  $u_2$  is activated and keeps  $u_1$  away from its limit.

\* Corresponding author.

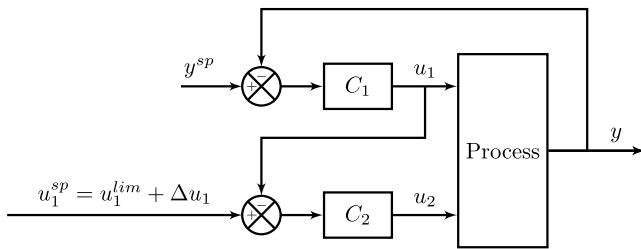
E-mail address: [sigurd.skogestad@ntnu.no](mailto:sigurd.skogestad@ntnu.no) (S. Skogestad).



**Fig. 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. 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.



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



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

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 ( $u_1$ ). *Comment:* the term “valve position control” is more commonly used for another case than in Fig. 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 paper, we propose a generalized split range control structure (Fig. 5), where the controller for each input can be designed independently. To avoid the use of different setpoints (Fig. 2), we use a *baton strategy*, in which undesired switching is avoided by allowing only one controller to be active at a time.

This paper is organized as follows. In Section 2 we briefly describe standard split range control and its limitations with respect to tuning. In Section 3 we present the new generalized structure, which overcomes these limitations. In Section 4 we use a case study to illustrate our proposed generalized structure and compare it to standard split range control. Then, in Section 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 paper in Section 6.

## 2. Standard split range controller

As shown in Fig. 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.

### 2.1. The split range block

The split range block has also been called *characterization function* [16], *splitter block* [19], and *function generator* [20]. Fig. 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. 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 ( $u_2$ ).

### 2.2. Slopes ( $\alpha_i$ ) in split range block

In Fig. 4(a), 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 [3,13,16,21–28]). 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. 4(b) (see [7,11,17,19,29–31]). 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  $u_i$ .

## 3. Generalized split range control structure

### 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. 1), represents a compromise. Fig. 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. 5 selects and computes the actual inputs ( $u_i$ ).

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), 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. 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  $u_i^{max}$ ).

<sup>1</sup> The desired tunings for these controllers can be found, for example, from the SIMC PID tuning rules [32].

<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 [32].

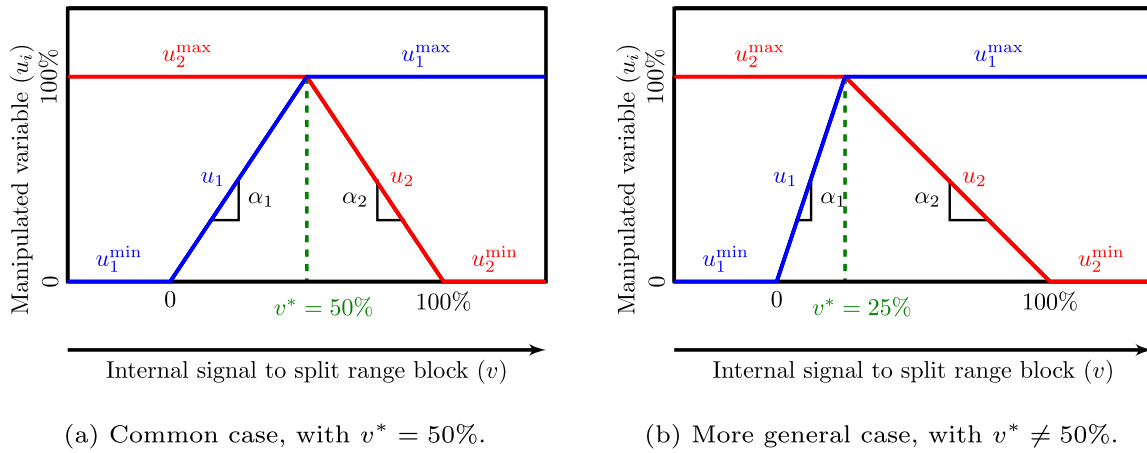


Fig. 4. Typical split range (SR) block for Fig. 1.

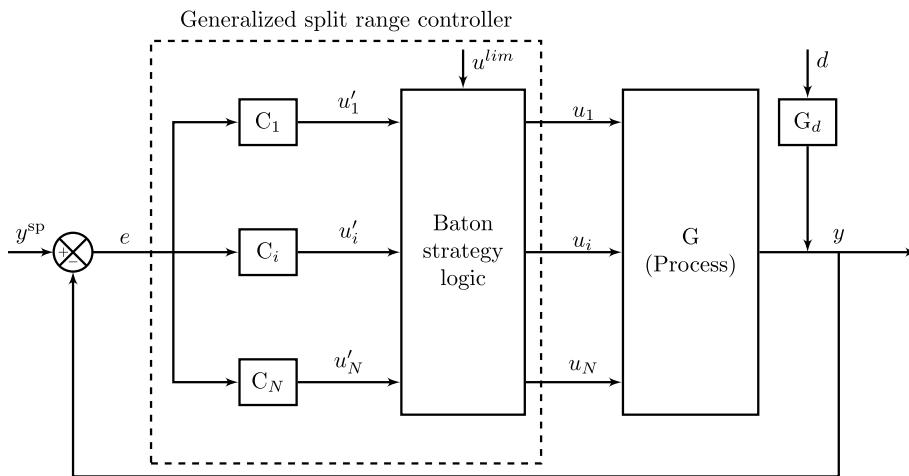


Fig. 5. Proposed generalized split range control using the baton strategy. Each controller computes a suggested input  $u'_i$  and the baton logic decides on the actual input  $u_i$ .



Fig. 6. Baton strategy for relay.

### 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. 4 (and Fig. 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.

### 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.

### 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. (1)), this means that the integral action starts at the time of the switching,  $t_b$ , when the baton is passed.

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

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. (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 5.4).

## 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 2.

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  $T^{sp} = 18$  °C. The main disturbance is ambient temperature ( $d = T^{amb}$ ), which is not measured and is nominally the same as the setpoint; thus,  $T_0^{amb} = 18$  °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.

### 4.1. Model

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

$$y(s) = G_p(s) u(s) + G_d(s) d(s) \quad (2)$$

where:

$$y = T \quad (3a)$$

$$u = [u_{AC} \ u_{CW} \ u_{HW} \ u_{EH}]^T \quad (3b)$$

$$d = T^{amb} \quad (3c)$$

$$G_p(s) = [G_{AC}(s) \ G_{CW}(s) \ G_{HW}(s) \ G_{EH}(s)] \quad (3d)$$

**Table 1**

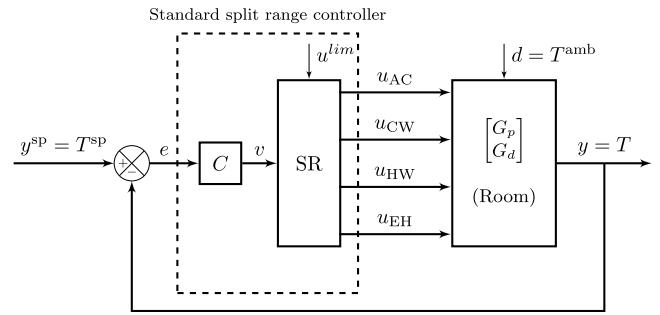
Parameters for  $G_{p,i}(s)$  from  $u_i$  to  $y = T$  and  $G_d(s)$  from  $d = T^{amb}$  to  $y = T$ .

$G_i$	$K_{p,i}$	$\tau_i$ (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

**Table 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



**Fig. 7.** Block diagram for standard split range control for room temperature control. The SR block is shown in Fig. 8.

**Table 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.

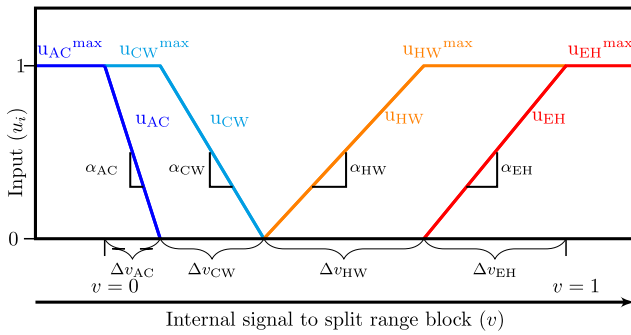
Note that since the gain for the disturbance in ambient temperature ( $d = T^{amb}$ ) is 1 and the inputs ( $u_i$ ) are scaled in the range 0 to 1, the gains  $K_{p,i}$  tell us the disturbance range that each input can handle. For example, since  $K_{p,HW} = 12$  we can handle ambient temperatures down to  $T^{amb} = T_0^{amb} - K_{p,HW} = 18$  °C - 12 °C = 6 °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 °C - 8 °C = -2 °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  $T_0^{amb} - K_{p,CW} - K_{p,AC} = 18$  °C + 5 °C + 10 °C = 33 °C before we lose control of  $y = T$  because both cooling sources (AC and CW) are at their maximum.

### 4.2. Standard implementation of split range control

**Fig. 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. 8**. For the common PI controller we choose  $K_C = 0.0592$  and  $\tau_I = 15$  min. **Table A.1** in **Appendix A.1** summarizes the parameters for the standard split range block in **Fig. 8**. The details about the design and tuning of this control structure can be found in [17].

### 4.3. Generalized implementation of split range control

**Fig. 9** shows the block diagram for the new proposed generalized split range control structure. We use PI controllers for



**Fig. 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. 7.

each input and tune each loop “tightly”, according to the SIMC tuning rules [32]. This is achieved by selecting the closed-loop time constant for each input equal to the time delay ( $\tau_{c,i} = \theta_i$ ). Table 2 gives the PI tuning parameters for each  $C_i(s)$ .

We next design the generalized split range control structure according to the procedure in Section 3.

#### 4.3.1. 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 ( $u_i = 0$ ).

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. 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  $u_i^{max}$ .

#### 4.3.2. Design of the baton strategy

We consider the block diagram in Fig. 9 and use Fig. 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 3.

When an input receives the baton, the integrator of its corresponding PI controller ( $C_k(s)$ ) is reset to zero, according to Eq. (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 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.

#### 4.3.3. Simulations

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

From Fig. 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  $t = 10$  min and  $t = 140$  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  $\tau_I = 15$  min, which is the same as for CW with generalized SRC (see Table 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.

## 5. Discussion

### 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 [11,12,33] 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. 1. For example, to change the PI-tunings from the original set 1 (in C) to the set 2, we may add a block  $K_{C,2}/K_{C,1}(1 + 1/\tau_2s)/(1 + 1/\tau_1s)$  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. 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 Wittenmark [33] and Hägglund [11] 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” to which two controllers it should give the baton if it reaches its maximum or minimum value, respectively.



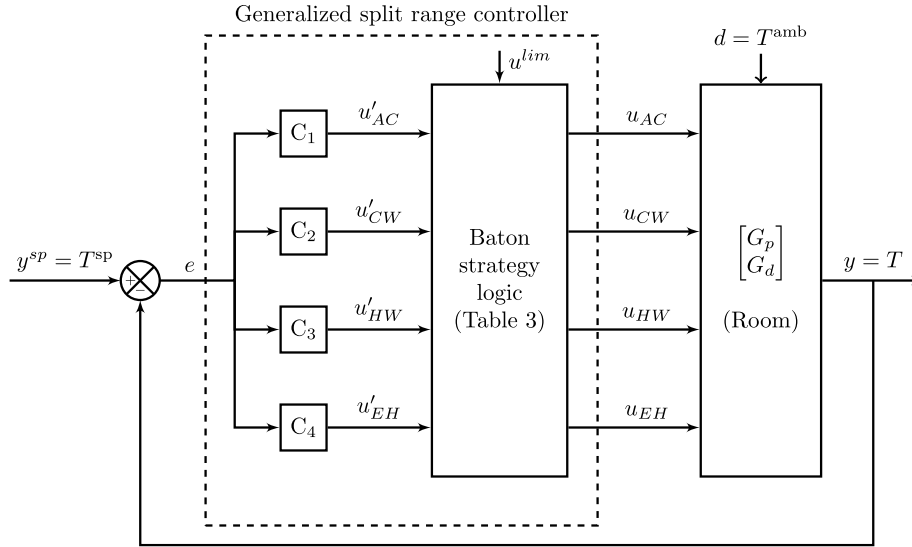


Fig. 9. Generalized split range control solution for room temperature control.

**Table 3**  
Baton strategy logic for case study.

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

### 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. 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.

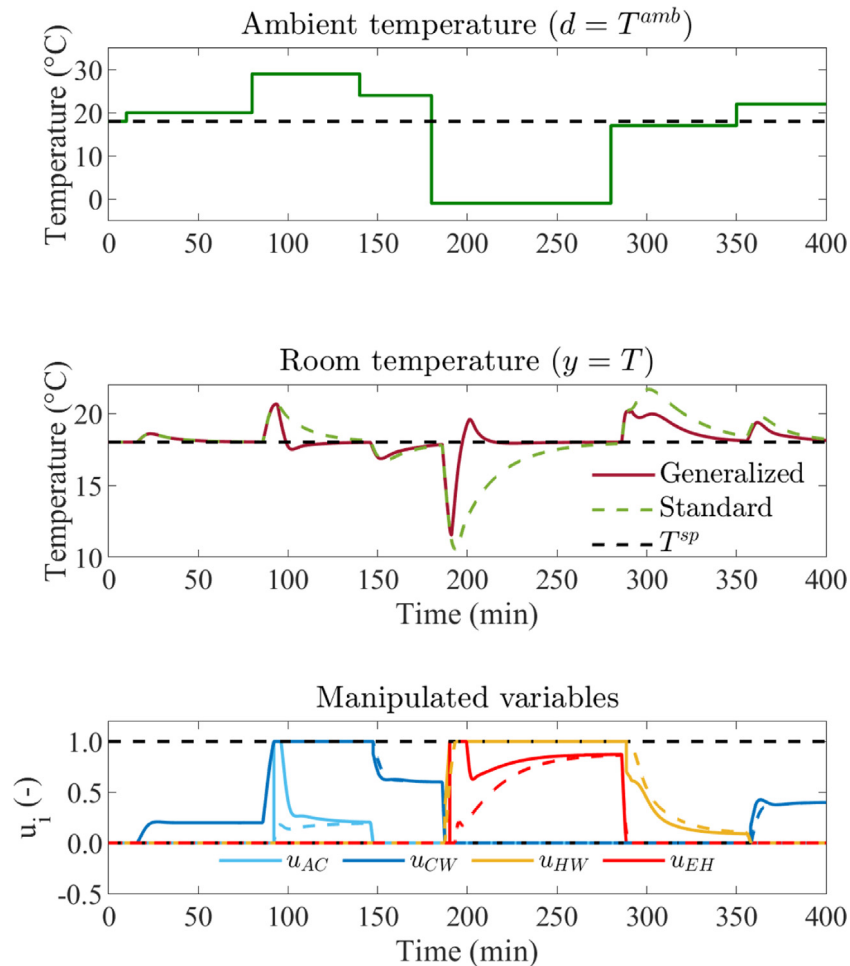
Our new generalized split range controller may be viewed as an extension of this, which avoids the use of different setpoints (Fig. 2). The use of different setpoints has the advantage of avoiding the logic block in Fig. 5, as the sequence of the inputs is indirectly given by the value of the setpoints. For example, for our room temperature case study, we could have used four controllers with setpoints 20 °C for AC, 19 °C for CW, 18 °C for HW and 17 °C for EH, assuming that we can have tight temperature control so that a setpoint difference of 1 °C is enough to avoid undesired switching.

### 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) [34]. 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. 11 compares the proposed generalized SRC with MPC. The generalized SRC is the same as the one studied earlier (Figs. 9 and 10) and the details of MPC are given in Appendix A.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 4. For example, consider the response at  $t = 280 \text{ min}$ , when  $d = T^{\text{amb}}$  increases from  $-1 \text{ }^\circ\text{C}$  to  $17 \text{ }^\circ\text{C}$  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.



**Fig. 10.** Comparison of standard and generalized split range controller (SRC) for the case study. The structure for standard SRC (dashed line) is shown in Fig. 7 and the structure for the generalized SRC (solid line) is shown in Fig. 9.

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.

#### 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. (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. 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.

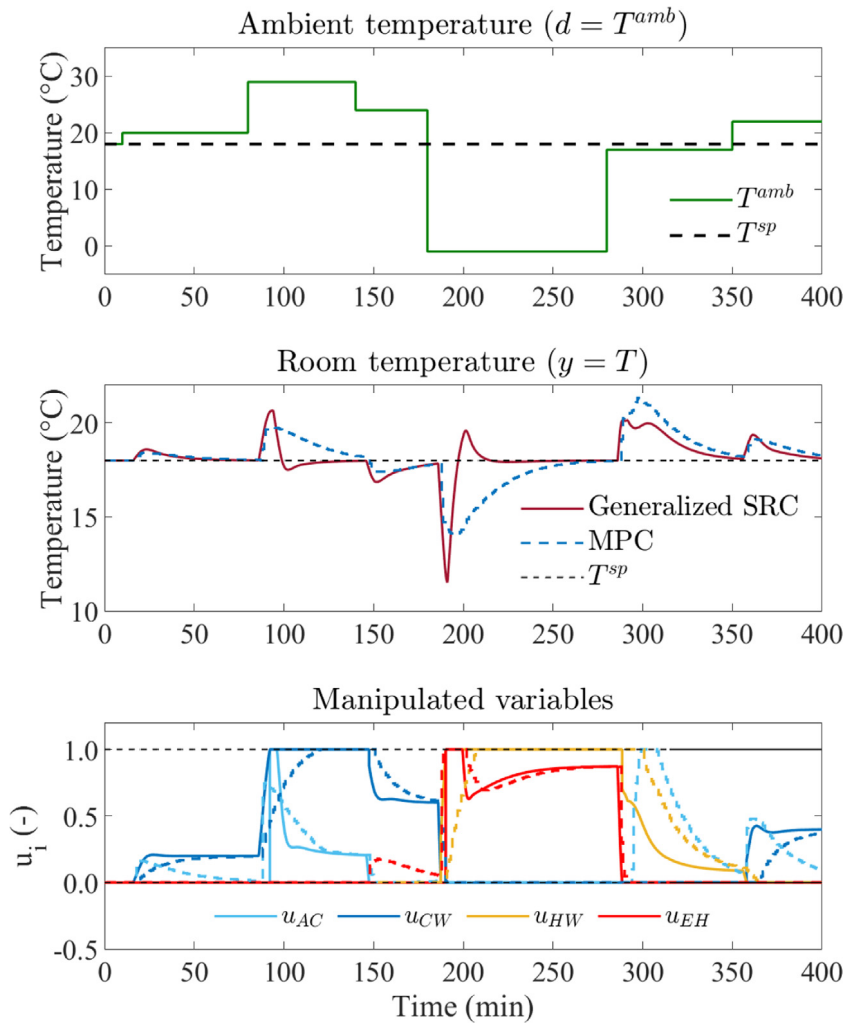
Windup can be avoided by implementing other anti-windup schemes, such as input tracking with *back-calculation* [14]. Fig. 12

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. 12, the tracking constant,  $K_T$ , [35] is used to reset the integrator dynamically [14]. 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 paper, 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.

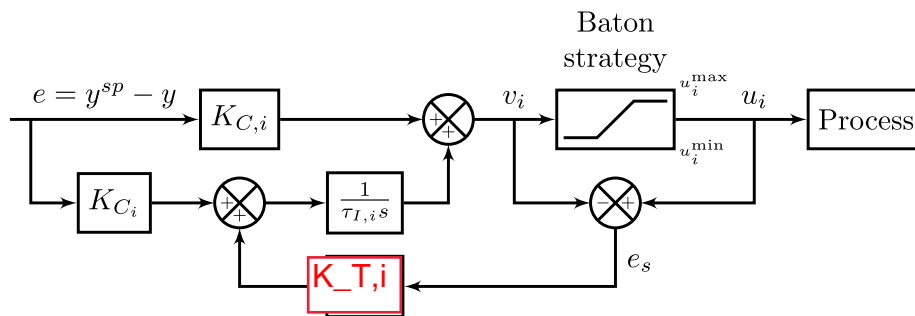
We implemented the back-calculation tracking scheme in Fig. 12, with  $K_T = 1$  for all inputs and with the same PI-tunings (Table 2) and switching logic (Table 3) as before. Fig. 13 compares back-calculation (dashed lines) with the strategy of integral resetting in Eq. (1) (solid lines). The differences 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 4).

#### 5.5. Stability for controllers extending the operating range

All the structures considered in this paper (Figs. 1–3 and 5) involve switching between different active controllers. During



**Fig. 11.** Comparison of standard MPC with generalized split range control for room temperature. The dashed lines correspond to MPC and the solid lines correspond to the strategy proposed in this paper, which is also depicted in Fig. 10.

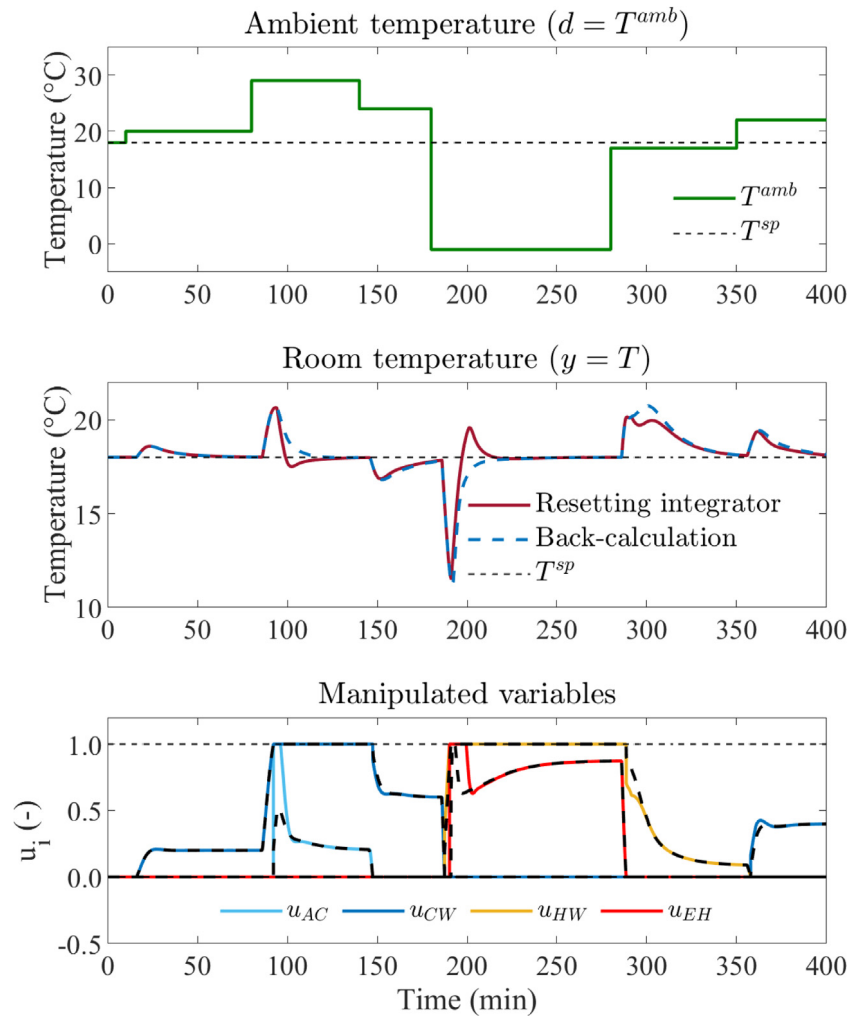


**Fig. 12.** Antiwindup with bumpless transfer: input tracking with back-calculation for input  $u_i$  used in Fig. 13.

**Table 4**  
Integral absolute error (IAE) for the case study with alternative controllers.

Controller	IAE
Standard split range control (Fig. 7): Fig. 10	448.6
Generalized split range control (Fig. 9), with integrator resetting (Eq. (1)): Figures 10, 11, and 13	202.4
Generalized split range control (Fig. 9), with back-calculation tracking (Fig. 12): Fig. 12	235.7
MPC (Eq. (6)): Fig. 11	327.9





**Fig. 13.** Comparison of two anti-windup strategies for the generalized split range control structure in Fig. 9. The dashed lines correspond to back-calculation (Fig. 12) and the solid lines correspond to the strategy of resetting the integrator (Eq. (1)), which is also depicted in Fig. 10.

normal operation, when there is no switching, we achieve robustness by using the SIMC PI tuning with  $\tau_c = \theta$  (Table 2), which guarantees a gain margin of about 3 and a delay margin of about 2.5 to 3 [32,36].

In general, switching may result in oscillations, and indeed, we encountered such problems with some of the other structures we tried, but not with our proposed structure (see Section 5.1). In practice, undesired oscillations may be overcome by introducing something on top of the switching, like adding a delay [12]. Another option is to use a two-step approach with a compensation (auxiliary) loop to avoid undesired switching. For example, Garelli et al. [37] 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. There exist no general analysis results for switched systems, for say analyzing whether undesired switching will occur and future theoretical work in this area will indeed be valuable for the implementation of switching strategies using advanced PID-based control structures.

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 paper. 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 [38] 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. 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. 5 and Table 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 in our case study.

**Table A.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

### CRedit authorship contribution statement

**Adriana Reyes-Lúa:** Conceptualization, Formal analysis, Investigation, Methodology, Software, Validation, Visualization, Writing - original draft, Writing - review & editing. **Sigurd Skogestad:** Conceptualization, Validation, Resources, Data curation, Writing - review & editing, Supervision, Funding acquisition.

### Declaration of competing interest

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

### Acknowledgment

The authors acknowledge Cristina Zotic for her insights for this paper.

### Appendix

#### A.1. Parameters for standard split range controller for case study

Table A.1 summarizes the information that describes the standard split range block in Fig. 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.

#### A.2. MPC implementation

To implement MPC for the system described in the case study, the dynamic optimization problem is set up using Matlab [39]. The transfer function model, Eq. (2), relating the inputs with the output, is converted to the discrete-time linear time-invariant (LTI) system described by:

$$x_{k+1} = Ax_k + Bu_k \quad (4a)$$

$$y_k = Cx_k + Du_k \quad (4b)$$

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:

$$A = \begin{bmatrix} 0.8825 & 0 & 0 & 0 & 0 \\ 0 & 0.9355 & 0 & 0 & 0 \\ 0 & 0 & 0.9048 & 0 & 0 \\ 0 & 0 & 0 & 0.8187 & 0 \\ 0 & 0 & 0 & 0 & 0.9355 \end{bmatrix} \quad (5a)$$

$$B = \begin{bmatrix} 0.94 & 0 & 0 & 0 & 0 \\ 0 & 0.9674 & 0 & 0 & 0 \\ 0 & 0 & 0.9516 & 0 & 0 \\ 0 & 0 & 0 & 0.9063 & 0 \\ 0 & 0 & 0 & 0 & 0.2418 \end{bmatrix} \quad (5b)$$

$$C = [-0.625 \quad -0.67 \quad 1.20 \quad 1.60 \quad 0.27] \quad (5c)$$

$$D = [0 \quad 0 \quad 0 \quad 0 \quad 0] \quad (5d)$$

Here,  $n = 5$ ,  $m = 5$ ; the states ( $x$ ) do not have a physical meaning and the input  $u_k$  in Eq. (4) contains the manipulated variables (Eq. (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:

$$\min \sum_{k=1}^N w (T_k - T_k^{sp})^2 + \sum_{k=1}^N Qu^2 \quad (6a)$$

$$\text{s.t. discretized model, Eq. (4) and (5)} \quad (6b)$$

$$u^{min} = 0 \leq u_k \leq u^{max} = 1 \quad \forall k \in \{1, \dots, N\} \quad (6c)$$

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 [40].

### References

- [1] P.S. Buckley, *Techniques of Process Control*, first ed., John Wiley & Sons, Inc., Delaware, 1964.
- [2] F. Shinskey, *Process Control Systems*, third ed., McGraw-Hill, 1988.
- [3] D.E. Seborg, T.F. Edgar, D.A. Mellichamp, *Process Dynamics and Control*, second ed., John Wiley & Sons, Inc., 2003.
- [4] D. Eckman, *Principles of Industrial Control*, John Wiley & Sons, New York, 1945, pp. 204–207.
- [5] E.D. Fink, *Monogram of Instruments and Process Control*, Delmar Publishers, Inc., Albany, NY, 1945, p. 120,149.
- [6] E. Bristol, After DDC idiomatic control, *Chem. Eng. Progress* 76 (11) (1980) 84–89.
- [7] B.G. Lipták, Control and on-off valves, in: *Instrument Engineers' Handbook*, second ed., Chilton Book Company, 1985, pp. 410–412.
- [8] B.X. Sun, A. Shah, J. Amalraj, A dual split-range control strategy for pressure and flow processes, *Control Eng.* (2015) [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/).
- [9] G. Stephanopoulos, Synthesis of control systems for chemical plants A challenge for creativity, *Comput. Chem. Eng.* 7 (4) (1983) 331–365, [http://dx.doi.org/10.1016/0098-1354\(83\)80018-0](http://dx.doi.org/10.1016/0098-1354(83)80018-0).
- [10] G. Stephanopoulos, *Chemical Process Control: An Introduction to Theory and Practice*, Prentice-Hall, 1984.
- [11] T. Hägglund, *Praktisk Processreglering*, second ed., Studentlitteratur, Lund, Sweden, 1997, p. 135.
- [12] T.E. Marlin, *Process Control. Designing Processes and Control Systems for Dynamic Performance*, McGraw Hill, 2000.
- [13] B.W. Bequette, *Process Control: Modeling, Design, and Simulation*, Prentice-Hall, 2002, p. 800.
- [14] K.J. Åström, T. Hägglund, *Advanced PID Control*, ISA, 2006.
- [15] N.P. Lieberman, *Troubleshooting Process Plant Control*, Wiley, 2008.
- [16] C.L. Smith, *Advanced Process Control. Beyond Single Loop Control*, John Wiley & Sons, New Jersey, 2010, p. 450.
- [17] A. Reyes-Lúa, C. Zotică, K. Forsman, S. Skogestad, 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, 2019*.
- [18] A. Reyes-Lúa, S. Skogestad, Multiple-input single-output control for extending the steady-state operating range—Use of controllers with different setpoints, *Processes* 7 (12) (2019) 941, <http://dx.doi.org/10.3390/pr7120941>, <https://www.mdpi.com/2227-9717/7/12/941>.
- [19] T.L. Blevins, *Control Loop Foundation: Batch and Continuous Processes*, International Society of Automation (ISA), 2011.
- [20] C.L. Smith, *Control of Batch Processes*, John Wiley & Sons, Incorporated, 2014.
- [21] S. Mercer, Cryogenics: A technological tool for space scientists, *Cryogenics* 8 (2) (1968) 68–78, [http://dx.doi.org/10.1016/0011-2275\(68\)90044-1](http://dx.doi.org/10.1016/0011-2275(68)90044-1).

- [22] R.M. Price, P.R. Lyman, C. Georgakis, Throughput manipulation in plantwide control structures, *Ind. Eng. Chem. Res.* 33 (5) (1994) 1197–1207, <http://dx.doi.org/10.1021/ie00029a016>.
- [23] M. Piovoso, K. Dahl, K. Kosanovich, Control of a batch reactor using a multivariate statistical controller design, *IFAC Proc. Vol.* 28 (9) (1995) 357–361, [http://dx.doi.org/10.1016/S1474-6670\(17\)47063-7](http://dx.doi.org/10.1016/S1474-6670(17)47063-7).
- [24] S. Wang, *Intelligent Buildings and Building Automation*, first ed., Spon Press, 2010.
- [25] R.R. Fonseca, J.E. Schmitz, A.M.F. Fileti, F.V. da Silva, A fuzzy-split range control system applied to a fermentation process, *Bioresour. Technol.* 142 (2013) 475–482, <http://dx.doi.org/10.1016/j.biortech.2013.05.083>.
- [26] D. Shen-Huii, Z. Gang, H. Mei-Rong, 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), IEEE, 2011, pp. 4350–4353, <http://dx.doi.org/10.1109/CECNET.2011.5768801>.
- [27] S.T. Fatani, R.S. Patwardhan, M.A.L. Andreu, A novel single-input two-output (SITO) strategy for split range control, in: 2017 6th International Symposium on Advanced Control of Industrial Processes (AdCONIP), IEEE, 2017, pp. 348–353, <http://dx.doi.org/10.1109/ADCONIP.2017.7983805>.
- [28] A. Bahadori, Control valves, in: *Oil and Gas Pipelines and Piping Systems*, Elsevier, 2017, pp. 483–571, <http://dx.doi.org/10.1016/B978-0-12-803777-5.00016-2>.
- [29] B. Glemmestad, *Optimal Operation of Integrated Processes - Studies on Heat Recovery Systems*, Norwegian University of Science and Technology (NTNU), 1997.
- [30] F.B. Bastani, I.-R. Chen, 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*, vol. 2, ACM Press, New York, New York, USA, 1988, pp. 1049–1058, <http://dx.doi.org/10.1145/55674.55724>.
- [31] P. Gupta, K. Rana, V. Kumar, P. Mishra, Split-range control of a Jacketed CSTR using self-tuning fuzzy PI controller, in: 2015 International Conference on Advances in Computer Engineering and Applications, IEEE, 2015, pp. 527–533, <http://dx.doi.org/10.1109/ICACEA.2015.7164750>.
- [32] S. Skogestad, Simple analytic rules for model reduction and PID controller tuning, *J. Process Control* 13 (4) (2003) 291–309, [http://dx.doi.org/10.1016/S0959-1524\(02\)00062-8](http://dx.doi.org/10.1016/S0959-1524(02)00062-8).
- [33] K.J. Åström, B. Wittenmark, *Adaptive Control*, second ed., Addison-Wesley Series in Electrical Engineering: Control Engineering, 1995.
- [34] S. Qin, T.A. Badgwell, A survey of industrial model predictive control technology, *Control Eng. Pract.* 11 (7) (2003) 733–764, [http://dx.doi.org/10.1016/S0967-0661\(02\)00186-7](http://dx.doi.org/10.1016/S0967-0661(02)00186-7).
- [35] MathWorks, Anti-windup control using a PID controller, in: *Matlab Documentation Center*, 2019, <https://se.mathworks.com/help/simulink/slref/anti-windup-control-using-a-pid-controller.html;jsessionid=0a5ce78fe614b664e892077af362>.
- [36] C. Grimholt, S. Skogestad, Optimal PI and PID control of first-order plus delay processes and evaluation of the original and improved SIMC rules, *J. Process Control* 70 (2018) 36–46, <http://dx.doi.org/10.1016/j.jprocont.2018.06.011>.
- [37] F. Garelli, J. Mantz, Ricardo, H. De Battista, *Advanced Control for Constrained Processes and Systems*, The Institution of Engineering and Technology. IET Control Engineering Series 75, 2011, p. 224.
- [38] A. Bemporad, M. Morari, Control of systems integrating logic, dynamics, and constraints, *Automatica* 35 (3) (1999) 407–427, [http://dx.doi.org/10.1016/S0005-1098\(98\)00178-2](http://dx.doi.org/10.1016/S0005-1098(98)00178-2).
- [39] MathWorks, Model predictive control toolbox, in: *Design and Simulate Model Predictive Controllers*, 2019, <https://se.mathworks.com/help/mpc/gs/control-of-a-multi-input-single-output-plant.html>.
- [40] C. Schmid, L. Biegler, Quadratic programming methods for reduced hessian SQP, *Comput. Chem. Eng.* 18 (9) (1994) 817–832, [http://dx.doi.org/10.1016/0098-1354\(94\)E0001-4](http://dx.doi.org/10.1016/0098-1354(94)E0001-4).