

Selection of Controlled Variables and Robust Setpoints

Marius S. Govatsmark and Sigurd Skogestad *

Department of Chemical Engineering

Norwegian University of Science and Technology (NTNU)

N-7491 Trondheim, Norway

Optimal operation of chemical plants is usually accomplished by first finding the optimal steady-state using the present (nominal) set of disturbances. This is usually implemented by sending constant setpoints for selected variables to the control system. The setpoints are then the nominally optimal values. However, because of disturbances this may result in feasibility problems, which we here try to avoid by adjusting the setpoints ("backoff"). First, we need to avoid infeasibility in the active constraints ("constraint backoff") [1]. Second, we need to adjust the setpoints of the unconstrained controlled variables. This may be done by offline computation of robust setpoints ("optimal backoff") or by online feasibility correction ("flexible backoff"). As a case study we consider a reactor-separator-recycle process. For this process the control structures based on Luybens rule ("fix a flow in every recycle loop") are infeasible if we use the nominal setpoints, but are feasible with reasonable loss if we use robust setpoints.

1 Introduction

This paper is concerned with the *implementation* of an optimal operation policy. We consider a strategy where the optimization layer sends setpoints for the controlled variables to be implemented by the control layer, see Figure 1. There are two classes of problems:

- *Constrained*: At the optimal solution all the optimization degrees of freedom are used to satisfy active constraints for all expected disturbances.
- *Unconstrained or partially constrained* (the focus of this paper): One or more of the optimization degrees of freedom are unconstrained for all or some expected disturbances.

Two important decisions are to be made:

- Decision 1: *Selection of controlled variables (c)*: This is a *structural* decision which is made offline before implementing the control strategy.
- Decision 2: *Selection of setpoints (c_s)* for the controlled variables. This is a *parametric* decision which is done offline or online.

For the constrained variables, active constraint control should be used and the variables lying on the optimally active constraints should be controlled [2]. To remain feasible it may be necessary to back off from the optimal value of the constraints, for example when the constraints are difficult to measure or difficult to control due to poor dynamics. This is here called "(simple) constraint backoff"

*Corresponding author. e-mail: skoge@chemeng.ntnu.no; phone: +47-7359-4154; fax: +47-7359-4080

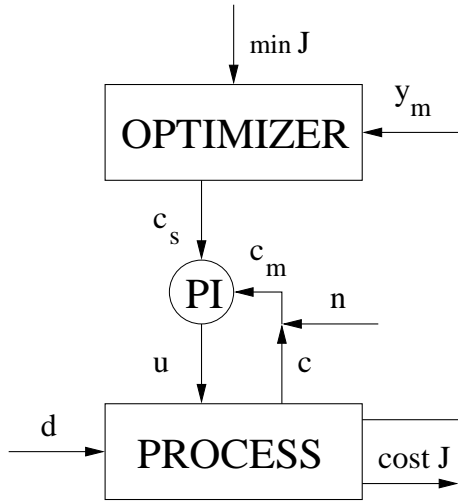


Figure 1: A typical optimization system incorporating local feedback: The process is disturbed (d) and the control system tries to keep the controlled variables (c) at their setpoints (c_s). Steady-state optimization based on process measurements (y_m) is performed at regular intervals to track the optimum by updating the setpoints.

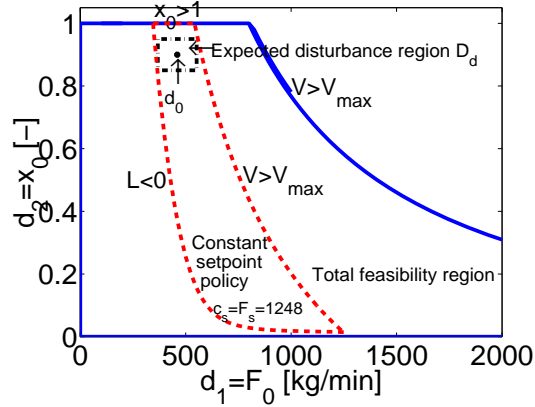


Figure 2: Total feasible region (D_d^{total}) (solid), feasible region for a constant setpoint policy (D_d^c) (dashed) and expected disturbance region (D_d) (dash-dot).

and is thoroughly discussed by Perkins and co-workers[1, 3, 4, 5, 6, 7]. Tracking optimally active constraints which are moving due to disturbances is discussed by Arkun and Stephanopoulos[8]. An exception to the rule of using active constraint control is when the optimally active constraints may move, and in order to avoid reconfiguration we choose to control unconstrained variables with good self-optimizing properties.

For unconstrained or partly unconstrained problems the selection of what to control (Decision 1) is crucial. Morari *et al.*[9] state that the objective is “to find a function c of the process variables which when held constant, leads automatically to the optimal adjustments of the manipulated variables, and with it, the optimal operating conditions”. Skogestad[10] presents a method for selecting "self-optimizing" variables based on minimizing the steady-state loss with constant nominal setpoints. However, in many cases the results are sensitive to the magnitude of the disturbances, and we may get infeasibility for large disturbances and implementation errors. This may result in unstable operation. To avoid infeasibility it may be necessary to back off from the nominal optimum (Decision 2), for example by using robust optimization [11]. In this case the effect of the "uncertainty" (disturbances and implementation errors) is reduced both through the selection of controlled variables and their setpoints.

2 Some definitions

All further considerations are based on a steady-state analysis, unless stated otherwise. All variables are vectors, unless stated otherwise. An element in a vector is denoted with a subscript j . A given operating point is denoted with a subscript i .

2.1 Degrees of freedom

The number of *dynamic degrees of freedom* is equal the number of manipulated variables. The number of *steady-state degrees of freedom* can be found by counting the manipulated variables, subtracting the number of variables that need to be controlled but which have no steady-state effect on the remaining process (e.g. liquid level in a distillation column), and subtracting the number of manipulated variables with no steady-state effect. The number of *degrees of freedom for steady-state optimization* (here denoted u) is equal to the number of steady-state degrees of freedom. The number of *unconstrained steady-state degrees of freedom* is equal the number of steady-state degrees of freedom minus the number of active constraints at the optimum.

2.2 Optimal operation

The optimal operation for a given disturbance (d) can be found by solving the following problem:

$$\begin{aligned} \min_{x,u} J(x, u, d) \\ f(x, u, d) = 0 \\ g(x, u, d) \leq 0 \\ x \in R^{dimx}, u \in R^{dimu}, d \in R^{dimd} \end{aligned} \tag{1}$$

The scalar objective function J describes the cost (quality) of operation, f represents the process model, g is the equality and inequality constraints related to operation, u is the independent variables (*manipulated variables* or *inputs*) we can affect, d is the independent variables (*disturbances*) we cannot affect, and x is the internal variables (*states*). The inequality constraints typically include upper and lower bounds on the input and output variables. In addition to the external disturbances d , we must also during actual implementation consider the "implementation" disturbances (implementation errors d_c and d_g , see later), but these are not included in the above "open-loop" optimization problem.

2.3 Constraints

We distinguish between transient and steady-state constraints. This is similar to batch processes where it is common to distinguish between path and endpoint/terminal constraints, see e.g. Loeblein *et al.*[12]. *Steady-state constraints* may be violated during transients, but not at steady-state or in average, e.g. this could be a product purity constraint. *Transient constraints* must be violated neither in transients nor at steady-state, e.g. this could be a maximum pressure constraint.

We also distinguish between *active constraints* and *inactive constraints*. For a given operating point i an active constraint j satisfies $g_{j,i} = 0$, whereas an inactive constraint j satisfies $g_{j,i} < 0$. Note here that "optimally active constraints" are usually called simply "active constraints".

In most cases we identify a single measured (or estimated) variable y related to each constraint and write (depending on whether g corresponds to a minimum or maximum constraint):

$$g = y - y_{max} \tag{2}$$

or

$$g = y_{min} - y \tag{3}$$

2.4 Controlled variables

We here define the *controlled variables* c as the variables that are specified (kept constant by the control system at steady-state). The controlled variables may consist of manipulated variables, measurements or combinations of measurements and manipulated variables. The number of selected controlled variables is here assumed equal to the number of steady-state degrees of freedom.

We distinguish between *constrained controlled variables* and *unconstrained controlled variables*. A constrained controlled variable is kept constant at an active constraint by the control system.

2.5 Uncertainties

Uncertainties in the operation are related to external disturbances (d) and implementation errors (d_c, d_g). Implementation errors may be related to poor control or to errors in the measurements of the controlled variables and constraints.

2.5.1 Disturbances

Disturbances d are independent variables that we cannot affect and that are not related to the control system implementation. These variables may also include uncertainties in the model parameters.

2.5.2 Implementation errors

The implementation error in the controlled variable d_c (sometimes denoted n) is the difference between the actual value of the controlled variable c and its setpoint c_s :

$$d_c = c - c_s \quad (4)$$

The implementation error d_c may be written as the sum of the measurement error ($c - c_m$) and the control error ($c_m - c_s$), see also Figure 1. With integral action in the controller, we may consider disregarding the control error. However, even with integral action we may not be able to reach the steady-state within the time period of interest, and we then need to include the control error. For controlled variables related to transient constraints we must also include the worst-case dynamic control error (see the work of Perkins *et al.*[1]).

Implementation errors in the constraints d_g should be included for the constraints that are measured (g_m) and used by the control system. *The implementation error in the constraint* d_g is the difference between the actual value of the constraint g and the measured or estimated value of the constraint g_m :

$$d_g = g - g_m \quad (5)$$

If we have a single measured (or estimated) variable that identifies the constraint (see equation 2 or 3), then:

$$d_{g,j} = y_j - y_{m,j} \quad (6)$$

For a constrained controlled variable the implementation error in the constraint is equal the implementation error in the controlled variable ($d_{c,j}=d_{g,j}$ for a maximum constraint and $d_{c,j}=-d_{g,j}$ for a minimum constraint).

2.5.3 Expected disturbance and implementation error region

A disturbance d can be expressed as the sum of its nominal value d_0 and some variation $\Delta d \in D_d$:

$$d = d_0 + \Delta d; \quad \Delta d \in D_d \quad (7)$$

Similarly, an implementation error in the controlled variables d_c can be expressed as the sum of its nominal implementation error $d_{c,0}$ (usually zero) and some variation $\Delta d_c \in D_c$:

$$d_c = d_{c,0} + \Delta d_c; \quad \Delta d_c \in D_c \quad (8)$$

and an implementation error in the constraints d_g can be expressed as the sum of its nominal implementation error $d_{g,0}$ (usually zero) and some variation $\Delta d_g \in D_g$:

$$d_g = d_{g,0} + \Delta d_g; \quad \Delta d_g \in D_g \quad (9)$$

The *nominal point* is given by the nominal disturbance (d_0) and the nominal implementation errors ($d_{c,0}, d_{g,0}$).

The *expected disturbance and implementation error region* $\tilde{D}_{deg} = (\{D_d, D_c, D_g\})$ consists of these expected variations. Note that "the expected disturbance and implementation error region" is here often simply called "the expected disturbances and implementation errors". The magnitude of \tilde{D}_{deg} depends on the considered period. It is largest when the process overall lifetime is considered. Use of online optimization reduces the period to considered and thereby \tilde{D}_{deg} . Note that when using online optimization the nominal point ($d_0, d_{c,0}, d_{g,0}$) will change.

The *maximum expected implementation error for a controlled variable j* is:

$$d_{c,max,j} = \max_{\Delta d_{c,j} \in D_{c,j}} |\Delta d_{c,j}| + d_{c,0,j} \quad (10)$$

and the *maximum expected implementation error for a constraint j* is:

$$d_{g,max,j} = \max_{\Delta d_{g,j} \in D_{g,j}} |\Delta d_{g,j}| + d_{g,0,j} \quad (11)$$

The *maximum expected implementation error in the constrained controlled variables* $d_{c,max}^*$ is equal to the maximum expected implementation error in each constrained controlled variable (see equation 10) and equal to zero for each unconstrained controlled variable. The *maximum expected implementation error in the active constraints* $d_{g,max}^*$ is similarly defined for active and inactive constraints.

2.6 Feasibility

2.6.1 Feasibility

For a set of disturbance variations $\Delta d \in D_d$, operation is feasible if there exist inputs u (and corresponding states x) such that the following constraints are fulfilled for all disturbances:

$$\begin{aligned} f(x, u, d) &= 0 \\ g(x, u, d) &\leq 0 \\ d &= d_0 + \Delta d; \\ &\forall \Delta d \in D_d \end{aligned} \quad (12)$$

The set D_d includes the nominal disturbance ($\Delta d = 0$).

The total feasibility region (D_d^{total}) is the disturbance region where at least one u fulfills the constraints f and g . Note that the feasibility region is defined as a deviation region that refers to a given nominal disturbance d_0 . D_d^{total} is larger than D_d ($D_d \subseteq D_d^{total}$) if we have feasibility.

2.6.2 Feasibility of a specific constant setpoint policy

For a given set of controlled variables c with setpoints c_s , a constant setpoint policy is feasible if the following constraints are fulfilled for all expected disturbances and implementation errors:

$$\begin{aligned}
 f(x, u, d) &= 0 \\
 g(x, u, d) &\leq 0 \\
 c(x, u, d) &= c_s + d_c \\
 d &= d_0 + \Delta d; \\
 d_c &= d_{c,0} + \Delta d_c; \\
 \forall \Delta d \in D_d, \Delta d_c \in D_c
 \end{aligned} \tag{13}$$

Note that the implementation error in the active constraints are included in d_c , so we do not need to explicitly include implementation errors in the constraints (d_g).

The specific constant setpoint policy feasibility region $\tilde{D}_{dc}^c = \{D_d^c, D_c^c\}$ is the disturbance and implementation error region where the constraints f and g are fulfilled for a specific constant setpoint policy. A specific constant setpoint policy is feasible when the expected disturbance and implementation error region (\tilde{D}_{dc}) is a subset of \tilde{D}_{dc}^c , i.e. $\tilde{D}_{dc} \subseteq \tilde{D}_{dc}^c$. Note that using online optimization reduces \tilde{D}_{dc} and thereby increases the possibility for achieving feasibility of a specific constant setpoint policy. The specific constant setpoint policy feasibility region for a given implementation error d_c ($\{\tilde{D}_{dc}^c | \Delta d_c\}$) is always a subset of the total feasibility region D_d^{total} for all expected implementation errors d_c , i.e. $\{\tilde{D}_{dc}^c | \Delta d_c\} \subseteq D_d^{total} \quad \forall \Delta d_c$. This is illustrated in Figure 2 for a specific example.

2.7 Active constraint control

We normally use *active constraint control* [2]. This implies that if a constraint becomes *optimally active*, we select the corresponding measurement or estimate y_j of the constraint as a controlled variable ($c_j = y_j$). For steady-state constraints we must include the measurement (or estimation) error for the constraint. For transient constraints we must also include the control error (which has been studied in detail by Perkins and co-workers). To ensure feasibility such that the constraints are never violated, the following setpoints for the active constraint controlled variables are used for a minimum constraint:

$$c_{s,j} = y_{min,j} + d_{g,max,j} \tag{14}$$

or for a maximum constraint:

$$c_{s,j} = y_{max,j} - d_{g,max,j} \tag{15}$$

2.8 Loss

For a given choice of controlled variables, the *loss* for a given disturbance (d), implementation error (d_c) and setpoint c_s is [10]:

$$L(d, d_c, c_s) = J_c(c_s + d_c, d) - J_{opt}(d) \quad (16)$$

where $J_c(c_s + d_c, d) = J(x(c_s + d_c, d), u(c_s + d_c, d), d)$. Figures 3 and 4 show the loss as function of the disturbance and implementation error for different sets of controlled variables for a specific example. Note that the losses depend strongly on the selected controlled variables.

The *percentage loss* $L(\%)$ for a given disturbance (d) and implementation error (d_c) with constant setpoint c_s is:

$$L(d, d_c, c_s)(\%) = \frac{J_c(c_s + d_c, d) - J_{opt}(d)}{J_{opt}(d)} \cdot 100\% \quad (17)$$

For a set of disturbances and implementation errors ("operating points") i , the *maximum loss* (in percent) is defined as:

$$L_{max} = \max_i \frac{J_c(c_s + d_{c,i}, d_i) - J_{opt}(d_i)}{J_{opt}(d_i)} \cdot 100\% \quad (18)$$

and the *average loss* or *weighted loss* (in percent) is defined as:

$$L_w = \frac{J_{c,w} - J_{opt,w}}{J_{opt,w}} \cdot 100\% \quad (19)$$

where $J_{c,w} = \sum_i w_i J_c(c_s + d_{c,i}, d_i)$ and $J_{opt,w} = \sum_i w_i J_{opt}(d_i)$.

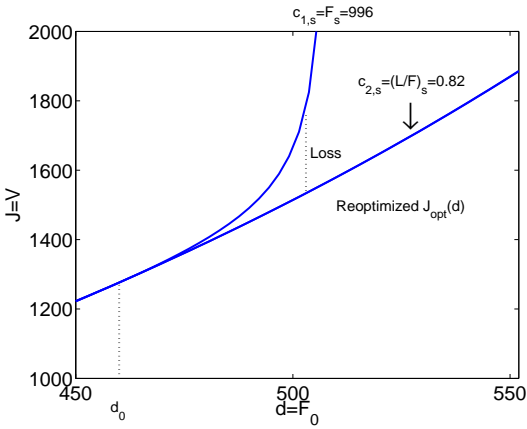


Figure 3: Cost as a function of the disturbance with (i) reoptimized setpoint (lower curve) and (ii) two alternative constant setpoint policies (c_1 and c_2). The loss is negligible with c_2 as a controlled variable.

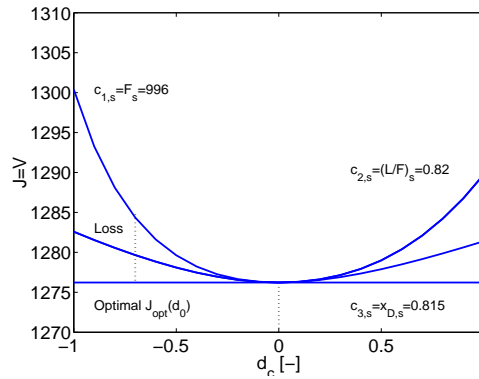


Figure 4: Cost as a function of the implementation error. The loss is negligible with c_3 as a controlled variable.

2.9 Self-optimizing control

A set of controlled variables has self-optimizing properties if a constant setpoint policy yields an acceptable loss L for the expected variation in disturbances and implementation errors [10].

2.10 Backoff

Backoff is a setpoint adjustment, which is primarily used to avoid infeasibility; also see Figure 5. More precisely, we define *the backoff* (b) as *the difference between the actual setpoints and some reference values for the setpoints*:

$$b = c_s - c_{s,ref} \quad (20)$$

3 Optimization problems

3.1 Ideal optimization

For a given disturbance (d) the optimal operation (u_{opt} with corresponding x_{opt}) is found by solving the following problem:

$$\begin{aligned} \min_u J(x, u, d) \\ f(x, u, d) = 0 \\ g(x, u, d) \leq 0 \end{aligned} \quad (21)$$

The corresponding solution is denoted $u_{opt}(d)$ and $x_{opt}(d)$ and we have:

$$J_{opt}(d) = J(x_{opt}(d), u_{opt}(d), d) \quad (22)$$

For the nominal case with $d = d_0$, the corresponding optimal setpoints are:

$$c_{s,opt}(d_0) = c(x_{opt}(d_0), u_{opt}(d_0), d_0) \quad (23)$$

If we try to implement these setpoints, we will get infeasibility if there are implementation errors (uncertainties) in the (optimally) active constraints. To avoid infeasibility we need to include uncertainty in the optimization as is discussed in the following. Note that we write the minimization problem as $\min_u J(x, u, d)$ instead of the equivalent problem $\min_{x,u} J(u, x, d)$ to show more clearly that u represents the degrees of freedom. Similar notation is used in equation 24, 27, 28 and 29.

3.2 Optimization with constraint backoff ("adjusted")

The first step in including uncertainties (disturbances and implementation errors) is to introduce constraint backoff ($d_{g,max}^*$ to avoid infeasibility caused by implementation error in the constraints that are active at the optimum. The "adjusted" optimal operation ($u_{opt}^*(d)$ with corresponding $x_{opt}^*(d)$) for a given disturbance (d) is found by solving the following problem:

$$\begin{aligned} \min_u J(x, u, d) \\ f(x, u, d) = 0 \\ g(x, u, d) + d_{g,max}^* \leq 0 \end{aligned} \quad (24)$$

The corresponding solution is denoted $x_{opt}^*(d)$ and $u_{opt}^*(d)$ and we have:

$$J_{opt}^*(d) = J(x_{opt}^*(d), u_{opt}^*(d), d) \quad (25)$$

For example, assume that we have maximum pressure constraint $p \leq 10$ bar (i.e. $g = p - 10$) and that the implementation (measurement) error for pressure is ± 0.2 bar ($d_{g,max}^* = 0.2$). In order to

guarantee that the actual pressure remains less than 10 bar (feasibility) we must then back off and require $p \leq 9.8$ bar where p is the model pressure. Note that the implementation error may have been accounted for when formulating the constraints, and should then not be counted twice. Also, for input constraints the implementation error is often zero ($d_{g,max}^* = 0$). For example, we are often able to implement exactly the requirement of a closed valve (zero flow) or a fully open valve (maximum flow). The constraint backoff results in an unavoidable additional loss at the nominal operating point.

For the nominal case with $d = d_0$, the corresponding "adjusted" nominally optimal setpoints for the controlled variables can be computed by:

$$c_{s,0} = c_{opt}^*(d_0) = c(x_{opt}^*(d_0), u_{opt}^*(d_0), d_0) \quad (26)$$

These are hereafter called the *nominal setpoints*.

3.3 Robust optimization

The above nominal optimization problem (eq. 24 with $d = d_0$) resulting in $c_{s,0}$ in eq. 26 is relatively easy to solve, and is what we normally would recommend for computing the setpoints for practical implementations. However, these nominal setpoints are not generally optimal. The "truly optimal" constant setpoints may be obtained by including all expected uncertainties (all expected disturbances (d) and implementation errors (d_c)) and evaluate the appropriate average cost.

Let us first consider the simpler problem without implementation errors. This can be regarded as a stochastic optimization problem [13]¹:

$$\begin{aligned} & \text{"min"}_u \text{" } J(x, u, d) \\ & f(x, u, d) = 0 \\ & g(x, u, d) \leq 0 \\ & d = d_0 + \Delta d \end{aligned} \quad (27)$$

where Δd is a random vector varying over the set D_d .

We here extend this stochastic optimization problem to consider the constant setpoint problem and also include the uncertainty related to implementing the optimal solution. Instead of stochastic optimization, we here consider the very similar deterministic problem where we minimize some "mean" weighted cost while fulfilling the constraints over all expected disturbances (d) and implementation errors (d_c). The problem is infinite dimensional, but we here simplify it by considering a discrete number of operating points i . The corresponding *robust optimization problem* was introduced by Glemmestad *et al.*[11] to find robust setpoints $c_{s,robust}$:

$$\begin{aligned} & \min_{c_s} \sum_i w_i J(x_i, u_i, d_i) \\ & f(x_i, u_i, d_i) = 0 \\ & g(x_i, u_i, d_i) \leq 0 \end{aligned} \quad (28)$$

¹Kall and Wallace[13] focus on stochastic optimization problems with recourse to achieve feasibility when the uncertain parameters are known. The recourse problem is not considered here. Feasibility is required for all expected uncertainties (disturbances and implementation errors), while "rare" uncertainties (disturbances or implementation errors) are assumed to be handled by the safety system.

$$\begin{aligned}
c(x_i, u_i, d_i) &= c_s + d_{c,i} \\
d_i &= d_0 + \Delta d_i \\
d_{c,i} &= d_{c,0} + \Delta d_{c,i} \\
\Delta d_i &\in D_d, \Delta d_{c,i} \in D_c
\end{aligned}$$

Implementation errors d_g on the active constraints are included in the variables d_c and the constraint backoff is "automatically" included when we obtain c_s . Note also that we need to solve the optimization problem for each candidate set of controlled variables.

3.4 Online feasibility correction

A constant setpoint policy may not be feasible, that is, there may not exist any solution to the robust optimization problem. Alternatively, a constant setpoint policy may be too conservative. Also, the computation load for the robust setpoints may be too heavy. An alternative strategy may be to adjust the setpoints online. In practice this may be handled by the steady-state optimization layer in model predictive control (MPC) [14], so implementation is straightforward if we have MPC software in place. The main point here is not to present a new method or algorithm, but to evaluate the resulting operation.

For our "offline" analysis of this scheme we use the nonlinear model. For the analysis we also need to include the implementation errors. We may use *hard prioritizing* among the controlled variables. For example, variables at active constraints should be kept constant, if possible. The set of controlled variables c is then divided in two subsets c_I and c_{II} :

- c_I : Controlled variables for which no backoff is allowed, if possible (variables with high priority). Often these are selected as variables at active constraints.
- c_{II} : Controlled variables for which backoff is allowed (variables with low priority).

The resulting optimization problem becomes:

$$\begin{aligned}
\min_{c_{II,flex}} & (c_{II,flex} - c_{II,s})^T Q_{II} (c_{II,flex} - c_{II,s}) \\
& f(x, u, d) = 0 \\
& g(x, u, d) + d_{g,max} - d_g \leq 0 \\
& c(x, u, d) = c_{flex} + d_c \\
& c_{I,flex} = c_{I,s}
\end{aligned} \tag{29}$$

and the resulting cost is $J_{flex}(c_{flex} + d_c, d, d_g)$. In practice, one may start with a small set c_{II} and increase it if feasibility is not achieved. The priority within the set c_{II} can be adjusted through the weight Q_{II} . If there is no feasible solution to equation 29, we need to remove constraints. This is not considered here. Since measurements of all constraints (including uncontrolled constraints) are used in the implementation, implementation errors in the constraints (d_g) need to be explicitly included. The online adjusted setpoints, labeled *flexible setpoints* c_{flex} , achieve feasible operation, but the resulting loss may be large. Note that using equation 29 with nonlinear model to evaluate the steady-state performance for a linear MPC is usually acceptable since there are online measurements of the controlled variables and constraints, and through feedback the model or disturbances are updated such that the computed controlled variables are equal to the measured values and the computed constraints are equal to the measured constraints.

Figure 5 shows the optimal backoff and flexible backoff for a disturbance. With the setpoint fixed at the nominal optimum ($c_s = c_{s,0}$) we get infeasibility with disturbance d_1 . With optimal backoff ($c_{s,robust} = c_{s,0} + b_{opt}$) we get feasibility and close to optimal operation in both cases (d_0, d_1). With flexible backoff the setpoint is $c_{s,flex} = c_{s,0}$ at the nominal point (d_0) and changes to $c_{s,flex} = c_{s,0} + b_{flex}$ with disturbance d_1 . This gives feasibility, but far from optimal operation. If the average cost is primarily determined by operation at or close to the nominal point, the use of flexible backoff may give a small loss. The backoff is then just done for ensuring feasibility in some "extreme" points. However, flexible backoff adjusts the setpoints without considering the actual cost function, so the loss may in some cases be very large as illustrated in Figure 5. The selection of controlled variables and corresponding setpoints with good self-optimizing properties therefore remains important, also with online feasibility correction.

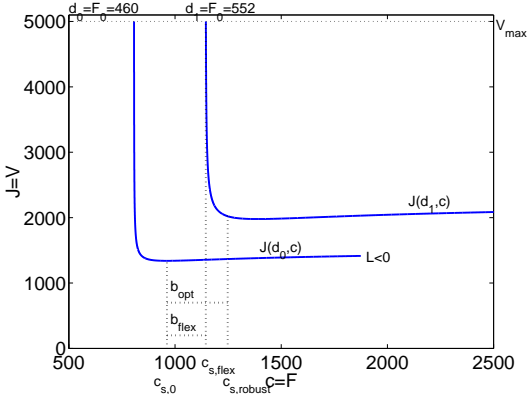


Figure 5: Cost (J) as function of the controlled variable (c) at nominal point (d_0 , lower curve) and with disturbance (d_1 , upper curve).

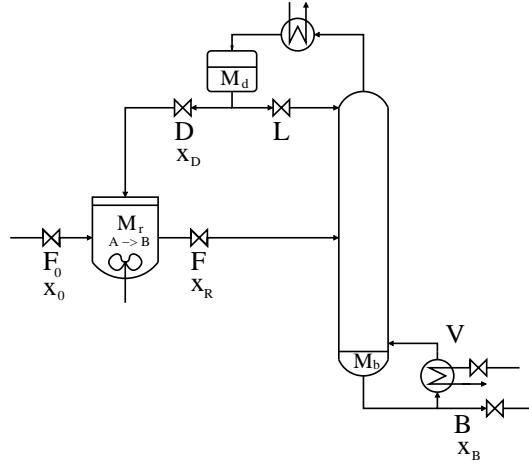


Figure 6: Reactor/separators process with liquid recycle

4 Example: Reactor, separator and recycle process

We now apply the above ideas to a case study. To select the controlled variables and their setpoints we use an extension of the method of Skogestad[10], consisting of the following steps:

1. *Initial system analysis:*
Identify the number of degrees of freedom, define objective function and constraints, identify main disturbances and measurements, optimize at nominal and expected disturbances, see equation 21.
2. *Identify sets of candidate controlled variables:*
Use active constraint control, eliminate variables with no steady-state effect, eliminate variables with large losses by using short-cut loss evaluation, eliminate variables based on process insight.
3. *Evaluate the loss for different sets of controlled variables, using:*
 - (a) Constant nominal setpoints $c_{s,0}$, see equation 26
 - (b) Constant robust setpoints $c_{s,robust}$, see equation 28

(c) Nominal setpoints with online feasibility correction c_{flex} (flexible setpoints), see equation 29.

4. *Final evaluation and selection of control structure:*

Stabilization, controllability analysis, selection of control configuration and simulation of proposed control structures.

The process consists of a reactor, a distillation column and a liquid recycle [15] and is shown in Figure 6. We use the model parameters from Wu and Yu[16]. There is no inert in the feed, so no purge is required. Larsson *et al.*[17] consider the control structure selection with emphasis on identifying promising sets of controlled variables when using constant nominal setpoints. We here consider a given feedrate (case I in Larsson *et al.*[17]).

4.1 Initial system analysis

The process has five manipulated variables (valves):

$$u_{dyn}^T = [L \ V \ B \ D \ F]$$

However, for stabilization we need to control two variables (the reboiler holdup (M_b) and condenser holdup (M_d)) which have no steady-state effect. We are then left with three degrees of freedom at steady-state. These may be selected as the reactor holdup (M_r), product composition (x_B) and recycle composition (x_D), i.e. $u^T = [M_r \ x_B \ x_D]$. The economic objective is to maximize the profit (the value of the products minus the cost of the utilities and raw materials). Since F_0 is given and there is no purge, it follows that B is given. Furthermore, L depends directly on V , so the economic objective can be simplified to minimize the boilup ($J = V$). The reactor volume (M_r) and boilup flowrate (V) are constrained and there is a product purity specification (x_B): The reactor volume constraint ($0 \leq M_r \leq 2800$) and boilup flowrate constraint ($V \leq V_{max} = 5000$) are transient constraints whereas the product specification constraint ($x_B \leq 0.0105$) is a steady-state constraint. The main disturbances are feedrate (F_0) and feed composition (x_0):

$$d^T = [F_0 \ x_0] = [460 \pm 92 \text{ kmol/h} \ 0.90 \pm 0.05 \text{ molA/mol}]$$

We consider the following 20 candidate controlled variables (9 manipulated variables and measurements and 11 flow ratios):

$$c^T = [L \ V \ D \ B \ F \ M_r \ x_r \ x_B \ x_D \ \frac{L}{F} \ \frac{V}{F} \ \frac{B}{F} \ \frac{D}{F} \ \frac{V}{L} \ \frac{B}{L} \ \frac{D}{L} \ \frac{B}{V} \ \frac{D}{V} \ \frac{B}{D} \ \frac{F}{F_0}]$$

Of these we want to control three. We will in this paper *not* consider the use of other variable combinations than the ratios listed here. The implementation errors are initially assumed to be $\pm 10\%$ for the flowrates, $\pm 0.25\%$ (absolute) for the compositions and $\pm 1\%$ for the holdups. The implementation error for the reactor holdup is the sum of expected measurement error and expected (dynamic) control error since the reactor holdup constraint is transient. The implementation error for the product composition is in practice the sum of expected measurement error and steady-state control error. Steady-state optimizations for the nominal point and the corner-points with the expected disturbance variation², see equation 21, show that the product composition (x_B) and the reactor holdup (M_r) are always at their constraints.

²We expect that only one disturbance d_j is perturbed from the nominal value at the same time

4.2 Identify sets of candidate controlled variables

There are 20 candidate controlled variables and three steady-state degrees of freedom. This gives $(20 \cdot 19 \cdot 18)/(3 \cdot 2 \cdot 1) = 1140$ alternative sets of three controlled variables. As a first step we want to reduce the number of candidate sets. We have already eliminated the condenser (M_D) and the reboiler holdup (M_B) which have no steady-state effect. In addition in order to nominally optimize the operation we choose to control the two (optimally) active constraints (c_1 =product composition and c_2 =reactor holdup). We are then left with 18 candidate controlled variables and 1 steady-state degree of freedom, which give 18 possible sets.

Initial screening is performed by considering the steady-state gain ($\underline{\sigma}(G(0)) = |G(0)|$) from the remaining unconstrained degree of freedom to the alternative controlled variables (c), which according to the singular value rule [18] should be maximized in order to achieve self-optimizing control. The gain matrix $G(0)$ is obtained with the active constraints kept constant. The candidate controlled variables are scaled with respect to variation in optimal values (absolute value of maximum deviation in the optimal value from the nominally optimal value, $\Delta c_i = \max_D |c_{opt,i}(d) - c_{opt,i}(d_0)|$) and implementation errors. x_D and L/F are the most promising controlled variables. We note that four candidate variables have a zero gain; B/V , V , x_r and B . This is easy to explain [17].

4.3 Loss evaluation

For the remaining 14 alternative sets we have evaluated the economic loss imposed by using constant setpoints instead of reoptimization. We have also evaluated seven alternatives, discussed in literature, which are not nominally optimal since they do not control the reactor holdup at its constraint. These include the Balanced Structure (BS) with control of x_B , x_r and x_D (Balanced Structure I in Larsson *et al.*[17]) and the Luyben Structure (LS) with control of x_B , F and x_D . Note that our definitions of Luyben structure (LS) and Balanced structure (BS) are not equal the definition of Luyben structure (free F_0) and proposed Balanced structures ($x_B, F/F_0, x_R$ or $x_B, F/F_0, x_D$) in Wu and Yu[16]. However, LS and BS used in this paper demonstrates the effect of controlling a flowrate in the recycle (Luyben) and controlling compositions instead of the reactor holdup (Balanced). The nominal point and corner points for expected disturbances and implementation errors are included as operating points with equal weights (w_i). We assume that only one disturbance d_j or implementation error $d_{c,j}$ is perturbed from its nominal value at the same time.

We select to study the following alternatives in more detail:

- x_D or L/F with good self-optimizing properties according to Larsson *et al.*[17] also the two best in initial screening
- F or D which follows Luybens rule by "fixing a flow in every recycle loop" [19] (but which seem poor from initial screening
- LS or BS with no control of reactor holdup (not nominally active).

Figures 9 and 10 show the loss as function of the disturbances and implementation errors with constant nominal setpoints, constant robust setpoints and flexible setpoints for these alternatives. We also compare with adjusted optimal operation with constraint backoff ("adjusted optimal").

Table 1: Average and maximum percentage loss (L_w, L_{max}) with constant nominal setpoints, constant robust setpoints and flexible setpoints for alternative sets of controlled variables.

$c_{1,c2,c3}$	Nominal setpoints			Robust setpoints			Flexible setpoints				
	$c_{3,s}$	L_w (%)	L_{max} (%)	$b_{3,opt}$	L_w (%)	L_{max} (%)	$b_{1,flex}^{max}$ (%)	$b_{2,flex}^{max}$	$b_{3,flex}^{max}$	L_w (%)	L_{max} (%)
Adjusted opt.	$c_{opt}^*(d)$	5.16	11.03	—	5.16	11.03	—	—	—	4.97	11.03
x_B, M_r, x_D	0.825	5.22	11.04	0.005	5.21	11.03	0	0	0	5.22	11.04
$x_B, M_r, L/F$	0.871	5.39	11.24	-0.021	5.35	11.34	0	0	0	5.39	11.24
$x_B, M_r, D/L$	0.600	5.40	11.15	0.047	5.34	11.35	0	0	0	5.40	11.15
$x_B, M_r, D/V$	0.375	5.60	11.15	0.026	5.49	11.46	0	0	0	5.60	11.15
$x_B, M_r, V/F$	1.392	6.05	11.37	-0.037	5.93	11.67	0	0	0	6.05	11.37
$x_B, M_r, B/L$	0.549	6.31	13.70	0.054	6.05	12.60	0	0	0	6.31	13.70
x_B, M_r, L	837.4	6.68	22.95	-64	6.46	15.87	0	0	0	6.68	22.95
$x_B, M_r, V/L$	1.600	8.64	41.31	0.206	5.89	12.19	0	0	0	8.64	41.31
$x_B, M_r, B/D$	0.916	11.2	47.77	-0.140	6.00	11.68	0	0	0	11.2	47.77
$x_B, M_r, F/F_0$	2.091	inf	inf	0.289	6.22	12.15	0	0	0.1280	36.82	291.78
$x_B, M_r, B/F$	0.478	inf	inf	-0.056	6.36	12.08	0	0	0.0145	36.87	291.78
$x_B, M_r, D/F$	0.522	inf	inf	0.061	6.50	12.24	0	0	0.0242	36.89	291.78
x_B, M_r, D	502.0	inf	inf	191	6.79	12.82	0	0	91	26.63	165.24
x_B, M_r, F	962.0	inf	inf	286	7.51	13.90	0	0	183	26.90	165.24
$x_B, F/F_0, V/B$	—	inf	inf	—	25.87	54.38	0	0.4341	0.6718	9.40	20.28
$x_B, F/F_0, x_D$	—	inf	inf	—	25.91	54.38	0	0.5971	0.0001	8.24	24.36
$x_B, x_D, x_R(BS)$	—	inf	inf	—	26.08	54.38	0	0.0868	≈ 0	6.36	16.24
$x_B, M_r, F, L/D$	—	inf	inf	—	26.11	54.38	0	0.7455	≈ 0	7.36	27.08
$x_B, F/F_0, L/D$	—	inf	inf	—	33.13	63.59	0	0.4646	0.0001	10.06	26.59
$x_B, F, x_D(LS)$	—	inf	inf	—	43.10	94.37	0	183	0.1735	30.84	165.24
$V/B, F/F_0, x_D$	—	inf	inf	—	45.74	78.75	0.6719	0.4870	0.0321	8.95	28.28

inf: infeasible operation

Constrained variables: $c_{1,s} = x_{B,s} = 0.008$ and $c_{2,s} = M_{r,s} = 2772$

Adjusted opt.: Adjusted optimization with constraint backoff, see equation 24.

$$b_{j,flex}^{max} = \max_i |b_{j,flex,i}|.$$

4.3.1 (a) Loss evaluation with nominal setpoints

The average and maximum percentage losses with constant *nominal* setpoints based on constraint backoff (see equation 26) are listed in Table 1. Considering the average loss gives the following ranking:

- Control of x_D (Figure 7) is best, closely followed by L/F (Figure 8), D/L and D/V , see also Larsson *et al.*[17]
- Control of F or D , which follows Luybens rule gives infeasibility
- None of the seven alternatives without control of reactor holdup ("below the line" in Table 1) yield feasible operation for all disturbances.

From Figures 9 and 10 we find:

- The implementation error in the product composition ($d_{c,1}$) gives a significant loss for all alternatives. This is because over-purifying the product increases the boilup rate (energy). Reducing this implementation error ($d_{c,1}$) will give a significant reduction in loss, but will not alter the ranking.
- Control of F or D (Luybens rule) gives large losses for disturbances in the feedrate (F_0). We even get infeasibility with feedrates (F_0) larger than 485 for F and 505 for D .
- Structures LS (Luyben structure) or BS (Balanced structure) give large losses with small F_0 and infeasibility with large F_0 .

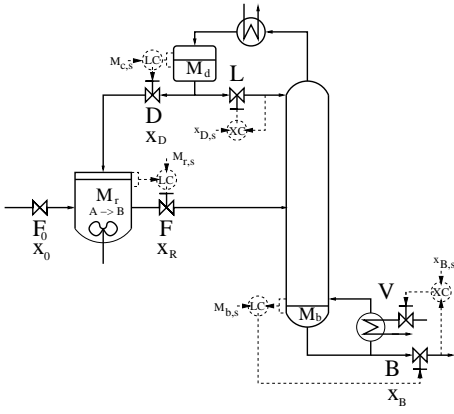


Figure 7: Alternative x_D

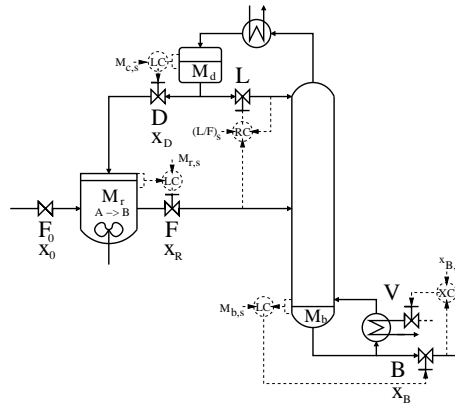


Figure 8: Alternative L/F

4.3.2 (b) Loss evaluation with robust setpoints

Use of constant nominal setpoints may exclude controlled variables that are workable. We therefore consider the use of constant robust setpoints, see equation 28. The average and maximum percentage losses and the backoff in the unconstrained variable with constant *robust* setpoints, are also shown in Table 1:

- In this case there is no backoff in the constrained controlled variables.
- For alternatives that were feasible with constant nominal setpoints, there are only minor changes.
- All alternatives are now feasible, but the loss may be large, especially for the cases "below the line" in Table 1, where we do not control reactor holdup. However, control of F or D (Luybens rule) which was infeasible with constant nominal setpoints, are now feasible structures and give acceptable losses.

These findings are confirmed by Figures 9 and 10. Note that the backoff to achieve feasibility for alternatives which are infeasible with nominal setpoints (e.g. F , D , LS and BS), results in a significant extra loss at the nominal point, but nevertheless the weighted average loss may be acceptable.

4.3.3 (c) Loss evaluation with flexible setpoints

We will now consider the use of online feasibility correction based on nominal setpoints. We assume hard ranking of the controlled variables (see equation 29). The constrained controlled variables have high priority and the unconstrained controlled variables have low priority, for example $c_I = [x_B \ M_r]^T$ and $c_{II} = [F]$ for alternative F . The weight matrix Q_{II} is selected as diagonal matrices with one over the expected implementation errors on the diagonal ($1/d_{c,i}$). No implementation error is included for constraints that are not controlled (e.g. M_r for LS and BS). From Table 1 we see:

- There is no backoff ($b_{flex} = 0$) for controlled variable alternatives that are feasible with nominal setpoints.
- All considered alternatives are feasible with flexible backoff, but the operation is far from optimal in some cases (much worse than with optimal backoff).

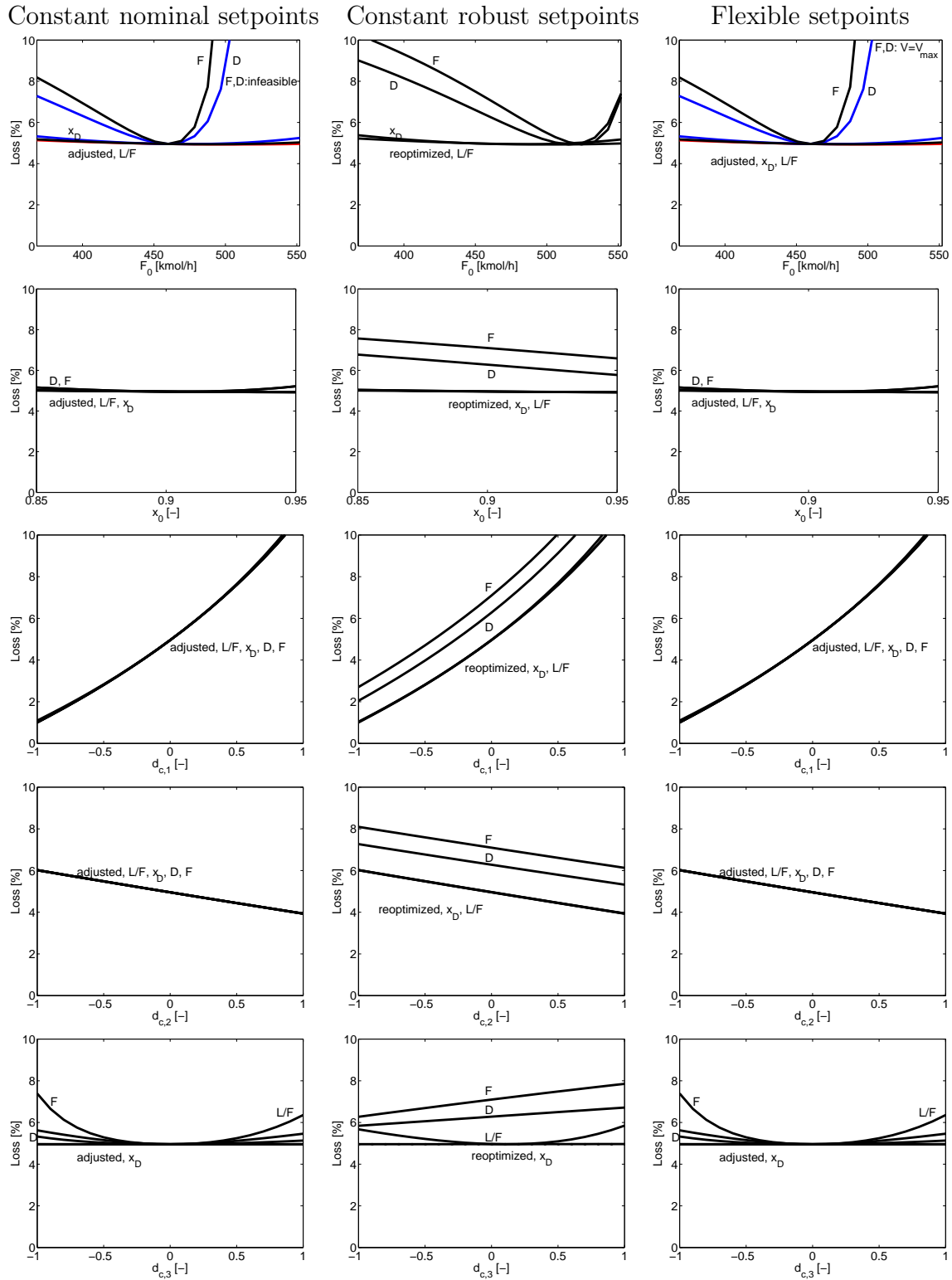


Figure 9: Loss as a function of disturbances (F_0 , x_0) and implementation errors ($d_{c,1}, d_{c,2}, d_{c,3}$) for the "good" controlled variables with (1) constant nominal setpoints (left column), (2) constant robust setpoints (middle column) and (3) flexible setpoints (right column).

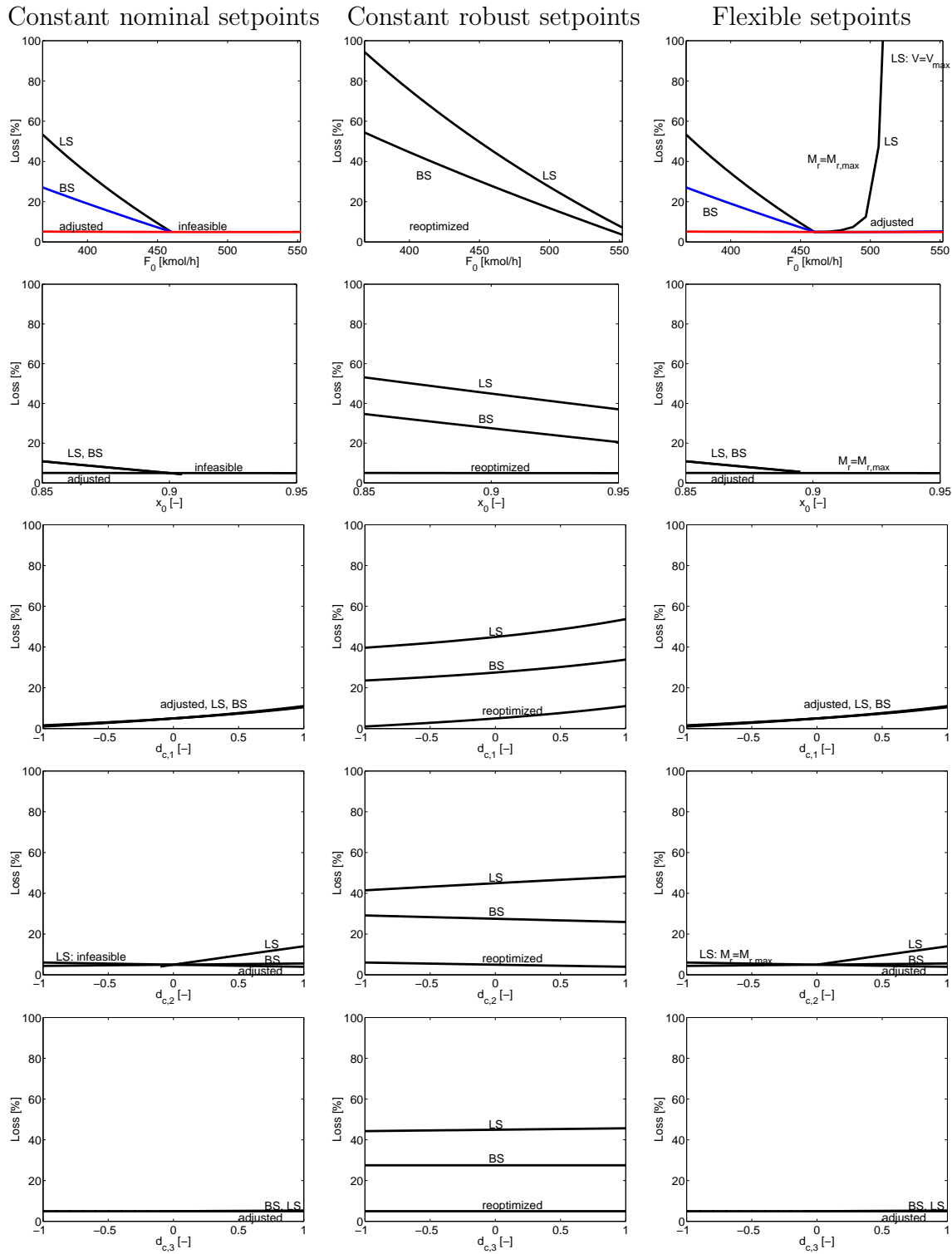


Figure 10: Loss as function of disturbances (F_0 , x_0) and implementation errors ($d_{c,1}$, $d_{c,2}$, $d_{c,3}$) with "poor" controlled variables with (1) constant nominal setpoints (left column), (2) constant robust setpoints (middle column) and (3) flexible setpoints (right column).

- There are cases where flexible backoff is better than "optimal" backoff. This is not surprising since optimal backoff uses constant setpoints.

These findings are confirmed by considering Figures 9 and 10:

- When controlling F or D with flexible backoff we avoid infeasibility when the column reaches maximum boilup rate at large feedrates ($F_0 > 485$ for structure F and $F_0 > 505$ for structure D). This corresponds to online reconfiguration and controlling $V (= V_{max})$ at large feedrates. However, the loss is large.
- When controlling BS we avoid violating the maximum reactor holdup constraints at large feedrates ($F_0 > 460$) and large reactant feed fraction ($x_0 > 0.9$) by increasing the setpoint F_s . Alternatively, we may reconfigure online and control the reactor holdup M_r instead of x_r . This corresponds to controlling x_D which gives small losses. Switching the priority of x_D and x_r has no effect, since x_r is given when M_r and x_B are given.
- When controlling LS we avoid violating the maximum reactor holdup constraint at large feedrates ($F_0 > 460$), large reactant feed fractions ($x_0 > 0.9$) and with implementation error in F ($d_{c,2} < 0$), by increasing the setpoint $x_{D,s}$. Alternatively, we may reconfigure online and control the reactor holdup (M_r) instead of the top composition x_D . This corresponds to structure F which gives large losses at large feedrates F_0 . To avoid violating the maximum holdup rate at large feedrates ($F_0 > 520$) we switch to controlling $V (= V_{max})$ instead of F . By choosing a better weight matrix Q we would avoid violating the maximum reactor holdup constraint by switching to control the reactor holdup M_r instead of F . This corresponds to structure x_D which gives small losses.

Anyway, the conclusion has not changed. The loss is smaller and control is simpler if we keep x_D or L/F at constant nominal setpoints rather than controlling D or F at constant robust setpoints or controlling BS or LS at flexible setpoints.

4.4 Final evaluation and selection of control structure

The control properties of the two alternatives with the smallest losses (x_D and L/F), the two alternatives that follow Luybens rule (F and D) and the two alternatives which do not control the reactor holdup (LS and BS) are checked by performing dynamic simulations. With constant *nominal* controlling F gives instability in x_B and the Luyben Scheme (LS) gives instability in M_r with nominal setpoints. This is expected because of the infeasibility mentioned earlier. F gives the fastest control and x_D the slowest control with constant robust setpoints. Note that the control deviations are significantly smaller than the allowed ("expected") implementation errors. However, the "expected" implementation error also includes the measurement error which is not included in the simulations.

5 Conclusion

We have introduced several alternative methods for computing setpoints. The simplest is to use constant nominal setpoints, but this may give large loss in some cases or infeasible operation. One alternative is to find the best constant setpoint ("optimal backoff") by solving a quite complex robust optimization problem. Another alternative is to allow for online adjustments of the nominal setpoints such that we achieve feasibility (MPC adjustment) ("flexible backoff").

As a case study we have used a reactor, separator and recycle process. Control of x_D and L/F show the best self-optimizing control properties. Alternatives which follow Luybens rule (F and D), require robust setpoints and give larger losses than x_D and L/F . Alternatives with variable reactor holdup (e.g. Luyben Structure and Balanced Structure) require flexible setpoints and give significantly larger loss than x_D and L/F .

Although the feasibility region and the loss for a specific constant setpoint policy can be reduced by use of logic, model predictive control and/or online optimization, a good choice of controlled variables will nevertheless reduce the need for these remedies and give a simpler and cheaper system. Note that the required backoff and the corresponding economic loss depend on the selected controlled variables. Thus, the primary issue is to select the right control structure (variables), whereas the backoff is just a setpoint *adjustment* to deal with nonlinearities and infeasibility.

References

- [1] J.D. Perkins, C. Gannavarapu, and G.W. Barton. Choosing control structures based on economics. 1989.
- [2] A. Maarleveld and J.E. Rijnsdorp. Constraint control on distillation columns. *Automatica*, 6(1):51–58, 1970.
- [3] L.T. Narraway, J.D. Perkins, and G.W. Barton. Interaction between process design and process control: Economic analysis of process dynamics. *J. Proc. Cont.*, 1:243–250, Nov 1991.
- [4] L.T. Narraway and J.D. Perkins. Selection of process control structure based on linear dynamic economics. *Ind. Eng. Chem. Res.*, 32(11):2681–2692, 1993.
- [5] L.T. Narraway and J.D. Perkins. Selection of process control structure based on economics. *Comp. Chem. Engng.*, 18:511–515, 1994.
- [6] I.K. Kookos and J.D. Perkins. An algorithmic method for the selection of multivariable process control structure. *J. Proc. Control*, 12:85–99, 2002.
- [7] I.K. Kookos and J.D. Perkins. Regulatory control structure selection of linear systems. *Comp. Chem. Eng.*, 26:875–887, 2002.
- [8] Y. Arkun and G. Stephanopoulos. Studies in the synthesis of control structures for chemical processes. part vi: Design of steady-state optimizing control structures for chemical process units. *AIChE Journal*, 26(6):975–991, 1980.
- [9] M. Morari, Y. Arkun, and G. Stephanopoulos. Studies in the synthesis of control structures for chemical processes. part i: Formulation of the problem. process decomposition and the classification of the control task. analysis of the optimizing control structures. *AIChE Journal*, 26(2):220–232, 1980.
- [10] S. Skogestad. Plantwide control: The search for the self-optimizing control structure. *J. Proc. Control*, 10(5):487–507, 2000.

- [11] B. Glemmestad, S. Skogestad, and T. Gundersen. Optimal operation of heat exchanger networks. *Comp. Chem. Engng.*, 23:509–522, 1999.
- [12] C. Loeblein, J.D. Perkins, B. Srinivasan, and D. Bonvin. Performance analysis of on-line batch optimization systems. *Comp. Chem. Engng.*, 21:S867–S872, 1997.
- [13] S. Kall and S.W. Wallace. *Stochastic Programming*. John Wiley and Sons Ltd, 1994.
- [14] S.J. Qin and T.A. Badgwell. An overview of industrial model predictive control technology. *Chemical Process Control - CPC '95*, 1996.
- [15] A. Papadourakis, M.F. Doherty, and J.M. Douglas. Relative gain array for units in plants with recycle. *Ind. Eng. Chem. Res.*, 26(6):1259–1262, 1987.
- [16] K-L Wu and C-C Yu. Reactor/separators processes with recycle - 1. candidate control structure for operability. *Comp. Chem. Engng.*, 20(11):1291–1316, 1996.
- [17] T. Larsson, M.S. Govatsmark, S. Skogestad, and C.C. Yu. Control structure selection for reactor, separator and recycle processes. *Ind. Eng. Chem. Res.*, 42:1225–1234, 2003.
- [18] S. Skogestad and I. Postlethwaite. *Multivariable Feedback Control - Analysis and Design*. John Wiley & Sons, 1996.
- [19] M.L. Luyben, B.D. Tyreus, and W.L. Luyben. Plantwide control design procedure. *AIChE Journal*, 43(12):3161–3174, 1997.

List of captions

Figure 1:

A typical optimization system incorporating local feedback: The process is disturbed (d) and the control system tries to keep the controlled variables (c) at their setpoints (c_s). Steady-state optimization based on process measurements (y_m) is performed at regular intervals to track the optimum by updating the setpoints.

Figure 2:

Total feasible region (D_d^{total}) (solid), feasible region for a constant setpoint policy (D_d^c) (dashed) and expected disturbance region (D_d) (dash-dot).

Figure 3:

Cost as a function of the disturbance with (i) reoptimized setpoint (lower curve) and (ii) two alternative constant setpoint policies (c_1 and c_2). The loss is negligible with c_2 as a controlled variable.

Figure 4:

Cost as a function of the implementation error. The loss is negligible with c_3 as a controlled variable.

Figure 5:

Cost (J) as function of the controlled variable (c) at nominal point (d_0 , lower curve) and with disturbance (d_1 , upper curve).

Figure 6:

Reactor/separator process with liquid recycle

Figure 7:

Alternative x_D

Figure 8:

Alternative L/F

Figure 9:

Loss as a function of disturbances (F_0, x_0) and implementation errors ($d_{c,1}, d_{c,2}, d_{c,3}$) for the "good" controlled variables with (1) constant nominal setpoints (left column), (2) constant robust setpoints (middle column) and (3) flexible setpoints (right column).

Figure 10:

Loss as function of disturbances (F_0, x_0) and implementation errors ($d_{c,1}, d_{c,2}, d_{c,3}$) with "poor" controlled variables with (1) constant nominal setpoints (left column), (2) constant robust setpoints (middle column) and (3) flexible setpoints (right column).

Table 1:

Average and maximum percentage loss (L_w, L_{max}) with constant nominal setpoints, constant robust setpoints and flexible setpoints for alternative sets of controlled variables.