NTNU
Norwegian University of
Science and Technology

# Control strategy selection for optimal operation using split-range-controllers – A moving bottleneck case study

**Amos Yaoyang Fang**

Chemical Engineering

| | |
|---|---|
| Submission date: | Jan 2019 |
| Supervisor: | Krister Forsman, IKP |
| Co-Supervisor: | Sigurd Skogestad, IKP |

Norwegian University of Science and Technology

Department of Chemical Engineering

# Abstract

Classical advanced control structures (ACS) are widely used in the chemical industry to augment the control capability of traditional PID controllers. Yet surprisingly, except for a few accounts in literature, they are not widely mentioned and the theory remains less mainstream and standardized given that textbooks take a cursory approach, if any, in explaining the concepts behind these controllers. Even when concepts are described, they relate to design for normal process applications and not for optimal operation of processes. ACS may be used to extend the operating range for normal process applications or be implemented to facilitate optimal operation. The split-range-controller (SRC) yields better optimal control performance compared to other ACS like mid-ranging controllers or selectors as it does not need to operate with back-off from the optimum. The design procedures for economic optimal control, classical advanced control structures for optimal operation and the split-range controller from literature are studied and discussed. The goal is to adopt a strategy to design an optimal control structure which can mitigate the moving bottleneck problem in flow networks involving storage tanks. The SRC with selectors control design is developed to select the optimal operation mode for two tanks in series, given the bottleneck location. To assess the SRC performance, we introduced flow disturbance, one at a time, to all the bottleneck locations and observe how the controller takes action to maintain optimal operation. The control objective is to maximize or optimize flow throughput while holding inventory level constant vis-à-vis the two tanks. The results show that the SRC with selectors, using proportional-integral controllers, can perform control strategy selection for optimal operation. This is achieved by controlling the flow closest to the bottleneck at the bottleneck flow rate to maximize flow throughput or overriding with a throughput setpoint to operate at the optimized throughput rate. The proposed control structure design can meet the control objectives by performing control structure selection for optimal operation. The shortcomings are that the control structure is inflexible towards supply disruption, especially upstream, and cannot bypass serious bottlenecks when the controller is in automatic mode. Robust and tight inventory level control is achieved with the proportional-integral level SRC. On the other hand, the approach for stabilizing flow is to use the proportional-only level SRC. The proposed split-range and selectors control structure can systematically handle the changing active constraint region posed by the moving flow bottleneck, to achieve optimal operation.

**Keywords:** optimal operation, advanced control structures, active constraint regions, inventory control system

# Preface

This thesis is written for the fulfillment of the requirements of the *TKP4900 Chemical Process Technology, Master's Thesis* carried out at the Norwegian University of Science and Technology (NTNU). This work was carried out during the Fall of 2018 at the Department of Chemical Engineering.

I would like to thank my supervisors Professor Krister Forsman and Sigurd Skogestad and PhD candidates Adriana Reyes-Lúa and Cristina Zotică for their guidance and feedback over my 5 months spent working on the masters thesis. This project gave me the opportunity to review concepts I learned from the Process Control courses I have taken at NTNU and reflect on the tasks I used to perform while working as a process control engineer in Singapore. There were solitary moments when I spent thinking and rethinking about the problem and concepts, but I believe that the time of revelation will come. I appreciate my mentors' time discussing the problem with me and their hope that I don't screw up again.

Most importantly and above everything, I deliver my sincerest hope that my supervisor Krister Forsman will be blessed with good health.

**Declaration of Compliance**

*I declare that this is an independent work according to the exam regulations of the Norwegian University of Science and Technology (NTNU).*

**Amos Fang**
Trondheim
22 Jan 2019

# Acronyms

| Acronym | Full Naming |
|---|---|
| ACS | Classical Advanced Control Structures |
| APC | Advanced Process Control |
| CSTR | Continuously Stirred Tank Reactor |
| CV | Controlled Variable (output) |
| DCS | Distributed Control System |
| DOF | degrees-of-freedom |
| ISR | Input Saturation pairing Rule |
| MIMO | Multiple-input-multiple-output (controller) |
| MISO | Multiple-input-single-output (controller) |
| MV | Manipulated Variable (input) |
| OP | Output (controller) |
| PI | Proportional-Integral (controller) |
| PID | Proportional-Integral-Derivative (controller) |
| PV | Process Value |
| SIMC | Simple-Internal-Model-Control |
| SISO | Single-input-single-output (controller) |
| SP | Setpoint (controller) |
| SRC | Split-Range Controller |
| TPM | Throughput Manipulator |
| UDM | Utilities Disturbance Management |

# Contents

# List of Figures

*Dedicated to my parents who have supported my decision to venture out of my comfort zone*

# 1 — Introduction

*Why do we need a strategy to design control structures?*

Two reasons are cited on why we need a strategy.

The first reason lies with the performance of control loops found in industry. About 80% of industrial PID control loops are badly tuned according to Van Overschee and De Moor (2000). Although bad PID controller tuning is easily attributed to poor tuning parameters, it could be linked to deeper issues such as poorly defined control objectives and/or non-optimal control performance (Seborg et al., 2003, page 317). As bad tuning does not mean that the controllers are performing incorrectly, unmitigated process disturbances could be glossed over and regarded as operational or process issues. However, if the problem lies with more than just bad controller tuning, and can be traced back to a bad control structure design, then the approach to control structure design requires an evaluation. Poorly designed control structures will require more than necessary manual interventions from shift operators, leading to productivity loss or loss in (energy) efficiency of the plant. In some cases, poor control structure design could trigger frequent alarms that distract the operator from his main responsibilities. On the other hand, control structures designed for optimal operations can potentially[1] improve overall plant performance by reducing the number of non-optimal decisions made and managing variable disturbances. Without a systematic approach, it may even be difficult to assess the quality of existing control structures to pinpoint on areas that need improvements or modifications, not to mention initiating control structure design for optimal operation of a modern plant with highly interacting processes and considerable complexity.

The second reason concerns the challenges faced by engineers when they rely only on experience to derive control structures. Control structures that are derived from engineering intuition and judgement could yield decent performance, but they may not necessarily operate optimally. Complicated logic is needed to reconfigure single loop PID controllers for supervisory control (Skogestad, 2004a). For this reason, it may be difficult for one to conceive the idea of designing

[1] besides having a good control structure design, effective operator training and change management are also essential in ensuring that control strategy is understood and applied correctly

advanced control structures for optimal control without a good grasp of key concepts. Left to his own devices, the engineer makes a hasty attempt in control structure design without following rules or guidelines. Moreover, without the right knowledge and appreciation of rules for optimal control, the input-output of single loop controllers may be paired incorrectly to begin with, manifesting non-optimal performance issue already in the control structure design of the regulatory layer.

In view of the presented challenges and drawbacks of poor control structure design, the standardization of design approaches is highly recommended, underpinning the usefulness of having a strategy. Given the human weakness in failing to notice knowledge gaps, or simply put, "you don't know what you don't know", a strategy serves to bridge this gap with a step-by-step approach.

## 1.1 Motivation

The moment has come for us to rethink control structure design for inventory control systems. Within the chemical process, there are complexities which the traditional control system may not be equipped to handle. Sometimes the control problem is defined only to consider the simplest or most idealistic case. The paper on "*Consistent inventory control*" published by Aske and Skogestad (2009) revealed *locally consistent* control design structures for cases where the throughput manipulator or TPM is set at the bottleneck for the network of tanks arranged in series. But the real process embodies flow disturbances that may just not be static. The moving bottleneck problem manifests itself in the ineffectiveness of traditional control structures handling disturbances that move across locations in a value chain, especially in flow networks involving storage tanks (Lindholm and Johnsson (2012), Aske et al. (2007)). With the complexity involved, the challenge will be to design a control structure that can switch and operate across operation modes. This thesis preaches the use of a strategy[2] that can simplify the control structure search with the aid of proper design procedures and design heuristics.

In the work of Minasidis et al. (2015) on economic plantwide control, the strategy culminates in a set of rules for control structure design to achieve economic optimal control of the entire plant or its specific parts, and adopts the hierarchical time-scale decomposition of the control structure for the controller algorithm design. Though this work formulated the rules for economic plantwide control design using supervisory controllers like ACS or MPC, specific rules and design procedures for ACS have not yet been developed.

This thesis weaves together the relevant design procedures from very recent sources which include the conference papers of Reyes-Lúa et al. (2018b) and Reyes-Lúa and Zotică (2019), as

[2] bottom-up strategy beginning with the regulatory control layer

well as selected sections of established procedures laid down by Skogestad (2004a) on economic plantwide control, to compile and elaborate[3] on these design procedures. The strategy is to use ACS and SRC design procedures to develop a split-range controller capable of control strategy selection for optimal operation. According to Aske et al. (2007), achieving optimal operation of a plant is by operating it at maximum throughput. This set of design procedures is applied to the moving bottleneck in a flow network.

[3]provide contextual explanation and detailed description for each step in the design procedures

## 1.2 Thesis structure

The outline of the thesis is as follows.

**Chapter 2** provides an overview of classical advanced control structures and the control hierarchy found in process plants. The *design procedures for optimal control* is also included so that the reader will be able to discern whether an optimal control problem is under or over specified and adopt the right approach to handle each case. The topic on *input saturation pairing rule* is particularly important as it emphasizes on the need for optimal input-output pairings before considering advanced controller features. This is followed by the guidelines for ACS design to achieve optimal operation. Last but not least, the split-range controller is compared with another mid-ranging controller called the valve-positioning controller.

**Chapter 3** explains the elements of SRC theory so as to lay the foundations of developing a systematic design approach for SRC. In particular, the two *degrees-of-freedom* SRC is of interest for many SRC applications in the industry, but the design of SRC is not limited to two *degrees-of freedom*. Different variations of SRC are explored and the generalized linear equation for SRC is proposed. Lastly, the approach to modeling and simulation of the SRC in Simulink is also explained and discussed.

**Chapter 4** develops the case study of the moving bottleneck in inventory control systems and use the design procedure for classical advanced controller to develop an optimal control structure using SRC with selectors.

**Chapter 5** evaluates the SRC performance of maintaining inventory levels at the level setpoint as flow disturbances, which are introduced at various locations in the flow network, are systematically elminated by the same ACS control strucutre. Another performance criterion is that the SRC is able to position the TPM at the flow bottleneck and control the flow throughput at the active constraint.

**Chapter 6** discusses the possible applications of the SRC at the industrial site in Perstorp AB and drawbacks such SRC design brings.

**Chapter 7** delivers the conclusion to the work performed in this thesis.

# 2 — Preliminaries

This chapter lays the foundation of classical advanced controllers and relevant optimal control design concepts. In the beginning, it is apt to clarify the terms that are used extensively in this thesis. These definitions are adapted from the references of Larsson and Skogestad (2000) and Skogestad (2004a), unless otherwise stated or indicated as “in this work”.

**Table 2.1:** Term Definitions

- ***Control structure design***- the structural decisions of the control systems which include the variables to control and the input-output control pairings used in control loops
- ***Optimal control structure***- the control structure that yields optimal performance while regulating the process at or close to its operating conditions
- ***Controlled Variable (CV)***- variable that is controlled at a given reference or setpoint
- ***Manipulated Variable (MV)***- degree of freedom used by a controller to control the CV
- ***Degrees of freedom (DOF)***- inputs or control valves that are manipulated to achieve a desired control output like flow rate or liquid level. The number of control degrees of freedom equals the number of MVs. (Seborg et al., 2003)
- ***Pairing***- selection of an MV to be paired with a CV for control
- ***Classical advanced control structures (ACS)***- control structure that uses different multivariable schemes to provide the set points for regulatory controller (Seborg et al., 2003)
- ***Active constraints***- related to variables (MVs or CVs) that are kept optimally at their limiting values. The examples include maximum cooling of a compressor, which is an active MV constraint and maximum reactor pressure, which is an active CV constraint. (Reyes-Lúa et al., 2018b)
- ***Control structure or control strategy***- refers to all structural decisions that manifest themselves in the design of a control system (Skogestad and Postlethwaite, 2005)
- ***Decentralized control***- comprises of a network of standalone feedback controllers, where a subset of CVs is controlled by a subset of MVs. When the control configuration is decided on, each subset should not be used by any other controllers. (Skogestad and Postlethwaite, 2005)
- ***Strategy***- in this work, it means a concerted effort to use recommended design procedures and/or heuristics to design control structure for optimal operation.
- ***Rule***- in this work, it means a practical guideline

Single-input and single-output (SISO) control structures account only for one MV to one CV pairing relationships and disregard how one MV may affect another CV. However, many control problems are inherently multivariable as they contend with two or more interacting controlled variables. For example, the inlet flow rate to a tank affects both the liquid holdup in a tank (or multiple tanks) and flow throughput. This prompts for the need to search for suitable multiple-input and multiple-output (MIMO) control structures (or strategies) that are capable of satisfying multiple control objectives or active constraints at the same time. (Viknesh et al. (2004); Seborg et al. (2003))

When the SISO or decentralized control approach is taken, each controller is designed to meet the control objective for that given control pairing. SISO controllers designed in isolation of each other may perform optimally with the chosen setpoints if the process only operates within an active constraint region. Single loop PID feedback controllers[1] cannot optimally handle the change in active constraint region without a top-down controller action to manage the order of priority of handling constraints. The control hierarchy within the control layer, as outlined in Figure 2.1, comprises of a supervisory control layer that is responsible for updating the setpoint for the regulatory control layer when active constraint changes.

[1] with the exception of multiple single loop controllers operating at different setpoints and controlling the same CV

## 2.1 Overview of advanced controllers

When there are multiple control objectives to be satisfied such as meeting the product quality and maximizing production throughput, a good multivariable centralized control strategy accounts for both dynamics and interactions in the process model by considering all non-trivial MV-CV pairing relationships. In addition to having models with good representation of the process, a priority list of constraints also forms an integral part of a well-functioning MIMO control strategy. There is no short of literature that discusses on MIMO strategies for process control, and some eclectic works on MIMO control strategies, particularly on centralized model-predictive control (MPC), are hereby provided for the reader's reference, in books Camacho and Bordons (1999), and in paper Skogestad (2004a) and Qin and Badgwell (2003).

As advanced process control (APC), which is the commonly used name in industry for multivariable centralized controllers, are expensive, they are deployed exclusively in the largest and most well-understood processes that are operating at tight profit margins. Though MIMO control strategies could easily outperform single loop controllers, the decision to deploy multivariable controllers is only tenable when the controller performance gain offsets the cost of building and maintaining these models and infrastructure. This also explains why smaller, more complex and unstructured processes are not likely to receive technical expertise and budget resources to commission APC, as the cost outweighs the benefits. (Feldmann et al. (2017), Forsman (2016))

For plants and processes that are not targeted for APC implementation, a simpler control structure approach could be employed to improve control performance over standalone single-loop PID feedback controllers. The opportunity arises by leveraging on the existing plant distributed control system[2] (DCS) to implement control structures capable of optimal control. These control structures, known as classical advanced control structures (ACS), use a combination of standard PID controllers, logic blocks and selectors to form a subset of advanced controllers that is distinct from their model-based counterparts.

The key operating philosophy of ACS in achieving optimal operation lies with its ability to manage moving active constraints when input saturates. Beginning with good input-output pairings for *decentralized control* obtained from the *input saturation pairing rule*, ACS can switch the input-output pairings when the situation warrants the need.

[2] *distributed* means that computational processing takes place across multiple nodes within the local DCS network

## 2.2 Hierarchical control systems

As mentioned previously, optimal control is carried out with the control system hierarchy. The higher layers of the control hierarchy relate to production scheduling and optimization, both of which are reserved for the domain of supply chain decision making so they are not covered in the scope of control structure design. The focus is then on the control layer outlined in dashed lines. The control layer in this hierarchy can be decomposed into the supervisory and regulatory layers where they operate at different time scales (Figure 2.1). The upper layer updates the setpoints of selected control outputs at periodic intervals (hourly) while the feedback controller in the lower layer regulates the process close to or at this optimal setpoint continuously (at a faster time scale) (Skogestad and Postlethwaite, 2005). In essence, the control hierarchy segregates the optimizer and controller function on the basis of time-scale difference in the decision making process.

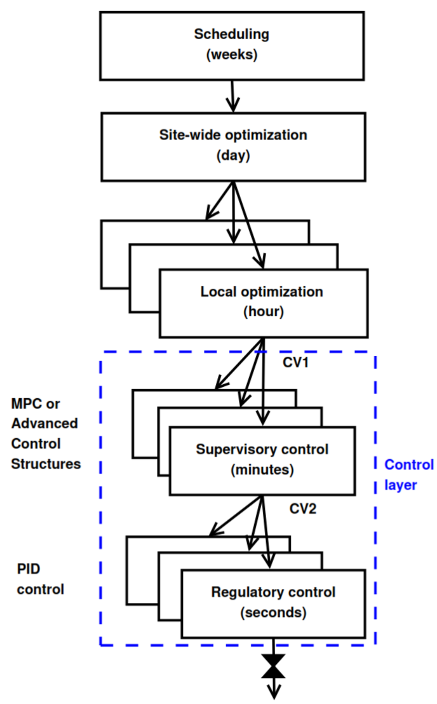



**Figure 2.1:** Typical control hierarchy in a process plant (Skogestad, 2004a)

Classical advanced control structures (ACS) are supervisory controllers that are widely used by the industry for many years. They are implemented on the plant's distributed control system (DCS) to augment the capabilities of standard single-loop PID control structures by incorporating multivariable control schemes. In contrast with other model-based multivariable control strategies, ACS do not require a process model and an optimization problem formulation.

With a good design basis, they serve as a supervisory or advanced controllers handling moving active constraints to achieve optimal operation. They are also used to extend the operating range of a regulatory controller by having multiple MVs to control a CV; this is the original purpose of the split-range controller. The advantage of having a hierarchy for the control layer is that standard PID feedback control resumes operating in the regulatory control layer even after advanced controllers are implemented (Seborg et al., 2003). As PID feedback schemes make up more than 90% of all control loops in the industry according to Åström and Hägglund (2001), this prevalence of PID controllers harbours untapped opportunities for existing single loop control structures to be upgraded to ACS.

## 2.3 Design procedures for optimal control

Skogestad (2004a), in his *economic plantwide control* paper, suggests guidelines to design the control layer[3] outlined in Figure 2.1. One of the key procedures relates to the selection of variables for *optimal economic operation.*

**Table 2.2:** Steps to select variables for optimal control (Skogestad, 2004a)

| | |
|---|---|
| **1. Identify the variables for control structure design** | |
| $n_u$ | number of degrees of freedom or inputs |
| $n_y$ | number of outputs |
| **2. Identify the variables with no steady state effect on the process but otherwise need to be controlled to prevent drifting** | |
| $n_{0u}$ | number of degrees of freedom or inputs with *no steady state effect* |
| $n_{0y}$ | number of outputs with *no steady state effect* |
| **3. Select the variables designated for optimal control** | |
| $n_a$ | number of active constraints (could be MV and/or CV active constraint) |
| $n_{\text{u, opt}}$ | number of degrees of freedom that affect the optimization cost function $J$, leaving out the DOFs which have no steady state effect on the process |
| $n_{\text{uc}}$ | number of unconstrained degrees of freedom, given by $n_{\text{u, opt}} - n_a$. These are also remaining degrees of freedom which are maintained at constant setpoints which give rise to only a small economic loss when disturbances occur |

In the ideal scenario, the number of inputs is equal to the number of active constraints, but this is not always the case. To cover all scenarios in optimal control problems, four cases are identified to provide guidance on how to control the process using active constraints and/or self-optimizing control to achieve close to optimal or optimal operations. Self-optimizing control is, by definition, the controlling of an unconstrained variable at a constant setpoint so as to operate at an acceptable loss from the optimal. For this thesis, self-optimizing control is not covered.

[3]both regulatory and supervisory control layers

**Table 2.3:** Cases in optimal control problems

| **Cases** | |
|---|---|
| **1** | $n_a = n_{\text{u, opt}} \Rightarrow n_{\text{uc}} = 0$<br>Control all the active constraints to achieve optimal operation |
| **2** | $n_a = 0 \Rightarrow n_{\text{u, opt}} = n_{\text{uc}}$<br>When there are no active constraints, apply self-optimizing control on all the unconstrained variables |
| **3** | $0 < n_a < n_{\text{u, opt}} \Rightarrow n_{\text{uc}} = n_{\text{u, opt}} - n_a$<br>Control all the active constraints and for the remaining $n_{\text{uc}}$, control them by using self-optimizing control. |
| **4** | $n_a > n_{\text{u, opt}} \Rightarrow$ not possible to control all the active constraints $n_a$<br>Some low priority active constraints have to be given up for the limited number of inputs to be paired with the remaining high priority constraints (MV or CV active constraints). To this end, soft constraints are used on the less important CVs. However, it is not feasible to operate the process at the true optimal point when there are fewer inputs than CVs, as some active constraints have to be given up resulting in the process drifting from the optimal operating point. |

In summary, these cases identify all possible scenarios for optimal control design. After meticulously classifying these variables, one can systematically perform the procedures for optimal control.

1. *Control all the active constraints.*

   According to definition, controlling all the active constraints allows the process to operate at the optimum, minimizing the *economic loss* $J$.

2. *Control sensitive and drifting variables in the regulatory control layer.*

   These variables do not have a steady state on the process, but it is necessary to keep these variables within the bounds of the control limits.

3. *Explore the possibility of applying the self-optimizing control concept on all the remaining unconstrained degrees of freedom.*

   According to the definition of unconstrained degrees of freedom in Table 2.2.

Other important considerations in control structure design are also hereby listed, for completeness.

4. *Inventory control loops should be designed around the throughput manipulator and never across the TPM.*

   According to Price et al. (1994), process control designers are mostly fixated on control of process quality variables, which are manifested as sensitive and drifting variables outlined in step 2, and often overlook the control structure design of production rate and inventory control. Throughput manipulators (TPM) , as the name suggests, control the production or flow rate through the system. They can be described as *explicit* TPMs. Price et al. (1994) further labels another class of TPMs as *implicit*, as these controllers do not fall into the traditional classification of flows, but rather influences the process conditions to give rise to quantity of product produced or separation of products. The good examples are pressure controller in a flash drum and temperature controller of the reboiler of a distillation column. These controllers control the conditions that vary the separation of the product mixture that affects both the liquid and vapor flow rates.

One useful rule that provides guidance on *consistency* of inventory control systems is the *radiating rule*. The rule suggests that the direction of inventory control should radiate away from the TPM, as evidently seen from the optimal operation modes found in Figure 4.1. The definition of *consistency* of a system is to have at least one of the flows (inlet or outlet) be controlled by its mass. If the TPM and inventory control loop crosses, then inventory control for the system will not be self-consistent.

5. *To avoid a long loop and resulting back-off, the TPM should be located close to the bottleneck.*

   Back-off means the deviation between the process value and the active constraint or optimal value; usually a back-off is applied as a margin from the active constraint to prevent the process from operating in the non-feasible region when disturbance occurs (Aske et al., 2007). The farther the TPM is located away from the bottleneck, the more allowance has to provided for margin of error (due to increased variability). For example, if the control valve is located very far downstream of the disturbance, then the flow setpoint has to be set very conservatively from the active constraint as time delay causes the process value to oscillate more, resulting in increased frequency of non-feasible operations. The effect of a long loop on process control is relatively straightforward. The longer the control loop, the longer is the time delay and that is not good/ideal for control. One good example is shown in Figure 6.2a, where there is a long flow control loop. Despite the issues of inconsistent inventory control, there are also problems with such control structures, particularly when the buffer tanks have very long *residence time* $\tau$. To conclude, it is prudent to always locate all potential bottlenecks of the system and place the TPM close to them.

In essence, optimal control design procedures encompass all steps relating not just to control the process at its optimal process values (or active constraints), but also to satisfy the product quality and ensure that the inventory control structure design is *consistent*.

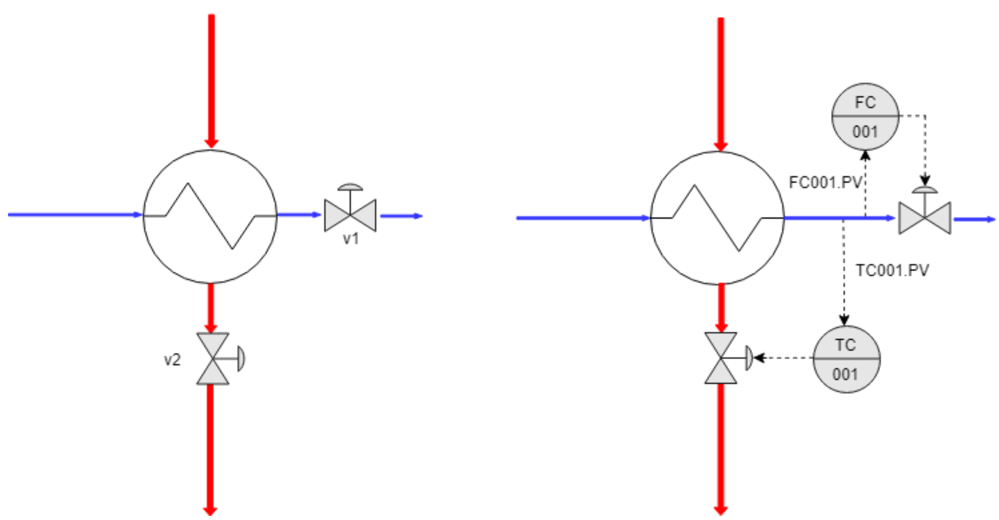
## 2.4 Input Saturation Pairing Rule

Input saturation occurs when a control valve is either fully open or closed or when a compressor/pump is operating at its full capacity or idling in standby mode. When the input *optimally saturates*[4], it operates at its physical constraint to satisfy the optimality conditions.

According to the input saturation pairing rule (ISR) for decentralized control, when more than one input have an effect on a CV, a *more important* CV should be paired with the input that is *not* likely to saturate (Minasidis et al., 2015). This rule sets aside the *less important* CVs to pair up with inputs that are more likely (optimally) saturate, given that one can afford to give up these CVs.

Intuitively, we would not want to pair our more important CV with an input that optimally saturates as this active constraint will be given up when disturbance is introduced, resulting in loss from the *economic objective*. The reason is that each input or control valve is bounded by its physical limits of not being able to actuate more than a fully open state (which leads to flow rate greater than the maximum) and actuate less than a fully closed state (which leads to negative flow rate - infeasible operation).

### 2.4.1 Application - Heat exchanger control problem




**(a)** Two degrees of freedom **(b)** Input-output pairing for optimal operation

**Figure 2.2:** Input-output pairing for heat exchanger control using the input saturation rule

[4]means that we want to keep these inputs at saturation to achieve optimal operation. One can call the control valve output value, for example `FC001.OP` tag in the plant's data historian to obtain this insight. A tag that sustains fully open (100%) or fully closed (0%) for extended periods during operation may indicate that the control valve might be operating at its active constraint.

The input saturation pairing rule is applied to the heat exchanger control example (Figure 2.2a) inspired by Reyes-Lúa et al. (2018b). The purpose of the heat exchanger unit is to provide sufficient heating for the outlet feed stream to leave at the desired temperature. This process understanding provides information needed to form a control objective for optimal operation, i.e. to maintain the feed stream at the maximum possible flow rate while controlling the temperature at the setpoint. As feed temperature is the more important CV that we control to prevent drifting, it should be paired with the input that does not optimally saturate. $v_1$ optimally saturates given that the maximum feed throughput is synonymous with optimal operation. Figure 2.2b shows the optimal input-output pairings for the heat exchanger system, which is summarised below.

1. The output of feed temperature (CV) should be paired with control valve $v_2$
2. The output of feed flow rate, being the less important CV, should be paired with valve that optimally saturates, which is $v_1$

It is possible that the input-output pairing obtained by the input saturation rule may become non-optimal when the input paired with a more important CV saturates due to disturbance. For instance, when there is a precipitous fall in heating fluid temperature, $v_2$ opens fully to compensate for the loss of heating duty until it reaches saturation. As the decrease in temperature persists, the existing control structure would eventually fail to honour the more important temperature constraint if feed rate is not cut back. $v_2$ becomes likely to saturate and the $v_1$ pairing with flow rate should be given up to control temperature at the setpoint.

In the practical case, operations would add a feed temperature alarm, without consultation from the process control engineer, so that the shift operator could intervene to close $v_1$ when the temperature constraint is not met. Such interventions imply that the operator takes on the role of the supervisory control layer. To reduce the number of alarms, operator touches and non-optimal decision making, control structure improvements may be made to reduce the number of manual interventions and improve overall efficiency of the plant.

*How should we design a control strategy for optimal operation?*

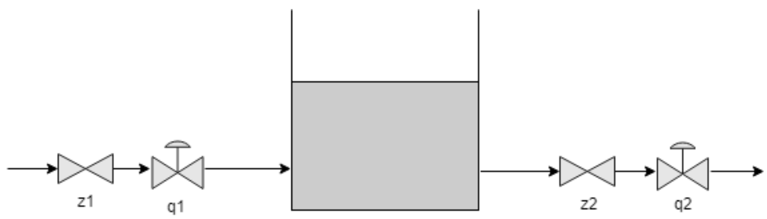
The ideal control strategy should be able to handle changes in active constraints, so that when these events do occur, the more important CV constraint in the priority list is still honoured at the expense of giving up the less important CV.

### 2.4.2 Application - Inventory level and flow control problem

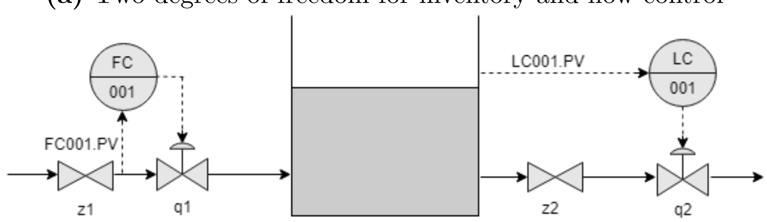
Similarly, the input saturation rule may be applied to the tank inventory control problem. The control objective for optimal operation is to maximize flow throughput while holding the inventory level in the tank constant.

The control valve closest to the process bottleneck should be assigned the TPM to maximize flow throughput, suggesting that it optimally saturates. When $z_1$ closes, the inlet flow rate is reduced. According to input saturation rule, the less important CV, flow rate, should then be paired with the valve ($q_1$) as it optimally saturates. The remaining *degree of freedom* pairs with the more important CV. When $z_2$ closes, the bottleneck occurs at the outlet and the flow rate should then be paired with $q_2$.

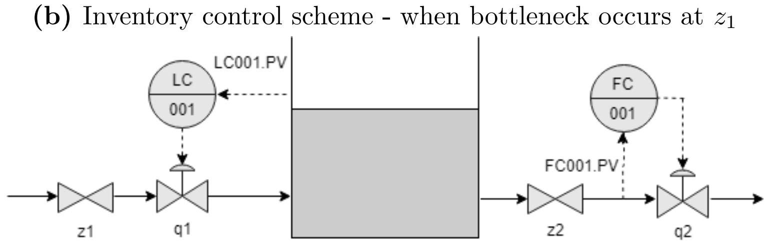
Figure 2.3b and Figure 2.3c summarize the control structures for optimal operation given the bottlenecks occurring at $z_1$ and $z_2$ respectively. The complication arises when the disturbance switches between $z_1$ and $z_2$. Assuming that we had the ISR pairings from Figure 2.3b case, a new disturbance is introduced at $z_2$ such that $q_2$ saturates in order the hold the level setpoint. As $z_2$ closes further, the level controller can no longer hold that setpoint without cutting back on $q_1$. Ideally, the control pairings should already have switched to those shown in Figure 2.3c.




**(a)** Two degrees of freedom for inventory and flow control




**(b)** Inventory control scheme - when bottleneck occurs at $z_1$




**(c)** Inventory control scheme - when bottleneck occurs at $z_2$

**Figure 2.3:** Input-output pairing for tank inventory control using the input saturation rule

In practice, operations may request the control engineer to create a soft button key in the DCS display panel, allowing the operators to switch between two controller modes to counteract the disturbance. From an implementation point of view, the control engineer can create a switch to operate between the two modes found for Figure 2.3b and 2.3c with mode selectors.

However, the operator has to manually invoke the switch and if the moving disturbance occurs too frequently, he will be preoccupied with button pressing. If the operator does not keep up with the switching, the tank level will swing. The system will consequently not operate at the active constraint if the operator does not keep up with switching of the operation modes.

*How should we design a control strategy for optimal operation?*

The ideal control strategy should be able to switch input-output pairings when the input which is originally paired with a more important tank level CV saturates (see Figure 2.3b and Figure 2.3c). When the pairing switch occurs, the input that saturates due to disturbance is now paired with the less important flow rate CV while the input that optimally saturates before is used to control the more important drifting tank level CV.

Table 2.4 conceives the scenarios with ballpark flow rates to show how the input-output pairings change with varying nominal flow rates at the inlet and outlet of the tank.

**Table 2.4:** Control structure selection for optimal operation; pair the input that optimally saturates with the less important variable (flow rate) that can be given up

| Scenario | Inlet flow rate | Outlet flow rate | Control pairing |
|---|---|---|---|
| A | 300 | 250 | Use control scheme in Figure 2.3c |
| B | 240 | 250 | Use control scheme in Figure 2.3b |
| C | 250 | 250 | Use control scheme in either Figure 2.3b or 2.3c |

### 2.4.3 Limitations of decentralized control

When single-loop PID feedback controllers are used for control, each input MV is paired with a CV. For processes with interacting variables, two or more inputs can have an effect on one CV or one input can have an effect on two or more CVs. Disturbances may cause one input that does not optimally saturate to saturate. Table 2.5 explains the role of ACS in reassigning MV-CV pairings when disturbance causes a change in active constraint.

**Table 2.5:** Procedure to manage MV-CV pairing when ISR no longer holds

**(a)** Originally when it is possible to apply the input saturation pairing rule

| **Pairing based on Input Saturation Pairing Rule** | | |
|---|---|---|
| MV does not optimally saturate | $\overset{\text{to pair with}}{\Rightarrow}$ | $CV_{\text{more important}}$ |
| MV that optimally saturates | $\overset{\text{to pair with}}{\Rightarrow}$ | $CV_{\text{less important}}$ |

**(b)** When the input previously paired with the more important CV saturates due to disturbance, the input saturation rule no longer holds. The input that optimally saturates should be given up to pair with the more important CV

| **When disturbance is introduced** | | |
|---|---|---|
| MV does not optimally saturate | $\overset{\text{becomes}}{\Rightarrow}$ | MV saturates due to disturbance |
| MV that optimally saturate | $\overset{\text{becomes}}{\Rightarrow}$ | MV should now be paired up with more important CV |
| **Resultant pairing due to disturbance** | | |
| MV saturates due disturbance | $\overset{\text{paired with}}{\Rightarrow}$ | $CV_{\text{more important}}$ |
| MV should be paired up with more important CV | $\overset{\text{paired with}}{\Rightarrow}$ | $CV_{\text{less important}}$ |

**(c)** Reassign another input that does not optimally saturate, but previously paired with a less important CV, to the more important CV

| **Reassigned pairing using advanced control structures** | | |
|---|---|---|
| MV should be paired up with the more important CV | $\overset{\text{reassigned to}}{\Rightarrow}$ | $CV_{\text{more important}}$ |
| MV saturates due to disturbance | $\overset{\text{reassigned to}}{\Rightarrow}$ | $CV_{\text{less important}}$ |

## 2.5 General Procedure for ACS design

The limitations of single-loop controllers or decentralized control in control structure design for optimal operation, particularly when an input saturates, were discussed previously. One exception to the limitations is when multiple single loop controllers are operating at different setpoints but controlling the same CV. This control strategy is not discussed extensively as the focus is on implementing ACS, and particularly the SRC.

The supervisory control layer works around these limitations by updating the setpoint of the regulatory controller when the active constraint changes. One class of supervisory controller capable of performing this function is the ACS, which is commonly implemented within the DCS.

Reyes-Lúa et al. (2018b) originally proposed the design procedures for ACS. The approach is hereby expounded.

1. *Define the control objectives based on the process understanding. Identify all the active constraints and variables related to optimal control.*

   The purpose is known and the control objectives are explicitly defined from process understanding. This is an important step which is often overlooked or taken lightly, particularly when operators give ambiguous problem descriptions or unclear expectations[a] (Forsman, 2016). Section 2.3 provided the approach to identify the variables related to optimal control, which includes active constraints.

2. *Develop the priority list of constraints in order of importance.*

   A list of constraints is prepared and the priority of constraints is created. As a recommendation, the ranking of constraints may follow this sequence from highest to lowest priority.

   – Constraints relating to **input saturation** hold the highest priority spot as these constraints set the feasible bounds of operation. These constraints, which as typically MV inequality constraints, are hardly breached as normal operation is carried out within those bounds.

   – **Sensitive or drifting variables** hold second spot as these variables are preferentially fulfilled. They are the conditions necessary to operate the process in a stable, feasible and efficient manner. They also typically do not have a steady state effect on the process. These constraints are usually CV equality constraints.

- **Active constraints** are operated at their limiting values to meet the economic objectives or conditions for optimal operation. They can possibly be given up. These constraints are usually MV or CV equality constraints. An example of a MV equality constraint is when a target flow rate (TPM) is set at 80% of the control valve opening.

- **Self-optimizing variables** can be given up. They are usually at the bottom of the priority list as they are usually *unconstrained variables* that do not have a limiting operating value. Consequently, these variables do not have a direct impact on *economic cost*.

3. *If multiple ACS are needed to satisfy the control objectives, choose a subset or group of active constraints from the priority list and commence the design of the ACS one at a time*

   Unlike MPC which adopts the centralized approach in handling multiple inputs and outputs, the ACS approach may, from the plantwide perspective, be perceived as decentralized given that each control objective is handled separately by each ACS. Recall that a SRC without selectors is a multiple inputs and single output (MISO) control strategy that is upgraded to MIMO when used together with selectors. Furthermore, using multiple ACS is needed for a system with multiple tanks where several control objectives are not related to each other, like maintaining the tank levels in multiple tanks at a constant setpoint. For such cases, every constraint is grouped by the ACS they belong to, so that each ACS performs independently of the other.

4. *Commence on control structure design with most, if not all, of the constraints met, including active constraints. This usually done by considering the process operating at its nominal point.*

   Refer to Section 3.5 for the SRC (the type of ACS covered in this thesis) design procedures.

   Considering that most of the constraints should be satisfied is a prerequisite for optimal control structure design. If most constraints are not satisfied, then the optimal control objective might not be met, especially when one of the requirements is to control all active constraints. The exception is self-optimizing variables as they are optional for optimal control.

5. *Identify all disturbances that could result in new constraints becoming active.*

   The plot of the active constraint regions provides a visualization of how disturbance affects the active constraint(s), which relate(s) to optimal operation. This step is helpful for understanding how active constraint changes in the presence of disturbance.

6. *When a MV constraint is reached, investigate into the possibility that an MV which is paired with a more important CV would reach saturation*

   Assuming that we already performed the input-output pairing using the *input saturation pairing rule*, then we investigate if disturbances would cause our inputs to saturate.

   – When an input that is paired with a *less important* CV saturates, no action is required as one can afford to give up this CV.

   – When an input is paired with a *more important* CV saturates, then more important CV must be reassigned to an input paired with a less important CV, at the expense of giving up the less important CV.

7. *When a CV constraint is reached, give up the less important CV constraint*

   If there were multiple CV constraints, we want to honour the CV constraints that are more important. `min` or `max` selectors allow the input to choose between the incoming signals in order to honour the CV constraint in the order of priority. This conditional selection of signals means that,

   – If both the **active constraint** and **more important constraint** are met, use the arbitrary input or control valve to control the process at the active constraint to achieve optimal operation.

   – If **more important CV constraints are not met**, give up the active constraints and use that same input to control the process with the more important constraint.

[a] to prevent misalignment of expectations, a management of change meeting is carried out by all stakeholders of the respective departments for major control structure design modifications or changes

# 3 — Elements of Split-Range Control Theory

## 3.1 Two degrees-of-freedom split-range controller

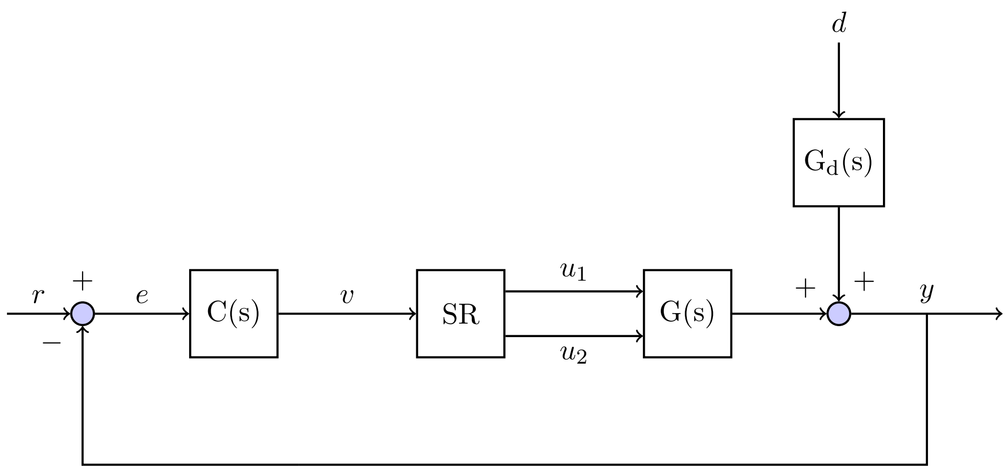



**Figure 3.1:** Block diagram of two degrees-of-freedom split-range feedback control system

Figure 3.1 shows the block diagram for the two degrees-of-freedom split-range controller that rejects the disturbance $d$ introduced to the system. The input to the controller C(s), which is a normal PID controller, is the *offset from the reference* $r - y$, also known as *controller error* $e$. The controller C(s) computes an internal signal $v$ which is resolved by the split-range (SR) block[1] into signals ($u_1$ and $u_2$) that determine the valve openings. In theory, the SRC can manage more than two degrees of freedom $u_i$ for a given controlled output $y$.

The two degrees-of-freedom split-range controller is the most basic and common implementation of the SRC. It exemplifies the SRC concepts and is used in a variety of control applications. SRC extends the operating range by having more inputs to control one CV or manages moving active constraints by manipulating another input when one input saturates.

[1] may be called a splitter block in the DCS. **Source:** *Emerson Delta-V* DCS

## 3.2 Split-range control preliminaries

This theory of SRC is inspired by the paper submitted for the 12$^{\text{th}}$ IFAC Symposium on Dynamics and Control of Process Systems to take place in Florianópolis, Brazil, April 2019 (Reyes-Lúa and Zotică, 2019).

### 3.2.1 SRC parameters

The internal signal $v(t)$ of the SRC is a deviation variable as it is calculated to be $K_c e(t)$ where $K_c$ and $e(t)$ are the proportional gain and the error of the measured output $y$ from the reference signal at a given time respectively.

The proportional only feedback controller has its *controller output* expressed in deviation variables $\Delta u$ (see eq. (3.1)). Hence, the internal signal $v$ from the analogous proportional only controller of the SRC is also expressed in terms of deviation variables (see eq. (3.2)).

$$\Delta u = u(t) - u_0 = K_c e(t) \tag{3.1}$$

$$v(t) = K_c e(t) \tag{3.2}$$

Input saturation occurs at two states, (i) maximum valve opening ($u_{i,\max}$ fully open) and (ii) minimum valve opening ($u_{i,\min}$ fully closed). When one of the inputs saturates in the internal signal $v$ range, the SRC will use another input to control the CV. This transition takes place when the internal signal $v$ is at the split value or $v^*$. Another SRC parameter, $\alpha_i$, which is also known as the slope of the split-range block for each input, determines how fast an input reacts to the internal signal $v$. The SRC action on an arbitrary input $\mathrm{MV}_i$ is said to be *direct-acting* when $\alpha_i > 0$ or *reverse-acting* when $\alpha_i < 0$.

Finally, the split-range SR block outputs, $u_1$ and $u_2$, are actual MV opening signals with biases already added (see eq. (3.21)).

### 3.2.2 Clarifications on split-range controllers

Most literature, notably (Bequette, 2002, Section 12.4) and other books on SRC, refer $v$ as the *controller output*. This can be misleading as *controller output* is supposed to be identified with the *degrees of freedom* manipulated by the SRC. As the SRC is constructed using two DCS elements, namely the PID control block and SR logic block, practitioners may be inclined to use the term *controller output* to refer to the PID controller output signal rather than the actual SRC outputs. As both elements collectively make up the SRC function, we advocate the term *internal signal from the controller* ($v$) instead of *controller output*. Therefore, we would now say that a two degrees-of-freedom SRC has two controller outputs, $u_1$ and $u_2$, both of which correspond to 0-100% control valve openings.

The split value[2] $v^*$ is quite often, conveniently set at 50%[3] (Bequette, 2002, Figure 12.5). This practice limits how $\alpha_i$ may vary for each control valve. The operating principle of the SRC is similar to the PID controller, i.e. the SRC parameters should also be inferred from process parameters like $K_p$ and $\tau$, and not pre-determined by $v^*$. Other conventions like fixing $v_{\min}$ and $v_{\max}$ for the internal signal $v$ range could also limit the SRC design.

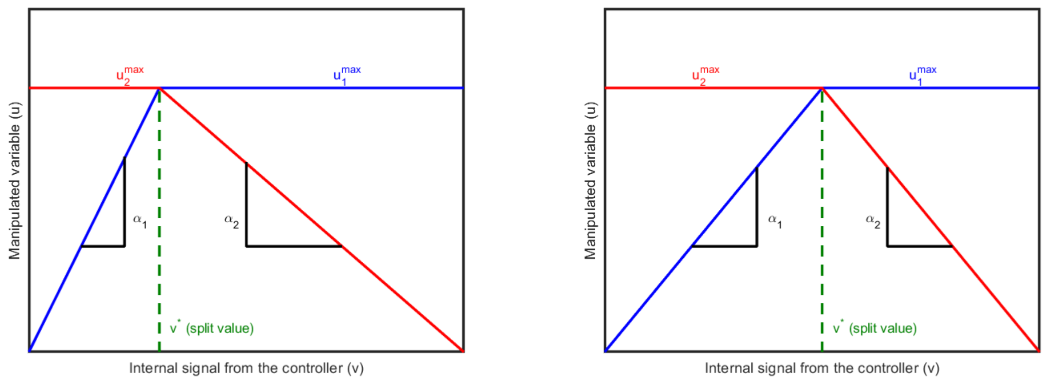
Another misconception arises from $\alpha_i$. The change in slope $\alpha_i$ modifies the closed loop response only and does **not** have an influence on the operational sequence of the MVs within the internal signal $v$ range. The order of MV control is determined by the logic sequence configured in the SR block and the integrator of the controller. In the Simulink simulation model, the MV order of sequence for a two degrees-of-freedom SRC is decided by the arrangement of each input is active with increasing $v$ and the saturation block on each branch performs the signal selection (see Section 3.6.1).

[2]most, if not all, DCS require specifying $v^*$ for the SRC implementation instead of $\alpha_i$. It is advised to calculate the slopes $\alpha_i$ and then extrapolate $v^*$ that yields these slopes.

[3]in the range of $[v_{\min}, v_{\max}]$

### 3.2.3 Types of SRC

The first type of SRC involves a control strategy where one of the inputs $u_1$ tends to saturate in the fully open position, i.e. $u_{\text{sat}} = u_{\text{max}}$.




**(a)** $|\alpha_1| > |\alpha_2|$ **(b)** $|\alpha_1| = |\alpha_2|$

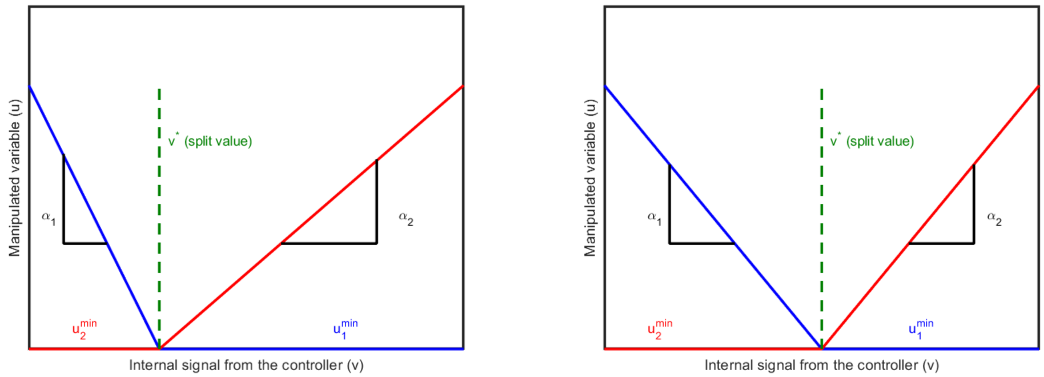
**Figure 3.2:** Split-range controller action - when MV saturation occurs at maximum valve opening

The graphical representation (Figure 3.2) of MVs that saturate fully open within the SRC operating range can be written as piecewise continuous functions.

$$\left.\begin{aligned} u_1 &= u_{1,\text{min}} + \alpha_1 v \\ u_2 &= u_{2,\text{max}} \end{aligned}\right\} \text{ for } v_{\text{min}} \leq v \leq v^* \tag{3.3}$$

$$\left.\begin{aligned} u_1 &= u_{1,\text{max}} \\ u_2 &= u_{2,\text{max}} + \alpha_2 (v - v^*) \end{aligned}\right\} \text{ for } v^* < v \leq v_{\text{max}} \tag{3.4}$$

Input saturation ($u_{\text{sat}}$) can also occur when one of the control valve tend to saturate in the fully closed position, i.e. $u_{\text{sat}} = u_{\text{min}}$.




**(a)** $|\alpha_1| > |\alpha_2|$ **(b)** $|\alpha_1| = |\alpha_2|$

**Figure 3.3:** Split-range controller action - when MV saturation occurs at minimum valve opening

Likewise, another form of SRC is also possible with the MVs saturating fully closed within the SRC operating range. The MV expressions are written as follows,

$$\left.\begin{aligned} u_1 &= u_{1,\text{max}} + \alpha_1 v \\ u_2 &= u_{2,\text{min}} \end{aligned}\right\} \text{for } v_{\text{min}} \leq v \leq v^* \tag{3.5}$$

$$\left.\begin{aligned} u_1 &= u_{1,\text{min}} \\ u_2 &= u_{2,\text{min}} + \alpha_2 (v - v^*) \end{aligned}\right\} \text{for } v^* < v \leq v_{\text{max}} \tag{3.6}$$

(Bequette, 2002, Section 12.4) designed this type of temperature SRC for a batch reactor which operates in two modes, heating or cooling, over the internal signal $v$ range. The split value $v^*$ marks the transition point where cooling stream saturates fully closed and the heating stream takes over.

In summary, the combination of MV saturation (fully open or closed) and choice of slopes ($\alpha_i$) represent different control strategies and possibilities for SRC implementation to control problems.

### 3.2.4 Slopes $\alpha_i$ on the SRC inputs

The gradient $\alpha_i$, also known as the slope of a SRC input, is an important SRC parameter which defines the change in valve opening signal $u_i$ with this internal signal $v$.

$$\alpha_i = \frac{\Delta\ u_i}{\Delta\ v} \qquad \forall\ i \in \{1, \ldots, N\} \tag{3.7}$$

The slopes $\alpha_i$ of SRC inputs in differential form is,

$$\alpha_i = \frac{\mathrm{d}u_i}{\mathrm{d}v} \qquad \forall\ i \in \{1, \ldots, N\} \tag{3.8}$$

We can infer the relationship between process gain $K_{\mathrm{p},i}$ and $\alpha_i$.

$$\alpha_i \propto \frac{1}{K_{\mathrm{p},i}} \tag{3.9}$$

$$\therefore \qquad = \underbrace{\frac{\mathrm{d}u_i}{\mathrm{d}y}}_{K_{\mathrm{c},i}} \cdot \underbrace{\frac{\mathrm{d}y}{\mathrm{d}v}}_{1/K_{\mathrm{c}}} \tag{3.10}$$

Each input $u_i$ or MV will likely have a different gain effect on the CV. There is a logical connection to pair the arbitrary input with the CV by calculating $K_{c,i}$, as if a standalone SISO controller is used.

Each $\alpha_i$ is multiplied with the controller proportional gain $K_c$ to obtain the effective controller gain $K_{\mathrm{c},i}$, which manifests as the actual controller gain an input has on the process - see eq. (3.13).

$$K_{\mathrm{p},i} = \frac{\mathrm{d}y}{\mathrm{d}u_i} \tag{3.11}$$

$$K_{\mathrm{c}} = \frac{\mathrm{d}v}{\mathrm{d}y} \tag{3.12}$$

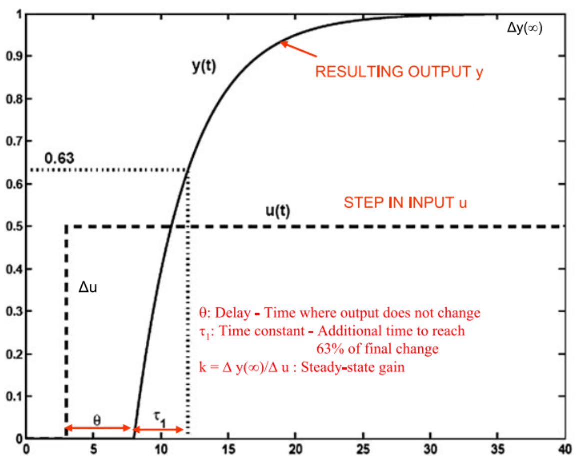
$$\frac{\mathrm{d}u_i}{\mathrm{d}y} = \frac{\mathrm{d}v}{\mathrm{d}y} \cdot \frac{\mathrm{d}u_i}{\mathrm{d}v}$$

$$\therefore K_{\mathrm{c},i} = K_{\mathrm{c}} \cdot \alpha_i \qquad \forall\ i \in \{1, \ldots, N\} \tag{3.13}$$

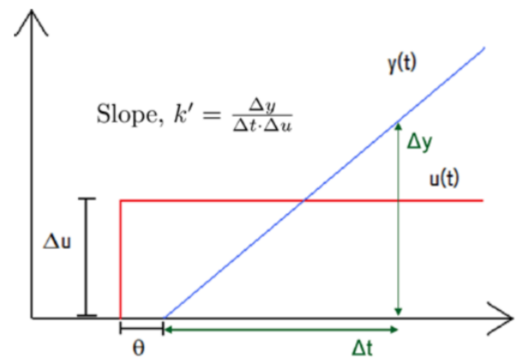
It is recommended to compute $K_{c,i}$ using SIMC tuning rules and then determine $\alpha_i$ from Equation (3.13).

### 3.2.5 Determine $K_{c,i}$ and $\tau_I$ on the SRC

The SRC controller parameter can be calculated using a systematic approach such as PID tuning rules developed by Ziegler and Nichols or Simple-Internal-Model-Control (SIMC). The details of the SIMC tuning rules can be found in Skogestad and Grimholt (2012).




(a) Open-loop step response of a first-order model




(b) Open-loop step response of an integrating model

**Figure 3.4:** Open-loop step response to determine the parameters of $K_{p,i}$, $\tau_I$ and $\theta$ (Skogestad and Grimholt, 2012)

The tuning equations are hereby listed to provide guidelines for robust controller tuning of the SRC. The nomenclature of the tuning parameters is provided below.

- $K_{p,i}$ - process gain computed for each MV
- $K_{c,i}$ - effective controller gain computed for each MV
- $\tau_i$ - process time constant for each MV
- $\tau_{I,i}$ - controller integral time for each MV. This is common for all the MVs that are linked to the SRC
- $\tau_c$ - closed loop time constant, this is a tuning parameter that is chosen for robust and tight control or smooth control
- $\theta$ - process time delay

From fig. 3.4a, the SIMC tuning parameters for each SRC MV, eq. (3.14) to (3.16), for the first-order plus time-delay process may be calculated,

$$K_{p,i} = \frac{\Delta\ y}{\Delta\ u_i} \tag{3.14}$$

$$K_{c,i} = \frac{1}{K_{p,i}} \frac{\tau_i}{(\tau_c + \theta)} \tag{3.15}$$

$$\tau_{I,i} = \min(\tau_i,\ 4(\tau_c + \theta)) \tag{3.16}$$

where $i \in \{1, \ldots, N\}$ and $N$ is the number of MVs that is related to the SRC.

Likewise from fig. 3.4b, the SIMC tuning parameters for each SRC MV, eq. (3.17) to (3.19), for the integrating plus time-delay process may be calculated,

$$K_{\mathrm{p},i} = \frac{\Delta\ y}{\Delta\ t \cdot \Delta\ u_i} \tag{3.17}$$

$$K_{\mathrm{c},i} = \frac{1}{K_{\mathrm{p},i}} \frac{1}{(\tau_c + \theta)} \tag{3.18}$$

$$\tau_{I,i} = 4(\tau_c + \theta) \tag{3.19}$$

where $i \in \{1, \ldots, N\}$ and $N$ is the number of MVs that is related to the SRC.

For both cases, $K_{\mathrm{c},i}$ is used to determine $\alpha_i$ for the SRC MV$_i$ profile and $\tau_{I,i}$ is used as the controller integral action if a PI-controller is used.

## 3.3 Mathematical formulation

The PID controller within the SRC computes an internal signal $v$ for the SR block (see SRC block diagram). The principle of a proportional-only controller is to eliminate offset $e$ and when $e$ is zero, the controller takes no counteracting actions. This also applies to the internal signal $v$, when $e = 0$ then $v$ is equal to 0, as $v = K_c e$ for the proportional-only controller. It is helpful for the start to conceptualize a two *degrees-of-freedom* SRC with split value $v^* = 0$.

Each SRC input valve opening profile is given by $\alpha_i v$, where $\alpha_i$ is the slope of each SRC input and $i$ is the number representing the SRC input. The simplest case will be the SRC input profile found in Figure 3.3, where both inputs saturate fully closed. Some textbooks like (Bequette, 2002, Figure 12.4) indicate the fail positions of the control valve relating to SRC to imply the directional movement of the valve or the sign of $\alpha_i$. One possible explanation might be that those books were written in an era when SRC was configured with a combination of hardware and software modifications. By default, the majority of the control valves in the plant are fail-closed, as the signal to control valve corresponds to a 4-20 mA to 0-100% valve opening. The instrumentation engineer[4] can install a relay to switch the signal, where 100% valve opening now corresponds to 4 mA, to configure the valve to fail-open. Modern DCS provide more tools like logic and arithmetic block elements (like gain multiplier and biases) for the control engineer to conceive more sophisticated SRC designs. To add variety and possibilities to the SRC strategy, a saturation bias term denoted as $u_{\text{sat}}$ is added to $\alpha_i v$. Since the engineer would probably be reluctant to modify the field hardware configurations, he could opt to add the saturation bias (signal bias etc) to each SRC input signal in the DCS. The SRC input profile for control valve that tend to saturate fully open is found in Figure 3.2, where the bias $u_{\text{sat}}$ is 100%.

Things get slightly more complicated when the split value $v^*$ is not zero[5]. When $v^* \neq 0$, the internal signal $v$ is translated by an amount of $-v^*$. To understand this behaviour, we return to the case where the inputs saturate fully closed. We know that $u_i = 0$ corresponds to $v = 0$ in the base case, but we now need $u_i = 0$ when $v = v^*$. This is possible if the incoming signal $v$ is transformed to $v - v^*$ before entering the split-range logic block. To this end, $-v^*$ bias is added directly to the internal signal $v$.

This conceptual writeup forms the basis of the simulation model which we will cover in Section 3.6. Also, the generalized mathematical expression for the SRC inputs may be

[4] the control engineer may also assume this responsibility in a medium sized plant

[5] Typically the PID controller in the DCS does not have an output in deviation variables, but rather a controller output that corresponds to a 0-100% reading or 4-20 mA signal. Therefore, it may seem strange to a practitioner to have negative $v$ readings, but that is fine as shown in eq. (3.2), the PID output in Simulink should be interpreted as deviation variables.

formulated.

### 3.3.1 Generalized MV equation for the SRC

In the beginning, we have the MV equation for SRC in the form written in Equation (3.20).

$$u_i = u_{\text{sat}} + \alpha_i(v - v^*) \qquad \forall\ i \in \{1, \dots, N\} \tag{3.20}$$

where

| | |
|---|---|
| $u_{\text{sat}}$ | - saturation bias term (typically 0 or 100%) |
| $v$ | - signal is calculated by the internal PID controller, C(s), within the SRC |
| $\alpha_i$ | - controller output gain for the $i^{\text{th}}$ MV |
| $N$ | - number of inputs that are used by the SRC |
| $v^*$ | - split value |

Equation (3.21) is the generalized MV equation for SRC, where $v^* = 0$. The bias term $u_{i,0}$ indicates valve opening when $v$ is equal to zero or at the origin.

$$u_i(t) = u_{i,0} + \alpha_i v \qquad \forall\ i \in \{1, \dots, N\} \tag{3.21}$$

## 3.4 Split-range controller applications

SRC may be used to expand the capabilities of traditional control structures or serve as advanced controllers in performing optimal control.

For process applications, they typically extend the steady-state operating range of a CV when the primary input saturates and the controller can no longer use the current input to control the CV at the setpoint. One classic use of SRC is temperature regulation of a building. The external temperature is 15 °C in winter and the building temperature setpoint is 20 °C. The cooling valve saturates at zero opening as the cooling water that is supplied at 10 °C will not be able to warm up the building. Hot water is supplied to the radiator via the heating valve to regulate the interior temperature at its setpoint. When it is summer, the outside temperature is 30 °C but the interior temperature setpoint still remains the same. This time, the cooling water valve opens while the heating valve saturates fully closed. The building temperature cannot be regulated at 20 °C if only the heating valve exists for this setup. Therefore, the operating range of the temperature CV is extended beyond 20 °C by adding to the SRC a cooling valve which operates when the heating valve saturates. Both the heating and cooling valves of the temperature SRC have opposite slopes $\alpha_i$, owing to the valve opening relationship with temperature. This illustration provides the basis of many SRC applications in the industry, such as temperature split-range controller of a batch reactor that runs on two campaigns, one that involves either an endothermic or exothermic reaction (Bequette, 2002).

Other novel applications of the SRC can be found in the control structure design for optimal operation, such as in the optimal energy storage of buildings (de Oliveira et al., 2016). This work on electric water heating explores ways of optimizing energy consumption in water heating. The split-range logic is used to control the hot water volume in the tank with a temperature SRC. During the day, the energy 'saving mode' is turned on and the temperature SRC controls the water temperature in the tank by manipulating the heat input $Q$ and the setpoint of the tank level controller. When the heat input valve saturates, the level setpoint falls correspondingly so that the inlet cold water feed to the tank is cut back.

## 3.5 Split-range control design procedures

The following SRC design procedures are adapted from Reyes-Lúa and Zotică (2019) and explained in greater detail.

1. *Define the saturation limits of each MV, $u_{i,min}$ and $u_{i,max}$.*

   $[u_{i,\min},\, u_{i,\max}]$ is usually [0%, 100%] for a control valve.

2. *Decide on the sequence of operating the MV over the range $[v_{min},\, v_{max}]$. Develop this graphical representation like one of the plots found in Figure 3.2.*

   See Section 3.2.3.

3. *Compute the controller tunings for each MV, taking note of the controller gains for each MV $K_{c,i}$. If PI controller is considered, then a common integral time $\tau_I$ is used on the PID controller within the SRC.*

   See Section 3.2.5.

4. *From $K_{c,i}$, calculate the slopes $\alpha_i$ of each MV.*

   See Section 3.2.4.

5. *Set the range for the signal to the SR block $[v_{min},\, v_{max}]$, where $v_{max}$ and $v_{min}$ are the upper and lower bounds of the internal signal $v$.*

   For instance, in our simple tank case study, $\alpha_i$ were found to be +1 and -1 for the two *degrees-of-freedom* SRC (See Table 4.4).

   Since most PID controllers are configured to output 0-100% in the DCS, most $[v_{\min},\, v_{\max}]$ is [0%, 100%]. However, this configuration limits the combination of $\alpha_i$ we can select for our SRC MVs. Given that $|\alpha_1| = |\alpha_2|$, the only split value $v^*$ that can achieve this is at 50%, but that will correspond to $|\alpha_i| = 2$. To achieve the desired $|\alpha_i|$, we have to extend the $[v_{\min},\, v_{\max}]$ range from 0-100% to 0-200% and define the split value $v^*$ to be 100%. This can be achieved by using a multiplier ($\times 2$) on the internal signal $v$. Note that this is done differently in the Simulink simulations and the concept is explained in Section 3.3 and steps are detailed in Section 3.6.

   The general expression for determining the $v$ range is given by the formula provided in Equation (3.22).

$$v_{\max} - v_{\min} = \sum_{i=1}^{N} \frac{u_{i,\max} - u_{i,\min}}{\alpha_i} \qquad \forall\ i \in \{1, \ldots, N\} \tag{3.22}$$

6. *Extrapolate the split value(s) $v^*$ from the computed $\alpha_i$. There are more than one $v^*$ if there are more than two degrees-of-freedom for the SRC.*

   When all the parameters are defined, then the last step is to determine the split value $v^*$. See Section 3.3.1 for the relationship of $v^*$ with the other parameters.

## 3.6 Modeling and simulation of SRC in Simulink

This section entails the steps to develop the split range logic in Simulink in order to mimic the SR block in the actual application. Simulink performs dynamic simulations and its workspace is activated from within the MATLAB environment.

The SRC block diagram in Figure 3.1 shows the SR logic block being placed between the controller C(s) and process transfer function G(s) blocks. This SR block receives the normal PID controller signal and resolves it into the input $u_i$ signals. There are three important parameters that are considered in the SRC implementation. They are $v^*$, $\alpha_i$ and $u_{i,\text{sat}}$ from Equation (3.20).

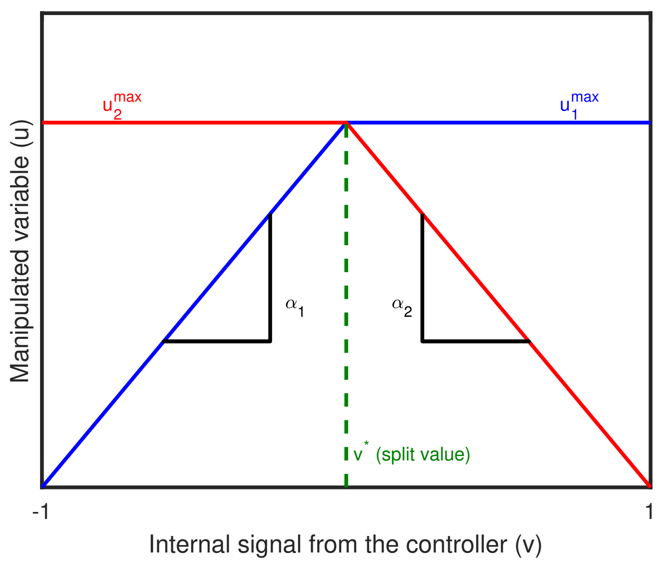



**Figure 3.5:** Level split-range-controller MV profiles for the buffer tank

We consider the two *degrees-of-freedom* SRC manipulating two inputs, $MV_1$ and $MV_2$, both of which saturate fully open in the internal signal range $v$. This symmetrical MV profile, where $|\alpha_1| = |\alpha_2| = 1$, for the SRC will be used later in the buffer tank case study given that the inlet and outlet flow specifications of the tank are similar for simplification.

The walkthrough of the simulation model creation procedure may be performed with reference to the screenshot of the Simulink block diagram found in Appendix A.1.

### 3.6.1 Add saturation block to each SRC input $u_i$ branch

The purpose of this saturation block at this step is to initiate the split for the SRC at $v^* = 0$. On the left hand side of the split ($v < 0$), one MV saturates and on the right hand side ($v > 0$), the other MV saturates.

The saturation block passes the signal through when it is within the range, otherwise it returns a saturation value; the saturation block returns the upper bound limit value when the signal is higher than the upper limit and returns the lower bound limit value when the signal is lower.

We will assume $v^* = 0$ at this stage and if $v^* \neq 0$, $v^*$ will be added through a bias in the last step. To resolve the internal signal $v$ into the two input signals $u_1$ and $u_2$, we use the saturation block widget obtained from the Simulink library.

**Table 3.1:** Simulink saturation block properties - as adapted from Mathworks (2018) documentation

**(a)** Lower and upper limit settings for the saturation blocks

| Saturation block for | Upper and Lower bounds |
|---|---|
| $u_1$ | $[-100,\ 0]$ |
| $u_2$ | $[0,\ 100]$ |

**(b)** Outputs of the first saturation blocks for $u_1$ and $u_2$ signals

| Input signal $v$ | Output |
|---|---|
| When $v = -20$<br>(-ve $v$ means that the level is lower than setpoint) | $u_1 = -20$<br>$u_2 = 0$ |
| When $v = 20$<br>(+ve $v$ means that the level is higher than setpoint) | $u_1 = 0$<br>$u_2 = 20$ |

The upper and lower bounds of the saturation block are set to be large enough to accommodate large and small $v$. It does not matter what value is chosen as long as the magnitude of the limits is large enough.

### 3.6.2 Add the saturation bias to each SRC input $u_i$

The next step is to determine the minimum and maximum values of each MV$_i$, denoted as $u_{i,\ \mathrm{min}}$ and $u_{i,\ \mathrm{max}}$ respectively. These values are typically deemed to be 0 and 100% respectively for control valves.

Depending on which direction the control valve saturates, the saturation bias term $u_{i,\,\mathrm{sat}}$ is added to each of the $u_i$ signal in the Simulink model.

### 3.6.3 Add another saturation block to each SRC input $u_i$ branch

Another saturation block is added after the saturation bias is added. This time the block ensures that each $u_i$ signal is within the 0-100% range, as any signal that is out of the range coming into the block saturates at the limits. The block settings are given in Table 3.2.

**Table 3.2:** Second saturation block upper and lower bound settings

| **Saturation block for** | $u_i$ **operating range** |
|---|---|
| $u_1$ | $[0,\ 1]$ |
| $u_2$ | $[0,\ 1]$ |

### 3.6.4 Apply the slopes $\alpha_i$ to each SRC input $u_i$ branch

For each SRC input $u_i$ branch, apply the gain $\alpha_i$ multiplier to each signal to obtain $\alpha_i v$.

### 3.6.5 Add the split value $v^*$ bias to the internal signal $v$

In the last step, $v^*$ is evaluated. When $v^* = 0$, the split value bias added is zero.

A split value $v^*$ is added to the internal signal $v$ before it is resolved by each saturation block placed at the beginning of each input $u_i$ branch that represents a logic sequence within the SRC.

## 3.7 Another classical advanced control strategy

According to Allison and Isaksson (1998), one of the most popular mid-ranging controllers used in industrial applications is the valve-positioning controller. The philosophy behind mid-ranging control is as follows. The main valve is $u_2$ which provides the required capacity and is used together with the trim valve $u_1$ given that $u_2$ has poor resolution. In order to provide higher resolution in operations, the trim valve $u_1$ should return to its midpoint or target value. The rule of thumb is that since $u_1$ has low capacity, it should return to its mid-range value.

### 3.7.1 Valve positioning control (VPC) or input resetting

The valve positioning controller also adopts the multiple-input and single-output (MISO) control strategy, like the split-range-controller (SRC). For process applications, the VPC is used to operate two control valves, one which is a large valve for coarser adjustments and the other is a small valve for finer resolution control (Ola Slätteke, 2006). VPC uses two controllers, the first one controls $y$ using $u_1$ and the second intervenes at the mid-range, given by the setpoint $r_2$, with $u_2$ (Allison and Isaksson, 1998). The following description of VPC for optimal control is given with reference to Figure 3.6. Compared to SRC for optimal control, there are slight differences in the operating mechanism. The VPC controller, denoted as $C_2(s)$, is applied on an existing single-loop feedback controller $C_1(s)$, and measures the offset between the first input $u_1$ with the setpoint $r_2$ for the valve-positioning-controller. The decoupling filter $C_D(s)$ is inapplicable in VPC for optimal control as $u_2$ is normally paired with another feedback controller and only under certain conditions like $u_1$ is close to saturation, then $u_2$ is used to control $y$. This conditional use of VPC with normal feedback PID controller is possible when VPC is used together with selectors.

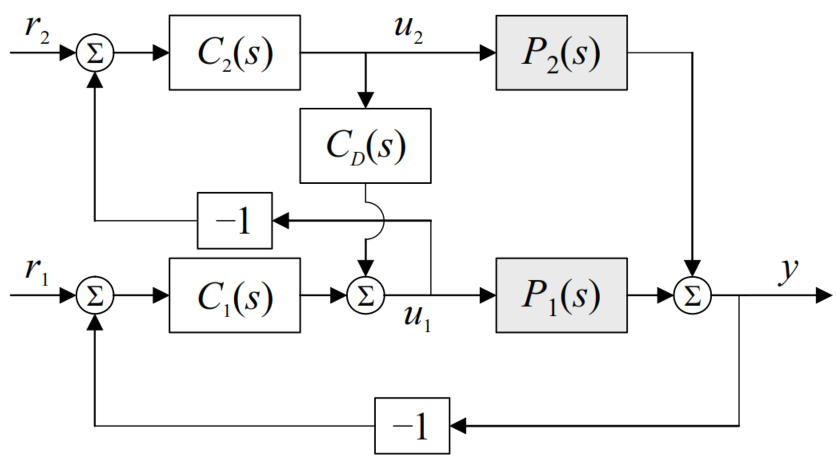



**Figure 3.6:** Block diagram of a valve positioning controller (Ola Slätteke, 2006)

When VPC is used for optimal control, the setpoint $r_2$ can be used to specify the equivalent of the split-value $v^*$ for the SRC. When $r_2$ is set, for example at 95% of $u_1^{\max}$, the controller $c_2(s)$ calculates the offset $e_2 = r_2 - u_1$ and returns a controller output $u_2$ for compensation. We can assume that $C_1(s)$ is direct acting on $u_1$. Usually $r_2$ is set as the threshold and a saturation bias is added to $u_2$. The purpose of the saturation bias is to keep $u_2$ in saturation when $u_1$ is less than $r_2$. In this instance, if $C_2(s)$ is reverse acting on $u_2$ when $u_1 > r_2$, then the control structure functions like the split-range-controller at $95\% \times u_1^{\max}$. However if $u_1$ optimally saturates at 100% valve opening, then the VPC configuration will always be inferior to the SRC and this comparison has been performed by Reyes-Lúa et al. (2018b). This is because the setpoint to the VPC has a backoff from $u_1^{\max}$.

# 4 — Case Study

## 4.1 Buffer tanks

The chemical industry generates value by converting raw materials to valuable end-products, sometimes involving two or more process plants or units. When two or more plants are involved in the conversion process, intermediary storage facilities also known as buffer tanks or tank farms are installed to provide temporary storage during the lead time between demand fulfillment or for subsequent downstream processing. Notwithstanding its purpose of holding inventory, buffer tank dampens flow or composition disturbance which propagates through the value chain, by momentarily adjusting its liquid volume. This is also known as *level averaging control* as the level controller averages out the incoming flow disturbances to control the outflow with a smoother flow profile (Faanes and Skogestad, 2003).

### 4.1.1 Process control of one buffer tank

The system which comprises of a single tank, with both inlet and outlet flows of an incompressible liquid. is the basis of the case study. There are two *degrees of freedom* at both the inlet and outlet flow of this tank. It is not possible to control both liquid level and flow rate on the same *degree of freedom* as both variables are invariably linked (Tippett and Bao, 2015). Quite simply put, if we change the level setpoint from 50% to 60%, the level controller will take action to close the inlet control valve, giving up the flow rate setpoint which was initially set. The discrepancy in the signals received by the arbitrary *degree of freedom* from the flow and level controllers explains the point made by Tippett and Bao (2015).

Though a relationship exists between these variables, flow rate has a *steady state effect* while liquid level in the tank has *no steady state effect* on process throughput. *Steady state effect* is the impact a controlled variable has on cost $J$. Flow rate directly influences throughput so it has an impact on cost whereas liquid level has to be controlled to prevent drifting of the process so it has no impact on cost. (Skogestad, 2004a) One should not automatically assume the *steady state effect* of a type of variable, but instead critcally examine the impact a variable has on the cost or loss function $J$. For example, liquid level in a continuously-stirred tank reactor (CSTR) may have a *steady state effect* on product composition if the amount of product formed from the homogeneous reaction is dependent on the volume of holdup in the reactor.

Changes in liquid level occur when there is accumulation or discharge of material in the tank. When both the inlet and outlet flow rates are equal, the liquid level in the tank holds constant by the principle of *conservation of mass*. As the operational objective is to maximize throughput for a given level setpoint, increasing or decreasing the flow rate will have an effect on the *throughput*. On the other hand, the choice of keeping liquid level at 50% or 80% has no impact on cost, as this volume of liquid held in the tank does not have an effect on system throughput. Nonetheless, to operate the process in a sustainable manner, it may be recommended to operate at level setpoint according to production schedule or within an acceptable range of liquid level (Lindholm and Johnsson, 2012).

### 4.1.2 Maintaining consistent inventory control for a tank system

According to Aske and Skogestad (2009), *consistency* of a system ensures that the steady state material balance (total, component and phase) is satisfied at the overall plant level. Additionally for the more stringent *local consistency* criteria, each individual unit or tank has to have one of its flow streams controlled by its inventory. The heuristics proposed by Aske and Skogestad (2009) ensure that *local consistency* is met for inventory control systems, by having to locate the TPM and check for *consistent* inventory control at all locations and the overall system (i.e. satisfying the steady state condition). The approach, however, does not include complexity, such as moving disturbances.

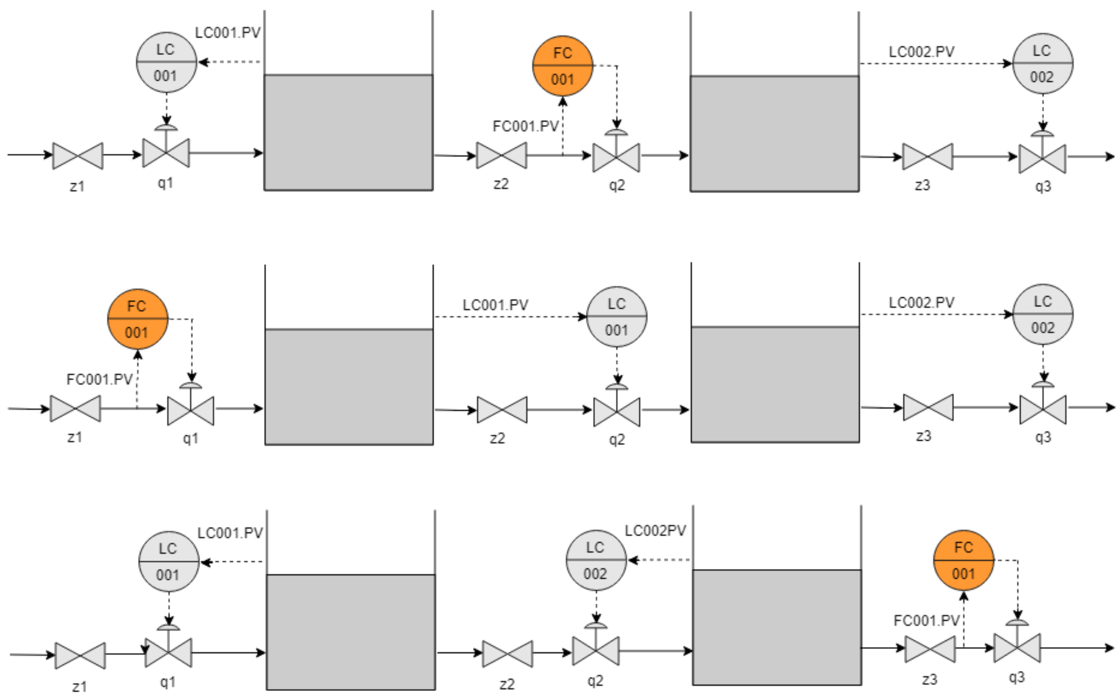



**Figure 4.1:** Operational modes of the two-tanks system to maintain local consistency when flow bottleneck moves around in $z_1$, $z_2$ and $z_3$

## 4.2 Optimal control of a single tank

This is a classic inventory tank system which is found in continuous processes of the chemical processing industry. This tank serves as material holdup for temporary inventory storage or blending facility for mixing of raw materials or products to meet plant specifications. Regardless of the tank's process design intent, the optimal control objective is to ensure that the system returns to or close to its nominal or steady state operating condition when process disturbances are introduced, while at the same time, maximizing flow throughput of the tank system. Two routine maintenance hand valves are manually operated from the field, both of which denoted as $z_1$ and $z_2$ for the inlet and outlet valves respectively. They simulate flow disturbance that may occur at either of the locations in this system. A bottleneck occurs at the point where closing the hand valve limits the flow through the network to the flow rate through it, with the assumption that the system is operating at steady state.

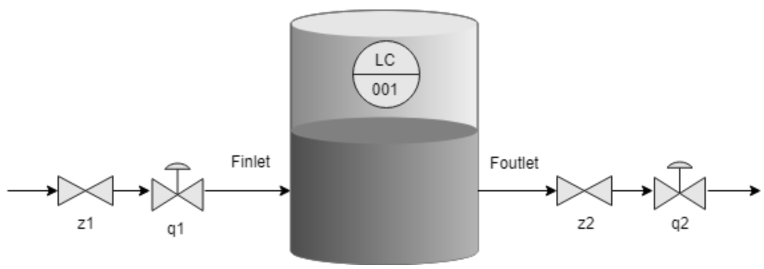



**Figure 4.2:** Single tank system

When one of the hand valves is closed, flow through that section is reduced by a fractional (0 - 100%) amount of the maximum allowable flow $F_{\max}$. In process modeling, this simplified tank setup could be extended to simulate actual bottlenecks in chemical plants as they arise from decreased process throughput due to process upsets[1], disruption from suppliers or a myriad of reasons that cause this disruption. Similarly, the goal will be to stabilize the system, while fulfilling the *economic objective* of throughput maximization. This should be carried out without having to reconfigure control structures or remind plant operators to intervene every time the disturbance moves to another location.

The ideal control structure should be able to select the control structure for optimal operation depending on the disturbance location; the operational modes are shown in Figure 2.3. As mentioned by Aske and Skogestad (2009), it is difficult to change the structure of the inventory control system after it is being implemented, so single loop controllers face moving bottlenecks in the flow throughput without having the mechanism to handle them. Previously, the limitations

[1] when there is a process upset, more process WIP (work-in-progress) is recirculated for reprocessing, reducing flow through the section momentarily. Note that each manual hand valve is taken as a black box for this process in this case.

of decentralized control strategies were addressed in Section 2.4.2 from the input saturation standpoint. Supervisory controllers may offer a solution to overcome these shortcomings and an ACS design for optimal control may be viable. Having established the process understanding and control objective for this case study, we proceed directly to step 2 of ACS design procedures, i.e. to prepare the list of constraints and rank these constraints in order of priority.

### 4.2.1 List of priority of constraints

We identify the constraints for this system. Valve sizing and pipe flow capacities are used to determine the input saturation constraints. The height dimension of the tank is obtained from vessel equipment sizing. The setpoint for flow throughput may be set based on the plant production schedule. With the gathered information, the constraints of this single tank system are tabulated in Table 4.1.

**Table 4.1:** Constraints of the single tank system

| **MV constraints** | **CV constraints** |
|---|---|
| $0 \leq F_{\text{inlet}} \leq F_{\text{inlet}}^{\max}$ | $h = h_{\text{sp}}$ |
| $0 \leq F_{\text{outlet}} \leq F_{\text{outlet}}^{\max}$ | $F = F_{\text{sp}}$ |

The physical MV constraints in Table 4.1 outline the feasible region for operation. These constraints are "hard" as they relate to the firm technical design specifications and should therefore be ranked highest in priority. Following next in the order of priority are sensitive and drifting variables. Lastly, the variables that are lowest in priority are controlled at the limiting or constant value in order to achieve optimal performance. They are the active constraints which are more important, followed by self-optimizing variables, which are less important as they are unconstrained variables by nature.

The concept of self-optimizing control or variable was coined by Morari et al. (1980) and it relates to having *unconstrained variables* controlled at a constant setpoint in order to achieve an acceptable loss $J_{\text{actual}}$ from the optimal loss $J_{\text{optimal}}$. The choice of whether to control the *self-optimizing* variables depends on whether there are unconstrained variables that need to be controlled to achieve an acceptable amount of loss from the optimum - see Section 2.3. In this case study, there are no *self-optimizing* variables because the variables are either active constraints or sensitive (drifting) variables that are controlled at a given setpoint.

With the list of MV and CV constraints, we formulate the priority list of MV and CV equality or inequality constraints with the rationale provided earlier.

(P1) **MV inequality constraints - Input saturation**

$$0 \leq F_{\text{inlet}} \leq F_{\text{inlet}}^{\max}$$

$$0 \leq F_{\text{outlet}} \leq F_{\text{outlet}}^{\max}$$

(P2) **CV equality constraint - Maintain constant inventory level in tank**

$$h = h_{\text{sp}}$$

(P3) **MV or CV equality constraint - Maximise throughput**

$F_{\text{throughput}} = F_{\text{sp}}$

### 4.2.2 Active constraint regions

Active constraints are variables that are always controlled at their limiting values to ensure optimal operation of the process. The active constraint regions are disturbance region spaces that are marked out where the active constraints are active (Reyes-Lúa et al., 2018b). These regions are marked by boundaries and when an operating point lies on the boundaries, the system is operating at the active constraint. Figure 4.3 provides insight on how the disturbances ($z_{\text{in}}$ and $z_{\text{out}}$) cause a change in active constraint region.

There are two MVs at the inlet and outlet streams of the tank, giving rise to two inequality constraints. There are three CVs, $F_{\text{inlet}}$, $F_{\text{outlet}}$ and $h$ but only two *degrees of freedom*. Any of the flow controller on $F_{\text{inlet}}$ or $F_{\text{outlet}}$ could be the throughput manipulator. At any given time, only one of the flow streams, $F_{\text{inlet}}$ or $F_{\text{outlet}}$ is controlled depending on which is closest to the bottleneck. When it is impossible to control the throughput active constraint, $F_{\text{throughput}}$ should equate to the flow rate that is most affected by the disturbance, i.e. $\min(z_{\text{in}}, z_{\text{out}})$. After the TPM is assigned to an input, the remaining *degree of freedom* is used for level control.

For illustrative purposes, we assign maximum flow $F_{\max} = 300$ and the throughput setpoint $F_{\text{sp}} = 240$ or 80% of the maximum flow. We also assume that field valve and piping specifications for both inlet and outlet are identical.

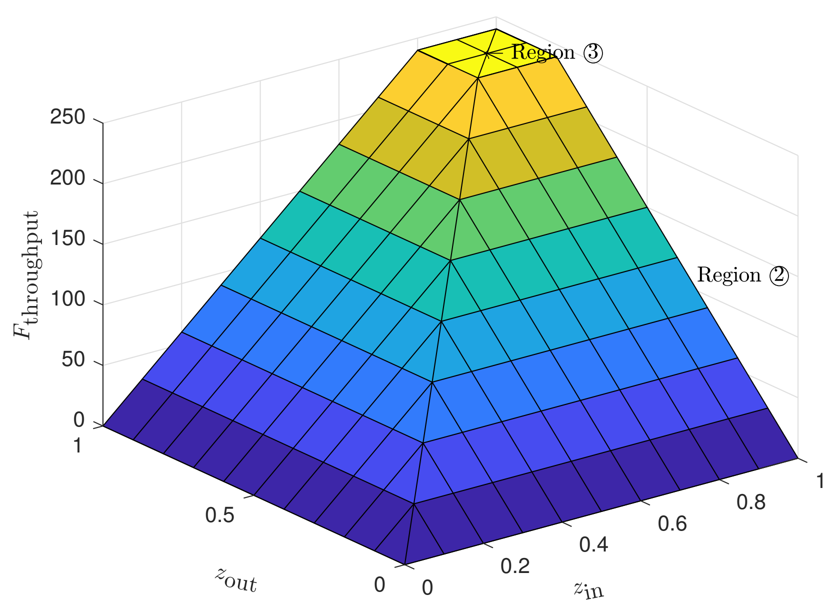



**Figure 4.3:** Active constraint regions for the single tank system

- **Region 1:** $F_{\text{throughput}} < \min(z_{\text{in}}, z_{\text{out}}) \cdot F_{\max}$

  $F_{\text{throughput}} > \min(z_{\text{in}}, z_{\text{out}}) \cdot F_{\max}$
- **Region 2:** $F_{\text{throughput}} = \min(z_{\text{in}}, z_{\text{out}}) \cdot F_{\max}$
- **Region 3:** $F_{\text{throughput}} = F_{\text{sp}}$ (when $F_{\text{sp}} = F_{\max}$, $F_{\text{throughput}} = F_{\max}$)

Region 1 covers the unsteady operating state, which includes tank filling or emptying procedure. Only the other two regions should be considered for steady state operation.

The pyramid slopes, both of which labeled Region 2, outline the operating regime where the ACS controls the active constraints at the flow rate that maximizes throughput given the hand valve disturbance. The TPM is set at the bottleneck flow rate, where the control valve is operated at full saturation for optimal operation. By closing the hand valve $z_i$ so that the flow rate falls below $F_{\text{sp}}$, the system gives up on this throughput active constraint. According to Aske et al. (2007), this mode of operation is known as optimized throughput and it occurs when the cost of operating at maximum throughput outweighs the cost of operating with decreased throughput, denoted as $F_{\text{sp}}$.

Lastly, the flat surface at the top of the pyramid corresponds to Region 3, which is the region where throughput is tracking the flow setpoint. If the flow setpoint is the maximum flow, then Region 3 converges to the point on the pyramid apex, where $F_{\text{throughput}} = F_{\text{sp}} = F_{\max}$.

### 4.2.3 Using SRC and selectors for the ACS control design

The general guidelines for using selectors and split-range are described as follows,

1. When $n_{\text{ac}} > n_{\text{u}}$, we use selectors to select the input-output pairing for different scenarios.
2. When $n_{\text{ac}} < n_{\text{u}}$, we use split-range to deliver one signal from the controller (output) to all the inputs that are used for control.

There are only 2 inputs $u$ and 3 CVs but this is not a problem as we need only to control one flow rate CV (one TPM for a single flow line) at any given time. However, the TPM location depends on where the bottleneck occurs so a `min` selector should be used. A selector is implemented on each *degree of freedom* so that either a signal from the liquid flow or tank level controller is used at a given time.

Pertaining to the level controller of the tank, we identified two inputs that are used to control holdup in the tank. We can justify a split-range for the level controller based on the guidelines aforementioned. One would note that the MV at the tank inlet is *direct-acting* while the MV at the outlet is *reverse-acting* on the tank level.

Figure 4.4 incorporates these design features to the control structure design of a single buffer tank.

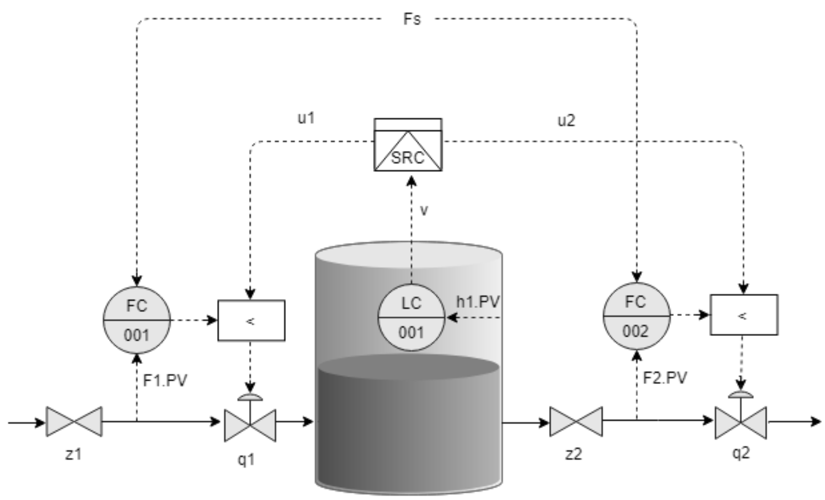



**Figure 4.4:** Single tank system with split-range control and selectors

### 4.2.4 Process Model

The control objective should be aligned with the process design intent for implementation of robust control. To understand the process intent of the buffer tank, we should ask if the tank level controller should be tuned for incoming *flow disturbance rejection* or *tank level setpoint tracking* (Reyes-Lúa et al., 2018a). With the formulation of the process model, the meaning of this question will become clearer.

#### Physical model

The buffer tank system can be modeled with the dynamic mass balance equation. Additionally, the disturbance variables, denoted as $z$, represent the manually operated hand valve openings that restrict the flow throughput.

$$\frac{\mathrm{d}h}{\mathrm{d}t} = \frac{1}{A}(z_{\mathrm{in}}q_{\mathrm{in}} - z_{\mathrm{out}}q_{\mathrm{out}}) \tag{4.1}$$

| | | |
|---|---|---|
| $h$ | : | tank level (controlled variable) |
| $z_{in}$ and $z_{out}$ | : | % opening of the manual hand valves at tank inlet and outlet |
| $q_{in}$ and $q_{out}$ | : | volumetric inlet and outlet flow rates |
| $A$ | : | base area of the tank |

**Operating point of the tank network**

The nominal flow rate of the process is 300 $\mathrm{m}^3\ \mathrm{min}^{-1}$. This yields the time constant of $\tau = V/q^* = 120000\ \mathrm{m}^3/300\ \mathrm{m}^3\ \mathrm{min}^{-1} = 400$ min. $\tau$ is also known as the nominal *residence time* of the tank. Typically, the buffer tank design uses residence time instead of tank volume (Faanes and Skogestad, 2003).

The other variables also assume nominal values.

**Linear model in deviation variables**

The tank balance equation (4.1) is written in terms of deviation variables. The deviation variables for the state variables ($h$), manipulated variables ($q_{\mathrm{in}}$ and $q_{\mathrm{out}}$) and the disturbance variables ($z_{\mathrm{in}}$ and $z_{\mathrm{out}}$) are used.

$$\Delta h = h - h^*, \quad \Delta q_{\mathrm{in}} = q_{\mathrm{in}} - q_{\mathrm{in}}^*, \quad \Delta q_{\mathrm{out}} = q_{\mathrm{out}} - q_{\mathrm{in}}^*$$
$$\Delta z_{\mathrm{in}} = z_{\mathrm{in}} - z_{\mathrm{in}}^*, \quad \Delta z_{\mathrm{out}} = z_{\mathrm{out}} - z_{\mathrm{out}}^*$$

$$\frac{\mathrm{d}\Delta h}{\mathrm{d}t} = \frac{1}{A}(\Delta z_{\mathrm{in}} \Delta q_{\mathrm{in}} - \Delta z_{\mathrm{out}} \Delta q_{\mathrm{out}}) \tag{4.2}$$

We simulate the tank as a linear system and take the Laplace Transform of its variables. The process model is written as,

$$H(s) = G_{\mathrm{u}}(s)\ u(s) + G_{\mathrm{d}}(s)\ d(s) \tag{4.3}$$

where

$$\begin{aligned} G_{\mathrm{u}}(s) &= [G_{q,\,\mathrm{in}}(s) \qquad G_{q,\,\mathrm{out}}(s)] \\ G_{\mathrm{d}}(s) &= [G_{z,\,\mathrm{in}}(s) \qquad G_{z,\,\mathrm{out}}(s)] \\ u(s) &= [Q_{\mathrm{in}} \qquad Q_{\mathrm{out}}]^{\mathsf{T}} \\ d(s) &= [z_{\mathrm{in}} \qquad z_{\mathrm{out}}]^{\mathsf{T}} \end{aligned}$$

On taking Laplace Transform in (4.2),

$$sH = \frac{1}{A}(Z_{\text{in}}Q_{\text{in}} - Z_{\text{out}}Q_{\text{out}}) = \frac{Q_{\text{in}}}{A}Z_{\text{in}} - \frac{Q_{\text{out}}}{A}Z_{\text{out}} \tag{4.4}$$

**Table 4.2:** Calculation of process gain for $\mathrm{G_d}$ and $\mathrm{G_u}$

| **Transfer Function** | | **Process gain** $\left(k' = \frac{K_p}{\tau}\right)$ |
|---|---|---|
| *$G_d$, disturbance transfer functions* | | |
| $G_{Z,\text{ in}} = \frac{H}{Z_{\text{in}}}$ | $= \frac{Q^*_{in}}{sA}$ | $\frac{300}{6000} = 0.05$ |
| $G_{Z,\text{ in}} = \frac{H}{Z_{\text{out}}}$ | $= -\frac{Q^*_{out}}{sA}$ | $-\frac{300}{6000} = -0.05$ |
| *$G_u$, process transfer functions* | | |
| $G_{q,\text{ in}} = \frac{H}{Q_{\text{in}}}$ | $= \frac{Z^*_{in}}{sA}$ | $\frac{1}{6000} = 0.000167$ |
| $G_{q,\text{ out}} = \frac{H}{Q_{\text{out}}}$ | $= -\frac{Z^*_{out}}{sA}$ | $-\frac{1}{6000} = -0.000167$ |

$\therefore$ The matrix form of the linear tank equation system (4.3) is (4.5),

$$\begin{aligned} H(s) &= G_d(s)\ d(s) + G_u(s)\ u(s) \\ &= \begin{bmatrix} \dfrac{0.05}{s} & \dfrac{-0.05}{s} \end{bmatrix} \begin{bmatrix} Z_{\text{in}} \\ \\ Z_{\text{out}} \end{bmatrix} + \begin{bmatrix} \dfrac{0.000167}{s} & \dfrac{-0.000167}{s} \end{bmatrix} \begin{bmatrix} Q_{\text{in}} \\ \\ Q_{\text{out}} \end{bmatrix} \end{aligned} \tag{4.5}$$

### 4.2.5 Controller tuning

#### Control Objective

If the control objective for optimal operation of the tank is to *maximize throughput*, then the tank level should be held steady as long as possible to minimize inventory building up in the system. The control objective for this tank system will be incoming flow *disturbance rejection* to keep the tank inventory holdup as steady as possible.

#### SIMC tuning rules

From eq. (4.5), we identified that the tank system is an integrating process. The level SRC may be tuned with SIMC rules for integrating processes (see Section 3.2.5) and the parameters are calculated using the process gain derived earlier (see Table 4.2). Alternatively, if the process model is too complex to derive, it is also possible to estimate the parameters by an open-loop step test.

$$K_c = \frac{1}{k'} \frac{1}{(\tau_c + \theta)} \tag{4.6}$$

$$\tau_I = 4(\tau_c + \theta) \tag{4.7}$$

#### $\tau_c$ selection

The selection of $\tau_c$ values are $\tau$, $\frac{\tau}{2}$, $\frac{\tau}{4}$ , ... $\tau_c$ should not be longer than the residence time of the tank ($\tau$), else the flow disturbance would have propagated out of the tank for $\tau_c > \tau$.

#### Time delay $\theta$

The time delay $\theta$ is zero for this tank system. Although no time delay is not realistic in practice, the objective of the simulations is to assess SRC performance in managing moving active constraints when input saturates so we use a simplified process model.

With the tank model derivation, the controller tunings for both *flow disturbance rejection* and *level setpoint tracking* are obtained. However, the objective of the simulations is to assess the split-range controller performance in rejecting flow disturbance of the tank, so the controller tunings are chosen accordingly.

**Table 4.3:** Calculation of $K_c$ using SIMC rules (4.6) for disturbance rejection and setpoint tracking for the two inputs

| **Process Gain,** $k'$ | **Controller gain,** $K_{\mathbf{c},i}$ | | |
|---|---|---|---|
| $k' = \frac{K_p}{\tau}$ | $\tau_c = \tau$ | $\tau_c = \frac{\tau}{2}$ | $\tau_c = \frac{\tau}{32}$ |
| *Flow disturbance rejection* | | | |
| 0.05 | $\frac{1}{0.05}(\frac{1}{400})$ = 0.05 | $\frac{1}{0.05}(\frac{1}{200})$ = 0.1 | $\frac{1}{0.05}(\frac{2}{25})$ = 1.6 |
| -0.05 | $\frac{1}{-0.05}(\frac{1}{400})$ = -0.05 | $\frac{1}{-0.05}(\frac{1}{200})$ = -0.1 | $\frac{1}{-0.05}(\frac{2}{25})$ = -1.6 |
| *Level setpoint tracking* | | | |
| 0.000167 | $\frac{1}{0.000167}(\frac{1}{400})$ = 14.97 | $\frac{1}{0.000167}(\frac{1}{200})$ = 29.9 | $\frac{1}{0.000167}(\frac{2}{25})$ = 479 |
| -0.000167 | $-\frac{1}{0.000167}(\frac{1}{400})$ = -14.97 | $-\frac{1}{0.000167}(\frac{1}{200})$ = -29.9 | $-\frac{1}{0.000167}(\frac{2}{25})$ = -479 |

$[K_{c,\text{inlet}},\ K_{c,\text{outlet}}]$ corresponds to [-1.6, 1.6] given that $\tau_c = \frac{400}{32}$ min and the controller is **tuned for flow disturbance rejection**. The integral time for the PI controller is calculated using the SIMC tuning rules for an integrating process in eq. (4.7), and the numerical value is obtained from eq. (4.8), where $\tau = 400$ min, $\tau_c = \frac{400}{32}$ min and $\theta = 0$. Skogestad (2004b) recommends eq. (4.6) and 4.7 to find a good trade-off between disturbance rejection and controller robustness to avoid slow oscillations; it is obviously necessary to use this parameter with the $K_c$ calculated from eq. (4.6) and presented in Table 4.3.

$$\tau_I = 4 \cdot \frac{400}{32} = 50 \text{ min} \tag{4.8}$$

### Determining the slopes $\alpha_i$ of the split-range block or splitter

The slopes $\alpha_i$ can be determined from the effective controller gain, $K_{c,i}$ and assigning the gain of the normal PID controller within the SRC to be $K_c$.

**Table 4.4:** Determination of the slopes $\alpha_i$ for SRCs given that $K_c = 1.6$

| **Effective controller gain, $K_{\mathbf{c},i}$** | **Slopes $\alpha_i$ of SRC** |
|---|---|
| 1.6 | = 1 |
| -1.6 | = -1 |

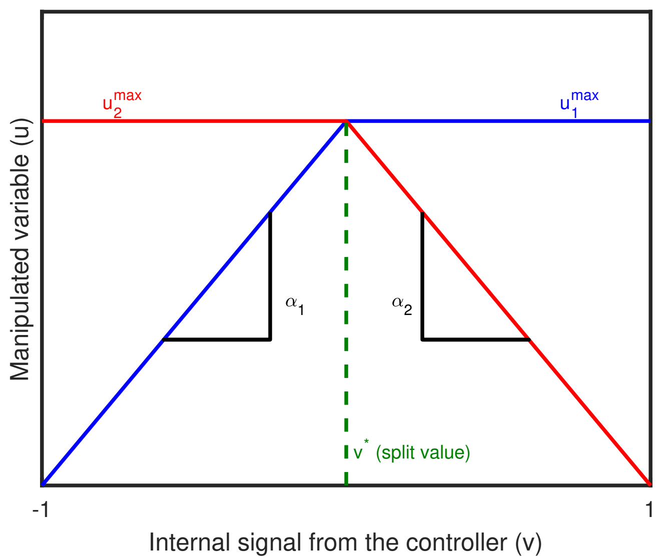



**Figure 4.5:** Slopes $\alpha_i$ of the split-range controller inputs

$\tau_I$ need not be defined in the split-range block as this integral time constant is only applied to the PI-controller within the SRC.

## 4.3 Optimal control of tanks in series using SRC

In the earlier case study, the split-range control scheme was designed for the single tank system to perform optimal control. Back then, the control objective was to maximize flow throughput of a single tank and hold the inventory level of that tank constant. We also noted that the split-range control scheme without selectors qualified as a multiple-input and single-output or MISO control strategy. With selectors, the split-range control scheme was upgraded to a multiple-input and multiple-output or MIMO control strategy.

When the problem scales up to include multiple tanks arranged in series (Figure 4.6), the requirement is for the inventory levels for all tanks to be held constant. Since there are multiple control objectives to be met by the control system, more than one SRC is needed. We write the list of constraints like how it was done for the single tank case. The additional step is to group the constraints and active constraints according to the SRC they belong to. It is also possible that the constraint is handled by both SRCs. The idea is to design the SRC for a subset of the process independently of the other. Like before, rank the constraints and perform the SRC control structure design for each group.

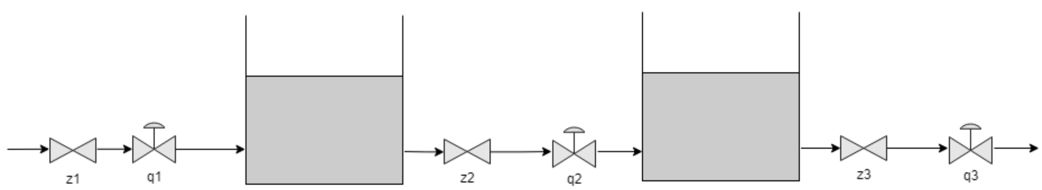



**Figure 4.6:** Two Tanks System

Each of the two SRC schemes is designed independently of each other. When their designs are completed, they are combined to obtain the final ACS design for the network of tanks. When combining both SRCs, the challenge lies in the flow section where the outlet of one tank is the inlet for the other. The control valve $q_2$, which manipulates the flow through this section, affects the level of both tanks. There are a total of three signals that could be used by $q_2$, signal from the SRCs for the left and right tanks and the flow controller. Based on the previous guideline where $n_{\mathrm{ac}} > n_u$, a selector should be used and the next question is whether a `min` or `max` selector should be used. The control objective is to maximize the throughput flow so we should set the TPM at the bottleneck flow rate.

In the single tank case, $F_{\mathrm{throughput}} = \min(z_{\mathrm{in}}, z_{\mathrm{out}}) \times F_{\max}$ evaluates the TPM for the single tank case. Likewise for the two tanks case, we apply the concept to the second tank to design the second SRC. To maximize $F_{\mathrm{throughput}}$ for both tanks, the TPM flow rate should be set at $\min(z_1, z_2, z_3) \times F_{\max}$. From the scenarios conceived in Table 4.5, it is shown that by applying the `min` selector on the signals from both SRC, $u_{12}$ and $u_{21}$ in Figure 4.7, for $q_2$, the objective of maximizing throughput the network may be met.

**Table 4.5:** Bottleneck flow rate determination for the single and two tanks system

| Scenario | $(z_1,\ z_2,\ z_3)$ | Bottleneck of | | |
|---|---|---|---|---|
| | | **Tank 1** $\mathbf{min}(z_1, z_2)$ | **Tank 2** $\mathbf{min}(z_2, z_3)$ | **Both tanks** $\mathbf{min}(\mathbf{min}(z_1, z_2), \mathbf{min}(z_2, z_3))$ |
| 1 | (1.0, 0.8, 0.6) | 0.8 | 0.6 | 0.6 |
| 2 | (0.5, 0.9, 1.0) | 0.5 | 0.9 | 0.5 |
| 3 | (0.8, 0.4, 1.0) | 0.4 | 0.7 | 0.4 |

Assuming that disturbances ($z_i < 1$) are introduced at multiple locations, the selected signal for the control valve $q_2$ should be that of the **bottleneck flow rate for both tanks** to force the input ($q_i$) closest to the bottleneck to saturate.

The final consideration is the flow setpoint $F_{\text{sp}}$ which is set to override any $F_{\text{throughput}}$ computed previously. Also, $F_{\text{throughput}} = F_{\text{sp}}$ corresponds to Region 3 of the active constraint region in Figure 4.3. This happens when $F_{\text{sp}} < F_{\text{bottleneck}}$.

### 4.3.1 SRC control scheme for the two tanks system

Figure 4.7 displays the SRC control structure design for the two tanks system with the abovementioned design features.

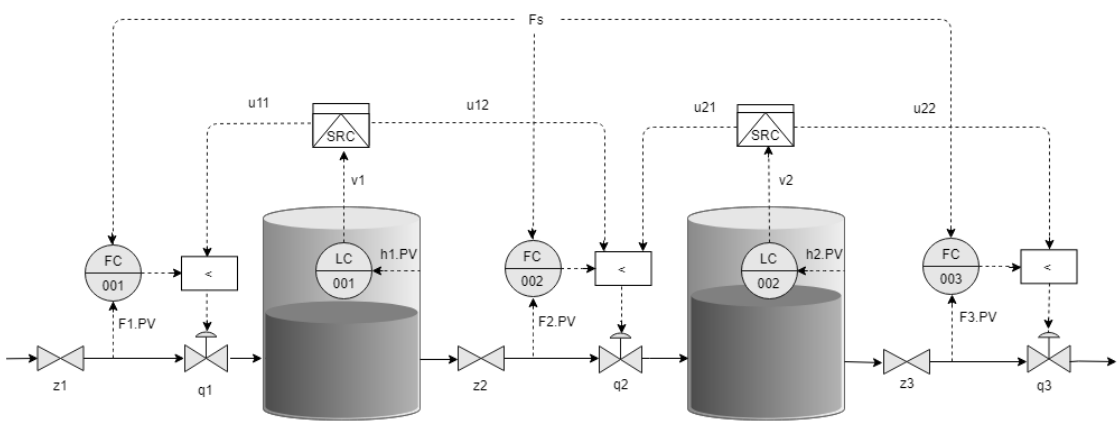



**Figure 4.7:** Two tanks system with the split-range control and selectors

# 5 — Performance of the Split-Range Control

The previous chapter formalized the advanced controller design using SRC with selectors for the two-tanks system. There are two criteria assessed for SRC performance. Firstly, we assess how well the level is restored for both tanks when disturbance is introduced at each hand valve location. Secondly, we evaluate whether the SRC selects the right control strategy for optimal operation so that the TPM is always positioned closest to the bottleneck.

To eliminate the offset in the tank levels, the PI controller is used for both SRCs.

## 5.1 Simulations

In this chapter, the simulations are performed in Simulink. The simulation cases are hereby listed.

(i) Closed loop response of the *proportional-only controller* with disturbance at $z_1$ – Section 5.2

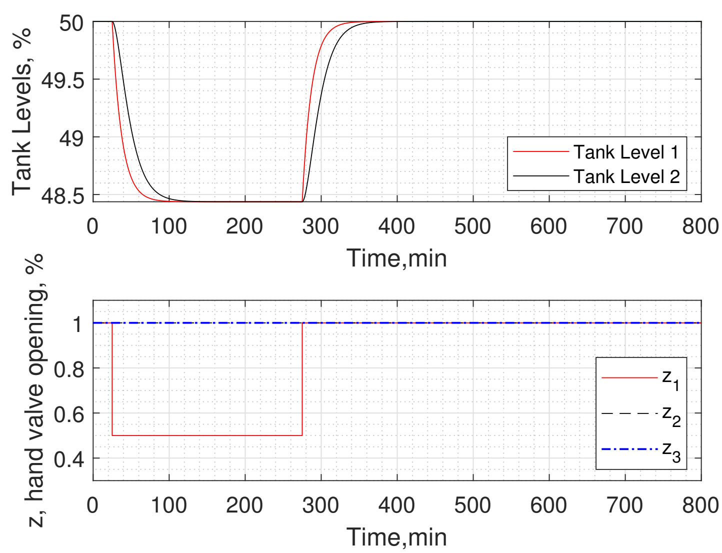
(ii) Closed loop response of *PI-controller* with disturbance at $z_1$ – Section 5.3

(iii) Flow throughput setpoint change with and without anti-windup – Section 5.4

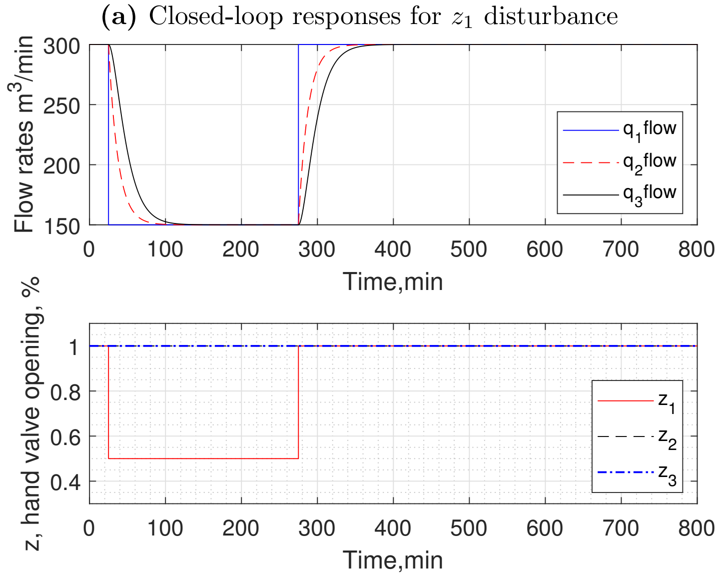
## 5.2 Closed loop response for P-controller

The closed loop response to a disturbance caused by a 50% closure of the hand valve $z_1$, located at the inlet of the tank system, for the level proportional-only controller is performed.

### 5.2.1 Flow disturbance at $z_1$




**(a)** Closed-loop responses for $z_1$ disturbance




**(b)** Flow throughput given $z_1$ disturbance

**Figure 5.1:** Level and flow process values given $z_1$ disturbance

Two step changes in the flow disturbance occur at 25 and 275 min respectively. At 25 min time, the hand valve $z_1$ closes by 50% and then at 275 min, it recovers by fully opening again.

The tank level profile shows that with the introduced disturbance at the inlet, the proportional-only controller takes corrective action to match the outlet flow with the inlet which has a decreased flow rate. This controller takes not further action to remove the level offset as there is no integral action to calculate the amount of offset that was previously accrued when the level is below the setpoint in order to compensate the flow rate to bring the level back to the setpoint.

The gain $K_c$ in the level controller only compensates for the offset by matching the inlet and outlet flow rates. The offset between the level setpoint and process value is not eliminated, so level does not go back to 50%.

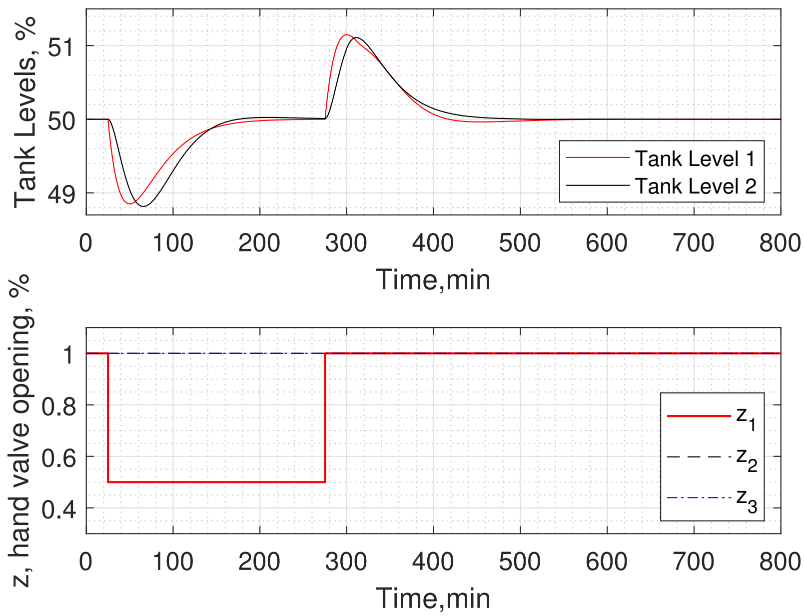
## 5.3 Closed loop response for PI controller

The closed loop response to a disturbance caused by a 50% closure of the hand valve $z_1$, located at the inlet of the tank system, for the level PI-controller is performed.

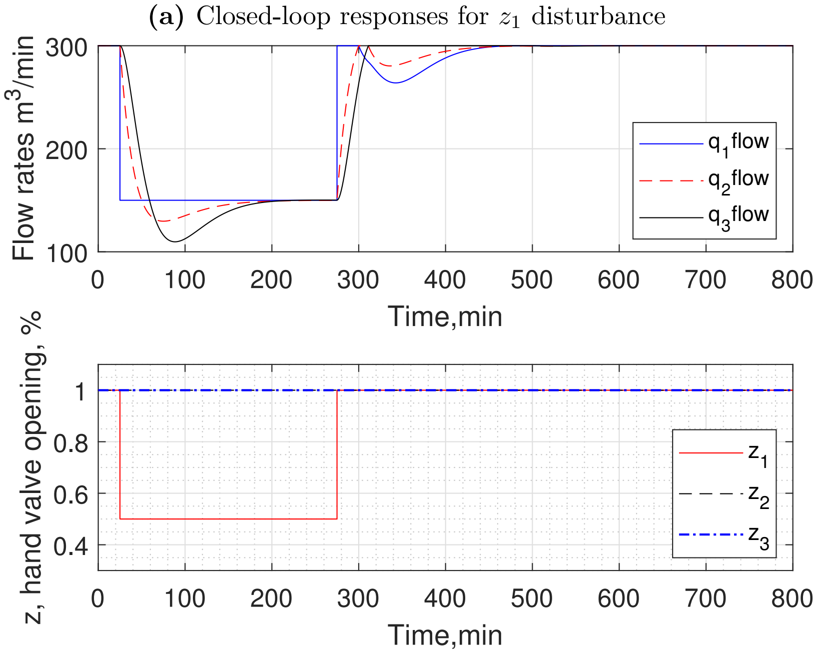
### 5.3.1 Flow disturbance at $z_1$

Two step changes in the flow disturbance occur at 25 and 275 min respectively. At 25 min time, the hand valve $z_1$ closes by 50% and then at 275 min, it recovers by fully opening again.

When inlet flow to Tank 1 decreases by closing $z_1$, flow rate through $q_1$ decreases and forms the bottleneck of the system. $q_1$ optimally saturates fully open to maximize flow rate through the bottleneck. $q_2$ also closes to match the decreasing inlet flow rate to Tank 1. This controller action helps to maintain Tank 1 at its setpoint. As $q_2$ closes, Tank 2 inventory level also falls and $q_3$ also closes to hold Tank 2 level at its setpoint.




**(a)** Closed-loop responses for $z_1$ disturbance




**(b)** Flow throughput given $z_1$ disturbance

**Figure 5.2:** Level and flow process values given $z_1$ disturbance

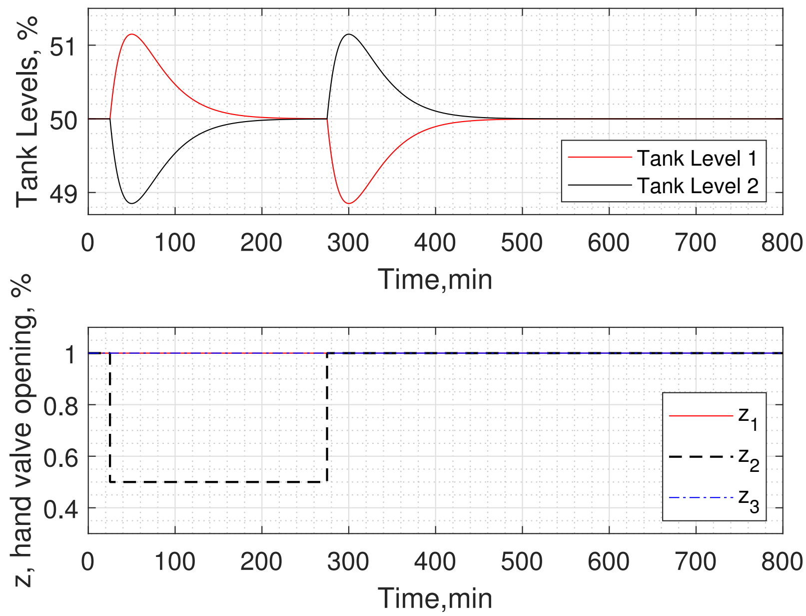
The remaining results for SRC performance when flow disturbance is introduced at hand valves $z_2$ and $z_3$.

### 5.3.2 Disturbance at $z_2$

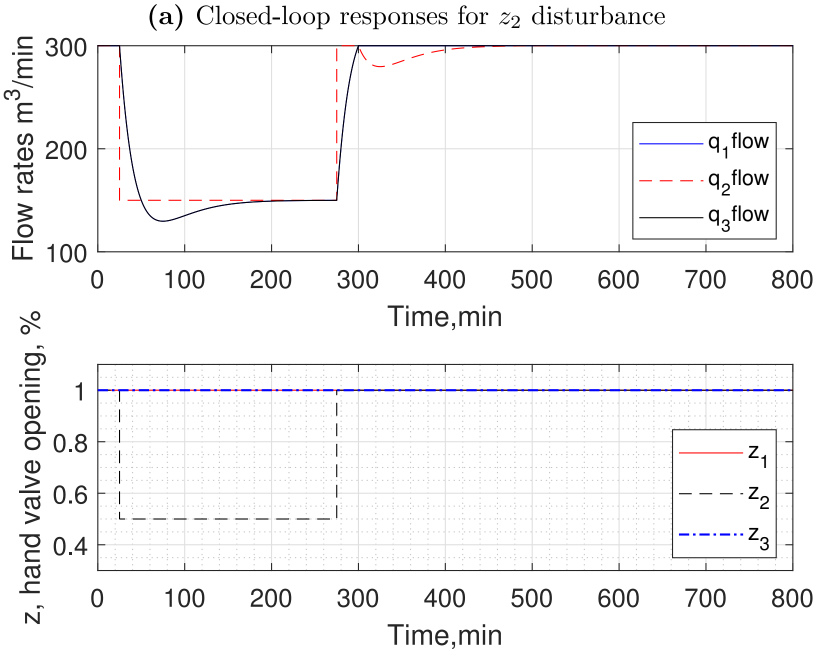
The closed loop response to a disturbance caused by a 50% closure of the hand valve $z_2$, located at the inlet of the tank system, for the level PI-controller is performed.

Two step changes in the flow disturbance occur at 25 and 275 min respectively. At 25 min time, the hand valve $z_3$ closes by 50% and then at 275 min, it recovers by fully opening again.

When inlet flow to Tank 2 decreases by closing $z_2$, flow rate through $q_2$ becomes the bottleneck. $q_2$ remains fully open as flow rate should be maximized at the bottleneck. $q_1$ closes as Tank 1 level increases. As flow rate through $q_2$ decreases, Tank 2 level decreases and so $q_3$ closes.




**(a)** Closed-loop responses for $z_2$ disturbance

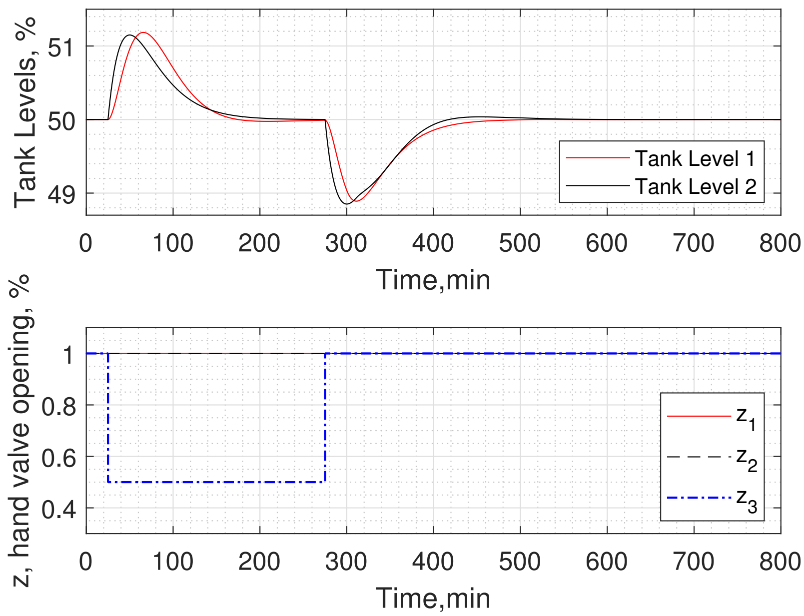



**(b)** Flow throughput given $z_2$ disturbance

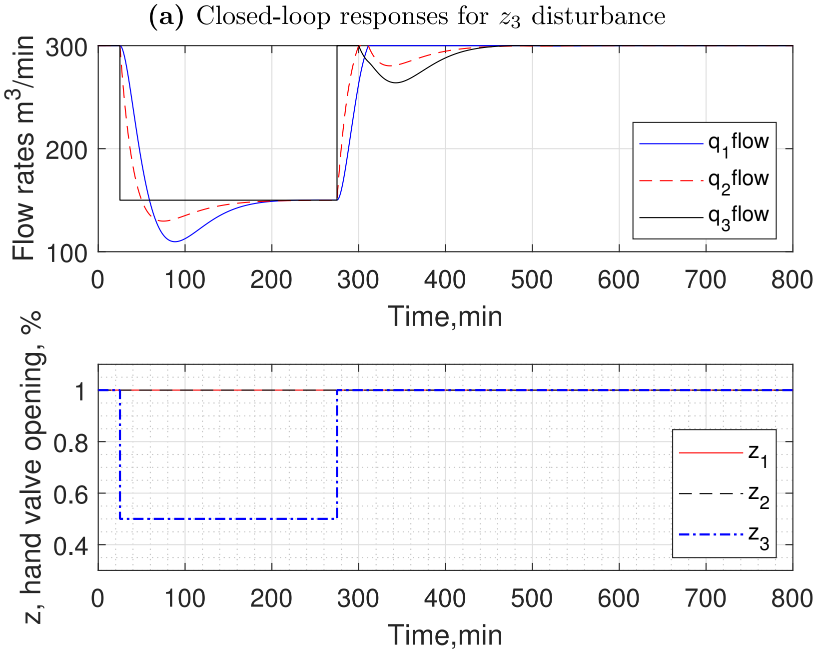
**Figure 5.3:** Level and flow process values given $z_2$ disturbance

### 5.3.3 Disturbance at $z_3$

The closed loop response to a disturbance caused by a 50% closure of the hand valve $z_3$, located at the inlet of the tank system, for the level PI-controller is performed.




**(a)** Closed-loop responses for $z_3$ disturbance




**(b)** Flow throughput given $z_3$ disturbance

**Figure 5.4:** Level and flow process values given $z_3$ disturbance

Two step changes in the flow disturbance occur at 25 and 275 min respectively. At 25 min time, the hand valve $z_3$ closes by 50% and then at 275 min, it recovers by fully opening again.

When outlet flow of Tank 2 is decreased by closing $z_3$, flow rate through $q_3$ becomes the bottleneck. $q_3$ remains fully open because flow should be maximized at the bottleneck. $q_2$ closes in response to the rising Tank 2 level. As $q_2$ closes, Tank 1 level also rises and $q_1$ closes to bring the level back to the setpoint.

## 5.4 Throughput rate $F_{\mathrm{sp}}$ step response

The change in throughput rate $F_{\mathrm{sp}}$ on the tank level is studied.

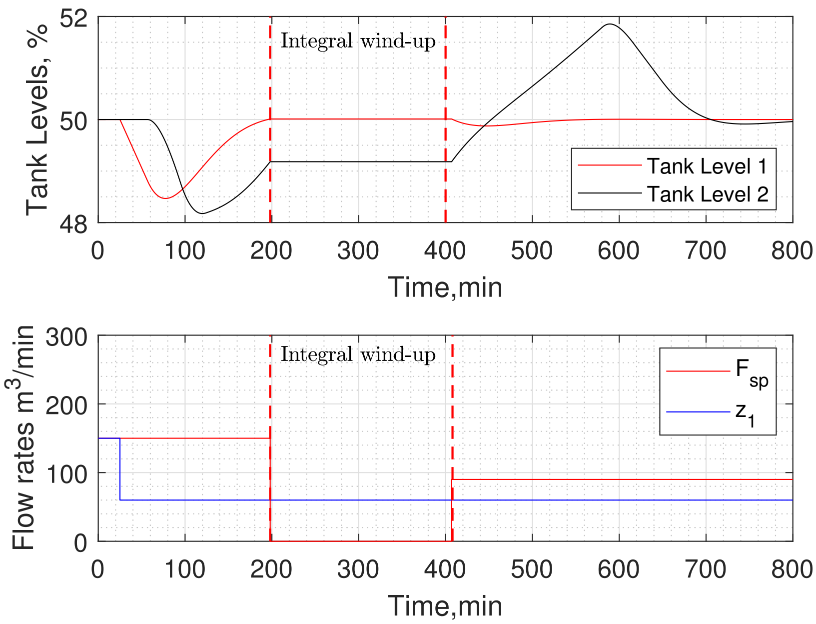



**Figure 5.5:** Integral action windup when $F_{\mathrm{sp}} = 0$ is set between $t = 200$ and 400 min

As always, the nominal tank levels is set at 50%. After closing the hand valve $z_1$ at about $t = 20$ min, the tank levels are allowed to return to the nominal values before commencing on the flow throughput setpoint change.

At $t = 200$ min, the flow throughput $F_{\mathrm{sp}}$ is abruptly set to zero and all the control valves $q_1$, $q_2$ and $q_3$ shut completely. The offset between Tank 2 level process value and setpoint remains for the next 200 minutes. At 400 mins, $F_{\mathrm{sp}}$ is increased to 100 m$^3$/min. During the period when Tank 2 level is rising, $q_3$ is closed as the PI controller unwinds the offset. The overshoot of the level process value reached to approximately 52%. From that point on, the level SRC takes the corrective action to bring the overshoot level value back to the setpoint.

### Integral windup

The integral windup scenario occurs when the throughput rate $F_{\mathrm{sp}}$ setpoint abruptly changes to zero and the control valves saturate and do not take any action to correct the level offset. The integral action from the PI controller continues to integrate the offset between the level setpoint and the actual process value for Tank 2. When the operator resumes the flow throughput by changing its setpoint, an overshoot in Tank level 2 occurs.

### 5.4.1 Adding anti-windup to the PI controller

The antiwindup mechanism is added to the PI feedback controller within the SRC to place a limit on how high the integrated error value can be. This feature prevents large overshoots when the integrated error is unwound.

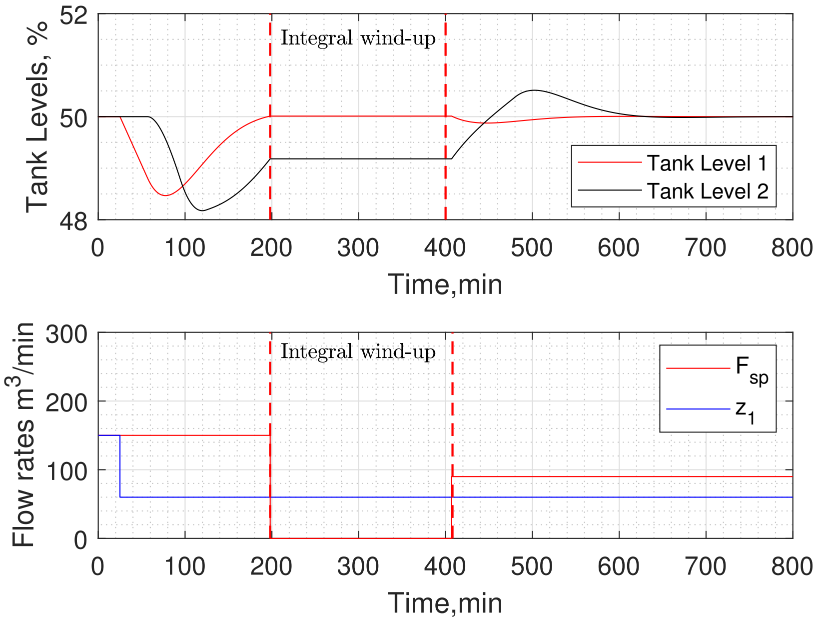



**Figure 5.6:** Anti-windup with saturation limits, between +5% and -5% of tank level, added to the PI feedback controller within the SRC

The saturation limits for the anti-windup must be applied for both the ± range of $e$. The *back-calculation* anti-windup method is used.

# 6 — Discussion

## 6.1 Perstorp AB Utilities Disturbance Management

This analysis is taken in the context of Lindholm and Johnsson (2012).

The economic effects of flow disturbances are felt significantly in the utilities industry, i.e steam and cooling water production plants. The economic loss arising from flow disturbances in utilities operation is substantial but they are either not tracked by operations or kept as proprietary information within the company. An optimal selection of a control strategy like the one proposed in this thesis for moving bottlenecks can serve as a study case for implementation of optimal control for buffer tanks.

Lindholm and Johnsson (2012) provides an insight on how Perstorp AB applies a framework, known as the Utilities Disturbance Management (UDM), at an industrial site in Stenungsund. They proposed two main strategies in the framework for plantwide disturbance management, Table 6.1, that are similar to the control objectives of our ACS design, i.e. to maximize flow throughput or control at flow setpoint $F_{\mathrm{sp}}$ while holding the buffer tank levels constant.

1. Choice of buffer tank levels
2. Control of product flow

**Table 6.1:** Strategies for product flow disturbances at Perstorp AB (Lindholm and Johnsson, 2012)

### 6.1.1 Choice of buffer tank levels

Buffer tanks used in the utilities industry can provide backup in the event when there is a supply disruption upstream. Maintaining higher inventory levels than necessary means that the process provisions more holdup than required, while having lower inventory levels means that the tank will not be able to provide sufficient buffering capacity in the supply disruption event. The trade-off is to find an inventory level that handles most disturbance cases. Lindholm and Johnsson (2012) proposed minimum inventory levels that can handle 90% of utility disturbances for each product at Perstorp AB. The desired inventory levels applies to all buffer tanks within the designated network through which each product flows.

The ACS control scheme proposed for the moving bottleneck case could be used as an inspiration to design the control system for the plant site of Perstorp AB. If the buffer tanks are designated within the same sector and connected to the flow network, the desired inventory level setpoint can be applied to all targeted tanks. If needed, the proposed ACS for the two tanks can be scaled up to manage more tanks in the flow network.

### 6.1.2 Choice of product flow

During the utilities disturbance, operations have to plan how to control the product flow rate to serve an area. There are uncertainties in estimating (i) how long the disturbance will last and (ii) the amount of product that flows to the customer. This decision making may be a production scheduling problem but when the flow rates are decided based on the estimated disturbance duration, they are subsequently implemented on the supervisory control layer. If the control scheme is similar to the ACS design used for the moving bottleneck case, the forecast of flow rate can be used as the flow throughput setpoint $F_{\text{sp}}$ and changed on demand, if necessary. Disturbance handling along the flow network will be managed by this advanced control structure design, which can select control strategy for optimal operation.

## 6.2 Limitations of the SRC control scheme

The decision to scale up the original SRC control scheme for the one tank case to include more SRCs to handle more tanks in the network is done not without its disadvantages. The problem arises when flow disturbance happens upstream in the network of tanks. Since the control objective is to maximize throughput while keeping inventory levels in the tank constant, the flow is significantly reduced when the disturbance in one location of the network affects the flow throughput across the entire system. The assumption for ACS design is that the process will always be in steady state, but in practice the inventory levels in the tanks could change when the upstream supply is disrupted and throughput has to be maintained downstream. As observed in Figure 6.1, when one of the hand valves is suddenly shut, all the SRC inputs will close, bringing the flow at all points to a complete stall. For this supply disruption scenario, the operator could intervene to lower the level SRC setpoint to release inventory from each tank, beginning with the tank closest to the customer. This demand make-up to customers discharges flow from the closest-to-customer buffer stock, which is also known as the last-in-first-out (LIFO) stock flow policy. In a nutshell, it is not acceptable to always operate at the bottleneck flow rate of the system and there are occasions when we need to isolate disrupted areas and operate the network in the unsteady mode.

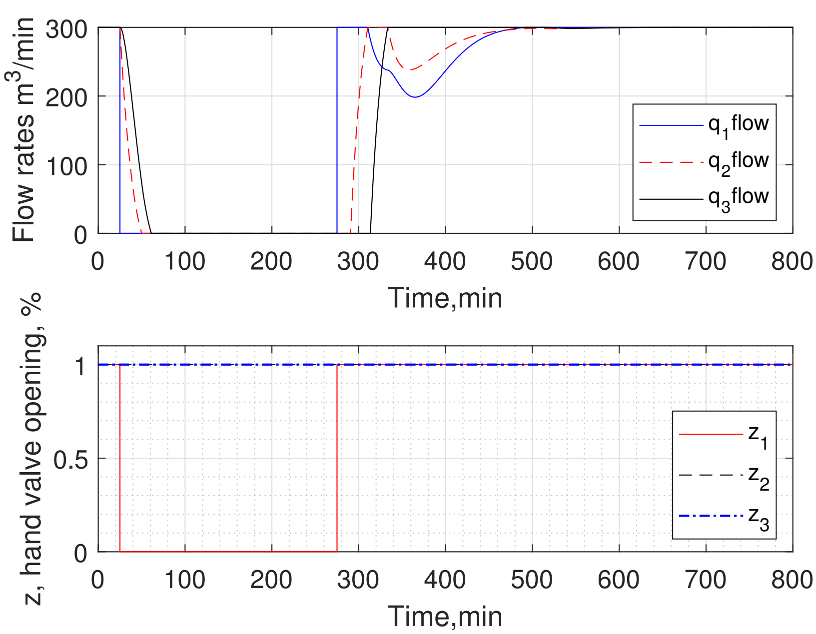
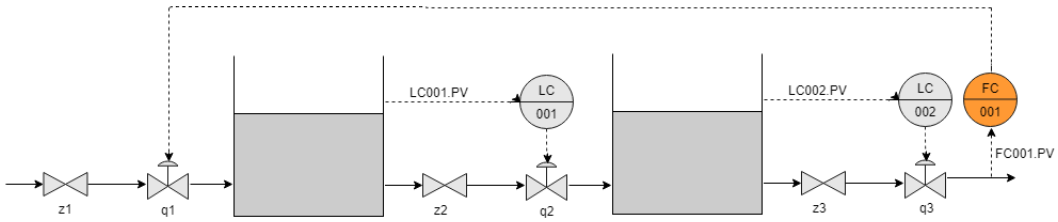
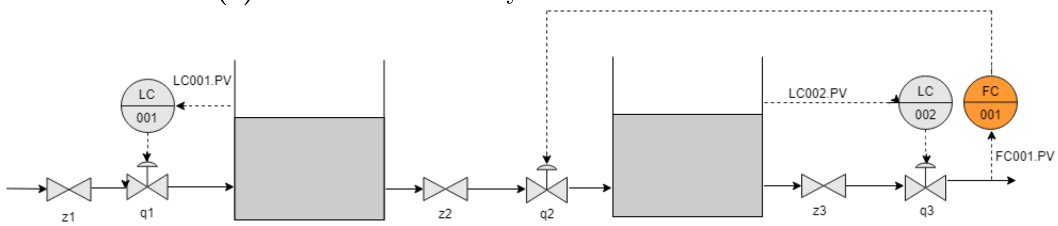


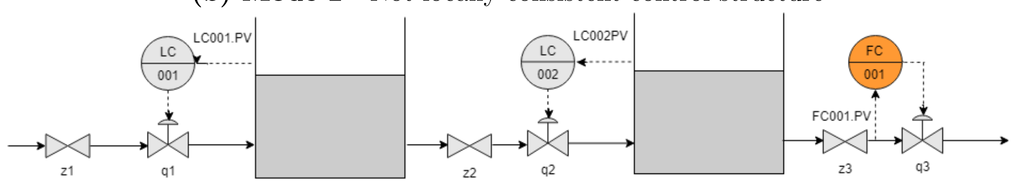

**Figure 6.1:** Fully closed hand valve $z_1$ simulating supply disruption at the flow inlet




**(a) Mode 1** - Not locally consistent control structure




**(b) Mode 2** - Not locally consistent control structure




**(c) Mode 3** - Only locally consistent control structure

**Figure 6.2:** Inventory control structures with flow controller at the tank network outlet

This brings about other ideas on how the flow controller or TPM can be positioned at the outlet of the tank network to bypass flow disruptions upstream by using existing inventory in the downstream tanks to supply customers. Control structures in Figure 6.2 could be the wish-list of an operator as they can use a mode selector and choose $q_1$, $q_2$ or $q_3$ to control the downstream flow, rather than using the level controller output to estimate the flow rate to customer. For example, in the event of serious bottleneck occuring at $z_1$ such as a total supply cut-off, Figure 6.2b is activated to bypass the affected region, using the inventory buffer levels to continue servicing the downstream. The problem with these control structure designs in Figure 6.2a and 6.2b is that they are not consistent[1], meaning that they lead to *unsteady state* operations of the flow network so they are not good inventory control practices (Aske and Skogestad, 2009). Other inconsistent inventory control loops proposed by operators at Perstorp AB can be found in Forsman (2016).

[1]not consistent inventory control means that either one of the inlet or outlet flows of each mass holdup is not controlled by their corresponding mass holdup

## 6.3 Limitations of the split-range controller

The next limitation relates to PI-controller tunings for SRCs in general. The integral action of the PI controller removed the error offset well (Figure 5.2) because the slopes $|\alpha_i|$ for both inputs are equal to 1 so they do not materially change the effective $K_{c,i}$ gains of the inputs from the controller gain $K_c$. This essentially means that the $K_{c,i}$ and $\tau_I$ tuning parameters are identical with those obtained from SIMC tuning rules for tight and robust control. However, if $|\alpha_i| \neq 1$ or $|\alpha_i|$ of the inputs are not equal, the $\tau_I$ calculated for PID controller still applies to all the SRC inputs. As a result, this is **not** the same set of tight and robust tuning parameters which are calculated if SIMC rules were applied to individual single-input and single-output feedback controllers. Each single-loop PI controller has a unique $\tau_I$, but an SRC PI-controller has the same $\tau_I$ calculated for all inputs.

# 7 — Conclusion

This work sets out to search for an advanced control structure that can perform control strategy selection for optimal operation of an inventory control system.

The tank in series case is found in numerous literature and the moving bottleneck problem poses a challenge for existing control structures to manage flow disturbances which may occur at various locations. This problem serves as a motivation for us to rethink existing single-loop control structures applied to this process. The *input saturation rule* serves as a starting point for us to evaluate whether single-loop control structures yield good *input-output pairings*. The desired pairings consider which inputs optimally saturates and these inputs are paired with the less important controlled variables which can be given up when disturbance occurs. If switching of the optimal pairings is required, as an input which is originally paired with a more important controlled variable saturates due to disturbance, then the use of advanced control structures should be considered for the control structure design. When the bottleneck moves within the two-tanks system, the control valve which optimally saturates (fully open) at the original bottleneck does not switch out the flow for level pairing accordingly for the single-loop controller case. The flow controller, being the throughput manipulator, sets the flow rate of the system so it has to be **placed close to the bottleneck to maximize flow** as a rule of thumb for steady state optimal operation.

The advanced control structure design procedures are used to design the controller for two-tanks in series, taking into account of the moving bottleneck problem. Firstly, ACS design was considered for a single tank case. After the priority list was created, the split-range controller was implemented with the guidelines for split-range and selectors. This was done by comparing the number of inputs and active constraints for each system, i.e the tank with inlet and outlet control valves system and each individual control valve system. Split-range was used for each tank system and selector was used on each control valve. Secondly, the SRC design for one tank had to be scaled up to two tanks. The challenge of combining the two SRCs was faced at the center control valve which controls both the outlet flow of the first tank and inlet flow of the second tank. This was resolved by applying the idea of controlling that valve at the bottleneck flow rate of the two tanks, since this was the maximum flow rate that could flow through the two-tanks system under steady state conditions.

Flow simulations through the two-tanks were then performed using Simulink. The performance of the SRC in handling disturbances at various locations for both the proportional-only and proportional-integral controllers were investigated. The proportional-only controller provided stability to the flow. There are no under or overshoots of flow for this configuration, but the trade-off is that the tank level never recovers back to its nominal operating level. The proportional-integral controller removes the offset of inventory level by tightly controlling the level at the given setpoint.

To sum up, the proposed SRC with selectors control structure design can perform automatic bottleneck handling for the inventory control system, by systematically operating in changing active constraint regions. This work has solved the moving bottleneck problem that cannot be handled by single-loop feedback controllers.

# Bibliography


B. J. Allison and A. J. Isaksson. Design and performance of mid-ranging controllers. *Journal of Process Control*, 8(5-6):469–474, 1998. ISSN 09591524. doi: 10.1016/S0959-1524(98)00012-2. URL http://www.sciencedirect.com/science/article/pii/S0959152498000122.

Elvira Marie B. Aske and Sigurd Skogestad. Consistent inventory control. *Industrial and Engineering Chemistry Research*, 48(24):10892–10902, 2009. ISSN 08885885. doi: 10.1021/ie801603j.

Elvira Marie B. Aske, Sigurd Skogestad, and Stig Strand. Throughput maximization by improved bottleneck control. *IFAC Proceedings Volumes (IFAC-PapersOnline)*, 40(5):63–68, 2007. ISSN 14746670. doi: 10.3182/20070606-3-MX-2915.00009.

K. J. Åström and T. Hägglund. The future of PID control. *Control Engineering Practice*, 9 (11):1163–1175, 2001. ISSN 09670661. doi: 10.1016/S0967-0661(01)00062-4.

B.Wayne Bequette. *Process Control: Modeling, Design and Simulation*. Prentice Hall PTR, 1st edition, 2002.

Eduardo F. Camacho and Carlos Bordons. *Model Predictive Control*. Springer-Verlag London, 1999. ISBN 1049-8923.

Vinicius de Oliveira, Johannes Jäschke, and Sigurd Skogestad. Optimal operation of energy storage in buildings: Use of the hot water system. *Journal of Energy Storage*, 5:102–112, 2016. ISSN 2352152X. doi: 10.1016/j.est.2015.11.009.

Audun Faanes and Sigurd Skogestad. Buffer tank design for acceptable control performance. *Industrial and Engineering Chemistry Research*, 42(10):2198–2208, 2003. ISSN 08885885. doi: 10.1021/ie020525v.

Robert Feldmann, Markus Hammer, Ken Somers, and Joris Van Niel. Buried treasure: Advanced analytics in process industries. *McKinsey Global Institute*, pages 1–8, 2017.

Krister Forsman. Implementation of advanced control in the process industry without the use of MPC. *IFAC-PapersOnLine*, 49(7):514–519, 2016. ISSN 24058963. doi: 10.1016/j.ifacol.2016.07.394.

Truls Larsson and Sigurd Skogestad. Plantwide control — A review and a new design procedure Scheduling Site-wide optimization Local optimization ( hour ) Supervisory. *Science*, pages 209–240, 2000. ISSN 0332-7353. doi: 10.4173/mic.2000.4.2.

Anna Lindholm and Charlotta Johnsson. Reducing revenue loss due to disturbances in utilities using buffer tanks - A case study at Perstorp Reducing Revenue Loss due to Utility Disturbances using Buffer Tanks – A Case Study at Perstorp. *Proceedings of the 17th Nordic Process Control Workshop*, pages 6–8, 2012.

Mathworks. Mathworks Documentation - Saturation, 2018. URL https://se.mathworks.com/help/simulink/slref/saturation.html.

Vladimiros Minasidis, Sigurd Skogestad, and Nitin Kaistha. *Simple Rules for Economic Plantwide Control*, volume 37. Elsevier, 2015. ISBN 1570-7946. doi: 10.1016/B978-0-444-63578-5.50013-X.

Manfred Morari, Yaman Arkun, and George Stephanopoulos. Studies in the Synthesis of Control Structures for Chemical Processes. *New York*, 2(2), 1980. ISSN 15475905. doi: 10.1002/aic.690260205.

Ola Slätteke. Modeling and Control of the Paper Machine Drying Section. *PhD Thesis, Lund University. Sweden*, pages 148–153, 2006.

Randel M. Price, Philip R. Lyman, and Christos Georgakis. Throughput Manipulation in Plantwide Control Structures. *Industrial and Engineering Chemistry Research*, 33(5): 1197–1207, 1994. ISSN 15205045. doi: 10.1021/ie00029a016.

S Joe Qin and Thomas A Badgwell. A survey of industrial model predictive control technology. *Control Engineering Practice*, 11:733–764, 2003. ISSN 09670661. doi: 10.1016/S0967-0661(02)00186-7.

Adriana Reyes-Lúa and Cristina Zotică. Systematic Design of Split Range Controllers. *DYCOPS CAB, 12th IFAC Symposium on Dynamics and Control of Process Systems, Florianópolis - SC, Brazil, April 23-26, 2019*, 2019.

Adriana Reyes-Lúa, Christoph Josef Backi, and Sigurd Skogestad. Improved PI control for a surge tank satisfying level constraints. *IFAC-PapersOnLine*, 51(4):835–840, 2018a. ISSN 24058963. doi: 10.1016/j.ifacol.2018.06.125.

Adriana Reyes-Lúa, Cristina Zotică, Tamal Das, Dinesh Krishnamoorthy, and Sigurd Skogestad. Changing between Active Constraint Regions for Optimal Operation: Classical Advanced

Control versus Model Predictive Control. *Computer Aided Chemical Engineering*, 43: 1015–1020, 2018b. ISSN 15707946. doi: 10.1016/B978-0-444-64235-6.50178-9.

Dale Seborg, Thomas Edgar, and Duncan Mellichamp. *Process Dynamics and Control*. John Wiley & Sons, Inc., 2nd edition, 2003.

Sigurd Skogestad. Control structure design for complete chemical plants. *Computers and Chemical Engineering*, 28(1-2):219–234, 2004a. ISSN 00981354. doi: 10.1016/j.compchemeng.2003.08.002.

Sigurd Skogestad. Simple analytic rules for model reduction and PID controller tuning. *Modeling, Identification and Control*, 25(2):85–120, 2004b. ISSN 03327353. doi: 10.4173/mic.2004.2.2.

Sigurd Skogestad and Chriss Grimholt. *PID Control in the Third Millennium*. Springer-Verlag London Limited, 2012. ISBN 978-1-4471-2424-5. doi: 10.1007/978-1-4471-2425-2. URL http://link.springer.com/10.1007/978-1-4471-2425-2.

Sigurd Skogestad and Ian Postlethwaite. *Multivariable Feedback Control - Analysis and Design*. John Wiley & Sons, Ltd., second edition, 2005.

Michael J. Tippett and Jie Bao. Distributed control of chemical process networks. *International Journal of Automation and Computing*, 12(4):368–381, 2015. ISSN 17518520. doi: 10.1007/s11633-015-0895-9.

Peter Van Overschee and Bart De Moor. RAPID: The End of Heuristic PID Tuning. *IFAC Proceedings Volumes*, 33(4):595–600, 2000. ISSN 14746670. doi: 10.1016/S1474-6670(17)38308-8. URL http://linkinghub.elsevier.com/retrieve/pii/S1474667017383088.

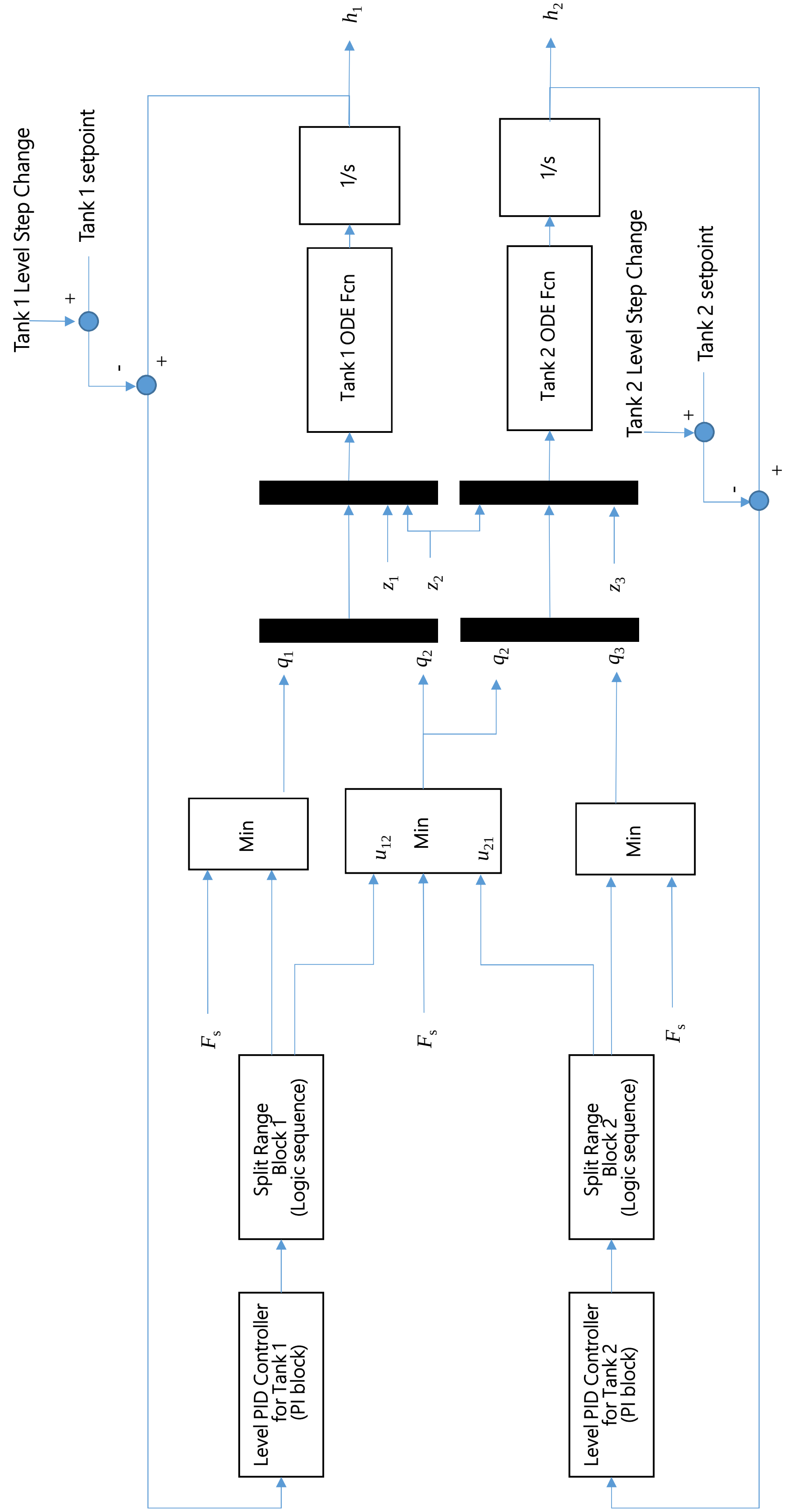
Rajagopalan Viknesh, Natarajan Sivakumaran, Jakka Sarat Chandra, and Thota K. Radhakrishnan. A critical study of decentralized controllers for a multivariable system. *Chemical Engineering and Technology*, 27(8):880–889, 2004. ISSN 09307516. doi: 10.1002/ceat.200402034.

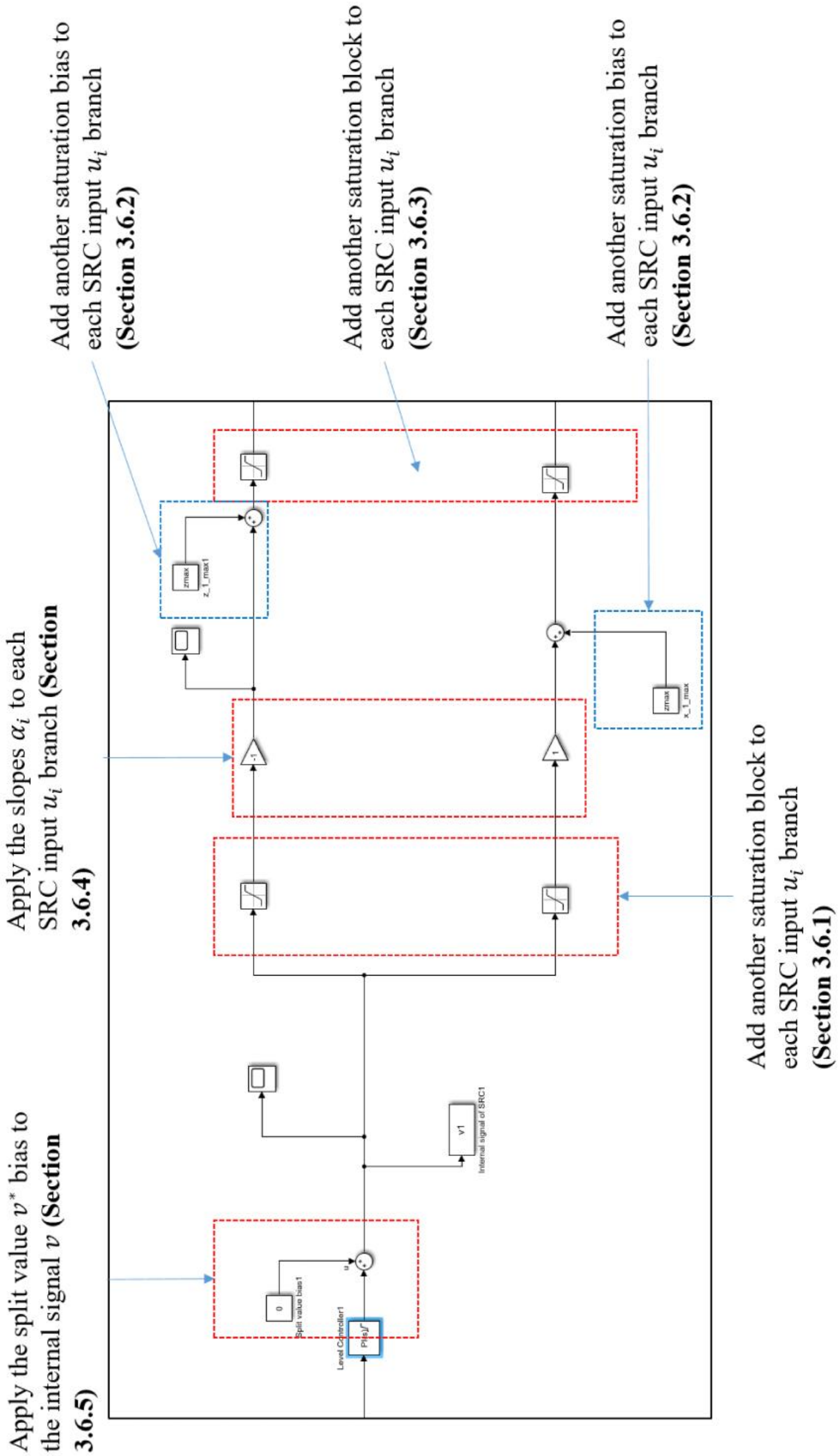
# A — Simulink Block Diagram

## A.1 Simulink block diagrams for SRC design

Two schematics are hereby presented.

- The simplified schematic of the Simulink block diagram developed for the two SRC control structure
- Step-by-step walk through for the SR logic block creation in Simulink (steps are described in Section 3.6 and the logic is explained in Section 3.3)


Tank 1 Level Step Change
Tank 1 setpoint
Tank 2 Level Step Change
Tank 2 setpoint
$h_1$
$h_2$
1/s
1/s
Tank 1 ODE Fcn
Tank 2 ODE Fcn
$z_1$
$z_2$
$z_3$
$q_1$
$q_2$
$q_2$
$q_3$
Min
Min
Min
$u_{12}$
$u_{21}$
$F_s$
$F_s$
$F_s$
Split Range Block 1 (Logic sequence)
Split Range Block 2 (Logic sequence)
Level PID Controller for Tank 1 (PI block)
Level PID Controller for Tank 2 (PI block)


Add another saturation bias to each SRC input $u_i$ branch **(Section 3.6.2)**
Add another saturation block to each SRC input $u_i$ branch **(Section 3.6.3)**
Add another saturation bias to each SRC input $u_i$ branch **(Section 3.6.2)**
Apply the slopes $\alpha_i$ to each SRC input $u_i$ branch **(Section 3.6.4)**
Add another saturation block to each SRC input $u_i$ branch **(Section 3.6.1)**
Apply the split value $v^*$ bias to the internal signal $v$ **(Section 3.6.5)**
zmax
z_1_max1
x_1_max
-1
1
v1
Internal signal of SRC1
Split value bias1
Level Controller1

# B — MATLAB Source Code

This section provides the MATLAB source code and code documentation from within the code.

## B.1 `main.m`

`main.m` controls program execution and initializes most variables that are used in Simulink. There are, however, some variables that do not appear in this code excerpt, like those that are written directly from Simulink to the MATLAB workspace.

```
% main script for simulating the dynamic tank model, and also the
% control structures.
% Select file you want to run by in sim('filename')
%
% Block comment (Ctrl + R); block uncomment (Ctrl + T)
%
% ************************************************************************
% @author: Amos Fang (adapted from Cristina Zotica HEX Temperature SRC)
% @organization: Process Systems Engineering, NTNU
% @project: Master Thesis 2019
% @since: Jan 2019
% @requires: MATLAB R2018b (not tested in other releases)
% @description: Main script for simulating the split-range control
% structure of two tanks
% @estimated run time: N/A
%
%
%
%
% ************************************************************************
% MATLAB commands
%
% t1       : program start-time
% ************************************************************************
clc
clear

t1=cputime;

%% ***********************************************************************
%  Information about the tank
%
% hmax     : maximum liquid level of the tank [m]
% A        : base tank area [m^2]
%
% ************************************************************************

hmax = 20;
A = 6000;           % Base Area of Tank

%% ***********************************************************************
%  Nominal operating conditions for the tanks
%
```

```
% Fsp      : throughput flow setpoint [-]
% q        : maximum throughput flow rate [m^3/min]
% d1       : nominal hand valve position (z1) [-]
% d2       : nominal hand valve position (z2) [-]
% d3       : nominal hand valve position (z3) [-]
%
% Define the states. The tank states are assigned x
% x10      : initial height of Tank 1 inventory [m]
% x20      : initial height of Tank 2 inventory [m]
% ***********************************************************************

Fsp = 1;
q = 300;

d1 = 1;
d2 = 1;
d3 = 1;

x10 = 10;
x20 = 10;

%% **********************************************************************
% Simulink Variables
%
% zmax    : Saturation bias for each input (MVs) of the SRC
%         :(Typically 0 - fully closed and 1 - fully open) [-]
% d1Tstep : Time when the disturbance is introduced to manual hand valve
%         : z1 [min]
% d2Tstep : Time when the disturbance is introduced to manual hand valve
%         : z2 [min]
% d3Tstep : Time when the disturbance is introduced to manual hand valve
%         : z3 [min]
% interval: Time interval as a multiple of d1Tstep, d2Tstep, d3Tstep
%         : when disturbance step remains [-]
% ***********************************************************************
zmax = 1;
d1Tstep = 25;
d2Tstep = 25;
d3Tstep = 25;
interval = 10;

%% **********************************************************************
% Simulink Variables for throughput flow setpoint step change
%
% FspStep1: 1st throughput flow setpoint step change [% of q]
% FspStep2: 2nd throughput flow setpoint step change [% of q]
% Fsp1Time: Time when 1st throughput flow setpoint change is introduced
% Fsp2Time: Time when 2nd throughput flow setpoint change is introduced
% F1on : Switch for first throughput flow setpoint step change(1:on 2:off)
% F2on : Switch for second throughput flow setpoint step change(1:on 2:off)
% ***********************************************************************
```

```

FspStep1 = -0.5;
FspStep2 = 0.5;

Fsp1Time = 18;
Fsp2Time = 37;

F1on = 0;
F2on = 0;

%% ***********************************************************************
% Simulink Variables for flow disturbance (moving bottleneck)
%
% d1on     : Switch for disturbance at hand valve z1(1:on 2:off)
% d2on     : Switch for disturbance at hand valve z2(1:on 2:off)
% d3on     : Switch for disturbance at hand valve z3(1:on 2:off)
% dStep    : disturbance step for hand valve closure
% ************************************************************************

d1on = 1;
d2on = 0;
d3on = 0;
dStep = 0.5;

%% ***********************************************************************
% Setpoint tracking of the SRC
% r1       : Tank 1 level [m]
% r2       : Tank 2 level [m]

% ************************************************************************

r1 = 10;                 % corresponds to 50% of the Tank 1 level
r2 = 10;                 % corresponds to 50% of the Tank 2 level

%% ***********************************************************************
% Simulink Variables for the level split-range controller
%
% r1Δ   : Tank 1 level setpoint step change
% r2Δ   : Tank 2 level setpoint step change
% r1on      : Switch for Tank 1 level setpoint change
% r2on      : Switch for Tank 2 level setpoint change
% t1SPstart : Start time for Tank 1 level step change
% t2SPstart : Start time for Tank 2 level step change
% ************************************************************************

r1Delta = 0; %0.1;
r2Delta = 0;
r1on = 0;
r2on = 0;
t1SPstart = 20;
t2SPstart = 20;
```

```matlab

%% ************************************************************************
% Controller tunning for SRC
%
% PI controller tuning parameters calculated with SIMC rules
%
% ************************************************************************

Kc = -1.6;
tauI = 50;

%% ************************************************************************
% MATLAB commands
%
% Run the two-tanks Simulink file; save variables into .mat file
%
% p        : Tank parameters
% t2       : program elapsed-time
%
% ************************************************************************

p = [A; q];
sim('TwoTanks')
t2 = t1-cputime;

h1 = y1;                 % Tank 1 ODE
h2 = y2;                 % Tank 2 ODE


save('outputTwoTanksSRC.mat','t','h1','h2','q1','q2', 'q3', ...
     'Kc','tauI')

%% ************************************************************************
% Plotting of results
%
% Figure 1 - subplot of the Tank levels 1 and 2 with time [%]
%          - subplot of hand valve openings with time [%]
% Figure 2 - subplot of SRC signals to control valve q2
%          - subplot of hand valve openings with time [%]
% Figure 3 - subplot of volumetric flow rate with flow disturbance [m^3/min]
%          - subplot of hand valve openings with time [%]
% Figure 4 - subplot of volumetric flow rate with flow throughput setpoint
%            change [m^3/min]
%          - subplot of hand valve openings with time [%]
%
% ************************************************************************

% ************************************************************************
% Figure 1
%
% This figure displays the plot of the inventory level in both Tank 1
```

```
% and 2 when a step change in d1, d2 or d3 is introduced.
%
% ***************************************************************************

figure(1)
set(figure(1),'Color','White')
subplot(211)
plot(t,h1*100/hmax,'r')
hold on
plot(t,h2*100/hmax,'k')
hold off
xlabel('Time,min')
ylabel('Tank Levels, %')
f2 = legend('Tank Level 1','Tank Level 2','Location','Southeast');
set(f2,'FontSize',10)
ax = gca;
ax.FontSize = 12;
grid on
grid minor

subplot(212)
plot(t,z1,'r')
hold on
plot(t,z2,'--k')
hold on
plot(t,z3,'-.b','LineWidth',1)
hold off
xlabel('Time,min')
ylabel('z, hand valve opening, %')
ylim([0.3 1.1]);
legend('z_1','z_2','z_3','Location','Southeast')
ax = gca;
ax.FontSize = 12;
grid on
grid minor

% conditional save
if d1on == 1
    saveas(gcf,'h1-dis','epsc')
elseif d2on == 1
    saveas(gcf,'h2-dis','epsc')
elseif d3on == 1
    saveas(gcf,'h3-dis','epsc')
end

% ***************************************************************************
% Figure 2
%
% This figure displays the plot of the split-range controller signals, x1,
% x2 or x3 to the control valve q2 when a step change in d1, d2 or d3
% is introduced.
```

```
%
% ***************************************************************************

figure(2)
set(figure(2),'Color','White')
plot(t,q2,'Color',[0 0.7 0],'LineWidth',2)
hold on
plot(t,x1,'b')
hold on
plot(t,x2,'--r')
hold on
plot(t,x3,'k')
hold on
ax = gca;
ax.FontSize = 12;
xlabel('Time,min')
ylabel('Valve opening, %')
ylim([0.3 1.1]);
f3=legend('q_2','x_1','x_2','x_3','Location','Southeast');
set(f3,'FontSize',10)
hold off
grid on

% Saving MATLAB plots to .eps
if d1on == 1
    saveas(gcf,'h1-dis-flows','epsc')
elseif d2on == 1
    saveas(gcf,'h2-dis-flows','epsc')
elseif d3on == 1
    saveas(gcf,'h3-dis-flows','epsc')
end

% ***************************************************************************
% Figure 3
%
% This figure displays the plot of the flow rate through each control
% valve, q1, q2 or q3 when a step change in d1, d2 or d3 is introduced.
%
% ***************************************************************************

figure(3)
set(figure(3),'Color','White')
subplot(211)
plot(t,v1flow.*z1,'b')
hold on
plot(t,v2flow.*z2,'--r')
hold on
plot(t,v3flow.*z3,'k')
hold on
ax = gca;
ax.FontSize = 12;
```

```
xlabel('Time,min')
ylabel('Flow rates m^3/min')
f4 = legend('q_1flow','q_2flow','q_3flow','Location','Southeast');
set(f4,'FontSize',10)
hold off
grid on

subplot(212)
plot(t,z1,'r')
hold on
plot(t,z2,'--k')
hold on
plot(t,z3,'-.b','LineWidth',1)
hold off
xlabel('Time,min')
ylabel('z, hand valve opening, %')
ylim([0.3 1.1]);
legend('z_1','z_2','z_3','Location','Southeast')
ax = gca;
ax.FontSize = 12;
grid on
grid minor

% Saving MATLAB plots to .eps
if d1on == 1
    saveas(gcf,'h1-dis-flows','epsc')
elseif d2on == 1
    saveas(gcf,'h2-dis-flows','epsc')
elseif d3on == 1
    saveas(gcf,'h3-dis-flows','epsc')
end

% ************************************************************************
% Figure 4
%
% Integral wind-up test
% 1) Introduce step flow disturbance to d1 so that the level will deviate
%    from 50%.
% 2) Before Tank 2 level can recover to 50%, introduce a step change to
%    set F1on = 1 (denoted as the variable x3 in the Simulink model
% 3) Leave Fsp = 0 for a period of time
% 4) Reinstate the Fsp to its original setpoint and note the rise in Tank 2
%    ; set F2on = 1
% 5) Apply the anti-windup on the PI controller.
%
% This figure displays the plot of the inventory level of Tank 1
% and 2 when a step change in flow throughput is introduced, before Tank 2
% level can recover to nominal level. The purpose of this Figure is to
% investigate the integral wind-up effect when PI controllers are used
%
% ************************************************************************
```

```
if F1on == 1 || F2on == 1
    figure(4)
    set(figure(4),'Color','White')
    subplot(211)
    plot(t,h1*100/hmax,'r')
    hold on
    plot(t,h2*100/hmax,'k')
    plot([198 198],[48 52],'r--','LineWidth',1)
    plot([400 400],[48 52],'r--','LineWidth',1)
    text(210,51.5,'Integral wind-up','Interpreter','latex')
    hold off
    xlabel('Time,min')
    ylabel('Tank Levels, %')
    f2 = legend('Tank Level 1','Tank Level 2','Location','Southeast');
    set(f2,'FontSize',10)
    ax = gca;
    ax.FontSize = 12;
    grid on
    grid minor

    subplot(212)
    plot(t,x3*300,'r')
    hold on
    plot(t,z1*300,'b')
    plot([198 198],[0 300],'r--','LineWidth',1)
    plot([408 408],[0 300],'r--','LineWidth',1)
    text(210,270,'Integral wind-up','Interpreter','latex')
    xlabel('Time,min')
    ylabel('Flow rates m^3/min')
    legend('F_{sp}','z_1','Location','Northeast')
    ax = gca;
    ax.FontSize = 12;
    grid on
    grid minor

    % Saving MATLAB plots to .eps
    saveas(gcf,'anti-windup','epsc')
end
```

## B.2 TankODE.m

The dynamic mass balance of the tank is listed in the `TankODE.m` file.

```
% ************************************************************************
% function for the dynamic equation of the tank in series
%
%
% Block comment (Ctrl- R); block uncomment (Ctrl-T)
%
% x = state (level); u = inputs(flows); d = disturbances;
%
% p - Tank parameters
%  -                 -         -      -
% |        Area       |       |  6000  | m^2
% |                   |   =   |        |
% | Nominal flow rate |       |   300  | m^3/min
% |                   |       |        |
%  -                 -         -      -
%
% ************************************************************************

function dxdt  = TankODE(p,u,d)

    % tank parameters
    Area = p(1);

    % input, u, valve opening 0 - 100%
    qIn = u(1)*p(2);  % inlet flow
    qOut = u(2)*p(2); % outlet flow

    % bottlenecks, hand valve openings
    zIn = d(1);
    zOut = d(2);

    % differential equation for the tank level
    dxdt = (zIn*qIn - zOut*qOut)/Area;

end
```

# Index