Chapter 6

A PARAMETRIC APPROACH TO OPTIMAL OPERATION OF HENs

The approach for optimal operation of HENs presented in this chapter is heavily inspired by the sign method in the previous chapter. However, instead of using only structural properties, *parametric* information in terms of linear steady state transfer matrices is utilized. The control configuration (pairing of manipulations and measurements) that minimizes utility cost is found from solving an integer programming problem during operation. As for the sign method described in the previous chapter, this parametric approach results in a decentralized control structure that may have variable pairing. The approach actually comprises three methods that differ from each other in the way controllability considerations are handled.

6.1 Introduction

The same problem as in the previous chapter is considered. That is, we want to develop a method for optimal operation of a given HEN. Decentralized control is used and the goal is to find an answer to the question: "*Which pairing reduces control error and minimizes utility cost at the same time?*" At steady state, the solution should give no control error and minimum utility cost. The information about the HEN is given by the transfer matrices G^{BP} and G^{U} introduced in the previous chapter. Recall that y^{BP} are the bypass controlled *temperatures*, while y^{U} are the *duties* of the utility exchangers assuming perfect control of the utility controlled temperatures.

$$y^{BP} = G^{BP}u \tag{6.1}$$

$$y^{U} = G^{U}u \tag{6.2}$$

In the previous chapter, the row vector g^{U} was introduced to denote the effect from each manipulation on the total utility consumption. The corresponding vector introduced in this chapter denotes the effect from each manipulation on the total utility *cost*. It is denoted $g^{\mathrm{U\$}}$ to indicate that it gives the cost rather than energy consumption. $g^{\mathrm{U\$}}$ is given by

$$g^{\mathrm{U\$}} = c^T G^{\mathrm{U}} \tag{6.3}$$

where c is a vector containing the utility cost for each utility exchanger. As opposed to the previous chapter where only sign matrices where utilized, we will in this section use the steady state gains of the transfer matrices to find the optimal control configuration. The method may be regarded as a "quantified" version of the sign method. The motivating example is presented in order to give a brief introduction to the idea.

Motivating example

The example uses the same HEN as in the illustrating example in section 5.2 (figure 5.1). The two following linear systems have been found from small (and positive) perturbations around the nominal operating point $u_{\mathrm{A}} = u_{\mathrm{B}} = 0$.

$$T_{\mathrm{o}}^{\mathrm{C2}} = \underbrace{\begin{bmatrix} 6.57 & -13.7 \end{bmatrix}}_{G^{\mathrm{BP}}} \begin{bmatrix} u_{\mathrm{A}} \\ u_{\mathrm{B}} \end{bmatrix} \tag{6.4}$$

$$\begin{bmatrix} Q_{\mathrm{c}} \\ Q_{\mathrm{h}} \end{bmatrix} = \underbrace{\begin{bmatrix} 4.49 & 6.87 \\ 7.76 & 0 \end{bmatrix}}_{G^{\mathrm{U}}} \begin{bmatrix} u_{\mathrm{A}} \\ u_{\mathrm{B}} \end{bmatrix} \tag{6.5}$$

Here, the bypass fractions are scaled from 0 to 1, the dimension of $T_{\mathrm{o}}^{\mathrm{C2}}$ is °C and the cooler and heater duties are in kW. Given a steam cost of 0.05 \$/kWh and cooling water cost of 0.01 \$/kWh, the cost vector is $c^T = [0.01\ 0.05]$. This yields

$$g^{\mathrm{U\$}} = c^T G^{\mathrm{U}} = \begin{bmatrix} 0.01 & 0.05 \end{bmatrix} \begin{bmatrix} 4.49 & 6.87 \\ 7.76 & 0 \end{bmatrix} = \begin{bmatrix} 0.433 & 0.069 \end{bmatrix} \tag{6.6}$$

That is, increasing u_{A} by one unit increases utility cost by 0.433 \$/h, and increasing u_{B} by one unit increases utility cost by 0.069 \$/h.

Dividing the elements in $g^{\mathrm{U\$}}$ by the elements in G^{BP} yields

$$\frac{g_1^{\mathrm{U\$}}}{g_{1,1}^{\mathrm{BP}}} = \frac{0.433}{6.57} = 0.066 \qquad \text{(Increasing } T_{\mathrm{o}}^{\mathrm{C2}} \text{ by 1°C } \textit{costs} \text{ 0.066 \$/h when } u_{\mathrm{A}} \text{ is used.)}$$

$$\frac{g_2^{\mathrm{U\$}}}{g_{1,2}^{\mathrm{BP}}} = \frac{0.069}{-13.7} = -0.005 \qquad \text{(Increasing } T_{\mathrm{o}}^{\mathrm{C2}} \text{ by 1°C } \textit{saves} \text{ 0.005 \$/h when } u_{\mathrm{B}} \text{ is used.)}$$

However, instead of *increasing* the bypass controlled temperature, the cost to bring it back to its target value is more interesting. Introducing the control error $e = T_{\mathrm{o}}^{\mathrm{C2}} - T_{\mathrm{t}}^{\mathrm{C2}}$, we can define a "priority matrix" P:

$$P = \begin{bmatrix} -\dfrac{g_1^{\mathrm{U\$}}}{g_{1,1}^{\mathrm{BP}}} e & -\dfrac{g_2^{\mathrm{U\$}}}{g_{1,2}^{\mathrm{BP}}} e \end{bmatrix} = \begin{bmatrix} -0.066e & 0.005e \end{bmatrix}$$

That is, if $e = 1$, it *saves* 0.066 \$/h to reach the target using u_A, while it would *cost* 0.005 \$/h if u_B is used. In general, P consists of as many rows as there are bypass controlled temperatures and as many columns as there are inputs that may be manipulated (here: bypasses). To minimize utility cost, the pairing is chosen such that the sum of the corresponding elements in P becomes as small as possible. In this case, the optimal pairing is easily found by hand. Assuming that $e = 1$, using u_A seem to be optimal choice. However, $g_1^{\mathrm{BP}} e > 0$ and this implies that u_A has to be decreased in order to reduce the control error. If we assume that $u_A = u_B = 0$, it is not possible to reduce e using u_A and we have to pair u_B and T_o^{C2}. The same result is found in the illustrating example in previous chapter (section 5.2). Here, however, the effects have been quantified. ■

Each time a control error changes sign or a new manipulation saturates, the procedure to find optimal pairing should be repeated. Also, the linear transfer matrices may be different in different operating regimes, and there may be a need to update the model in addition to the pairing. There is a need for a general model that enables automation of the procedure to find the pairing that optimizes utility cost.

In the next section, the integer programming formulation to find the pairing that minimizes utility cost is presented (model A). In that model, no dynamic considerations are made, and section 6.3 explains how control aspects may be included in the model. In section 6.4, model A is extended into a hierarchical strategy for operation that includes a simple controllability requirement (model B). Section 6.5 extends the model further into a MILP formulation (model C), which includes a slightly more complex controllability consideration. Before the chapter concludes, some examples to demonstrate the parametric approach are presented.

6.2 A model to find optimal pairing (model A)

The model for obtaining the optimal control configuration is based on the information contained in G^{BP} and G^{U}. These may be found from perturbation of a steady state model of the HEN. The vector $g^{\mathrm{U\$}}$ is obtained from G^{U} and utility cost data according to equation (6.3). Each element in the priority matrix P is given by

$$p_{i,j} = -\frac{g_j^{\mathrm{U\$}}}{g_{i,j}^{\mathrm{BP}}} e_i \tag{6.7}$$

where e_i is the control error of bypass controlled output i. Each entry of the priority matrix denotes the total utility cost (money per time unit) after control error i has been removed using manipulation j. When $g_{i,j}^{\mathrm{BP}}$ is small the corresponding value of $p_{i,j}$ is a large (positive or negative) number. A large negative $p_{i,j}$ may seem to be a good choice, however, this also indicate that the manipulation has a small effect on the output. A value of $p_{i,j}$ of infinity ($\pm\infty$) simply means that $g_{i,j}^{\mathrm{BP}}$ is zero and this pairing can not be used for control at all. To prevent such pairings, an element in P can be denoted [×] to indicate that the corresponding

element in G^{BP} is zero. In the actual implementation given by equation (6.13), the "×-values" in P can be given any numerical value since constraint (6.13f) prevents these pairings from being considered.

Note the similarity and difference from the definitions of P in equation (6.7) and equation (5.8). The definition in chapter 5 is applicable for sign matrices only, thus it does not matter whether we multiply or divide. In order to get the correct numerical value in equation (6.7), *division* by the elements of G^{BP} is required, while for sign matrices, multiplication is sufficient to get the correct sign.

The pairing that results in minimum total utility cost according to P should be selected. The total utility cost is found simply by adding elements in P corresponding to the selected pairing. In order to do this automatically, some variables and constants with logical (binary) elements are introduced. First, the variable matrix X with the same dimension as P is defined as

$$x_{i,j} = \begin{cases} 1 & \text{if } p_{i,j} \text{ is a selected pairing.} \\ 0 & \text{otherwise.} \end{cases} \tag{6.8}$$

A value of X corresponds to a given pairing, and we seek the optimal value of this variable. In addition, the following constant binary vectors/matrices are needed:

$$g^0_{i,j} = \begin{cases} 1 & \text{if } g^{BP}_{i,j} = 0 \\ 0 & \text{otherwise.} \end{cases} \tag{6.9}$$

$$y_{i,j} = \begin{cases} 1 & \text{if } g^{BP}_{i,j} e_i > 0 \text{ (which implies } u_j \downarrow \text{ to reduce } e_i\text{).} \\ 0 & \text{otherwise (which implies } u_j \uparrow \text{ to reduce } e_i\text{).} \end{cases} \tag{6.10}$$

$$z^{lo}_j = \begin{cases} 1 & \text{if } u_j \text{ is saturated at lower bound.} \\ 0 & \text{otherwise.} \end{cases} \tag{6.11}$$

$$z^{up}_j = \begin{cases} 1 & \text{if } u_j \text{ is saturated at upper bound.} \\ 0 & \text{otherwise.} \end{cases} \tag{6.12}$$

The values of these vectors/matrices may change during operation. For example, the values in y are dependent on the sign of the control error which will change. Further, which manipulations that are saturated (z^{lo} and z^{up}) can change. The values of y, z^{lo} and z^{up} are based on information from the DCS (Distributed Control System) in the plant. This information is updated immediately before each optimization to find a new pairing. Even if these vectors/matrices may change during operation, they are not subject to any changes by the optimization procedure. Hence, they are regarded as *constants* in the model to be optimized. The optimal pairing is found from the formulation given by the equations (6.13a) to (6.13f). Note that the model is a pure integer programming problem. The formulation is linear in the variables, thus it is denoted an Integer Linear Programming (ILP) problem.

$$X = \arg\min_{X} \sum_{i}\sum_{j} x_{i,j}\, p_{i,j} \quad \text{Find pairing } X \text{ that minimizes utility consumption.} \tag{6.13a}$$

subject to

$$\sum_{j} x_{i,j} = 1 \quad \forall i \quad \text{One bypass pr. bypass controlled temperature.} \tag{6.13b}$$

$$\sum_{i} x_{i,j} \le 1 \quad \forall j \quad \text{Each bypass can not control more than one output.} \tag{6.13c}$$

$$x_{i,j} + y_{i,j} + z_j^{\text{lo}} \le 2 \quad \forall i, j \quad \text{Reject pairing that requires } u\downarrow \text{ when } u = 0. \tag{6.13d}$$

$$x_{i,j} - y_{i,j} + z_j^{\text{up}} \le 1 \quad \forall i, j \quad \text{Reject pairing that requires } u\uparrow \text{ when } u = 1. \tag{6.13e}$$

$$x_{i,j} + g_{i,j}^{0} \le 1 \quad \forall i, j \quad \text{Reject pairing if } g_{i,j}^{\text{BP}} = 0\,. \tag{6.13f}$$

The objective function (equation 6.13a) is the pairing that minimizes total utility cost, i.e. the sum of $p_{i,j}$ (cost to restore a control error) for the selected pairing ($x_{i,j} = 1$). The purpose of the different constraints are explained in each of the equations (6.13b) to (6.13f). As an example, constraint (6.13d) forbids pairings that require a bypass decrease if it is already zero. If a pairing requires the manipulation to be decreased in order to reduce control error ($y_{i,j} = 1$) *and* that manipulation is saturated at lower bound ($z_j^{\text{lo}} = 1$), then equation (6.13d) forces $x_{i,j}$ to be zero (pairing is forbidden). Otherwise, $x_{i,j}$ is free to take any value (0 or 1). The equations (6.13a) to (6.13f) are referred to as "model A" in the subsequent.

If, for a given operating state, there is no pairing that reduces the control error, model A will not produce any feasible solution. In practice, it is important to know what to do if this should happen. One possibility is to keep the existing pairing and accept the control error, alternatively the plant could be switched into a predefined (and safe) control configuration. In the subsequent it is assumed that model A contains feasible solutions. It is emphasized that in practice, it is important to consider the consequences and remedies if no feasible solution exists.

The complete control algorithm can be summarized by the four following points:

1. Let SISO-controllers regulate bypass controlled output temperatures according to the last pairing that has been found. While this control configuration is active, check steps 2 and 3 continuously (or with frequent time intervals).
2. If the operating point has exceeded some limits or some timer has elapsed, update linear steady state model (G^{BP} and $g^{\text{U\$}}$).
3. Optimize model A to find a new pairing if

   a) a new bypass has saturated, or
   b) a control error has changed sign, or
   c) the model just has been updated.

   An alternative to the points a, b and c is to optimize model A at regular time intervals.
4. Repeat from step 1.

Note that the criterion for selecting pairing according to equation (6.13) is completely different from what is normally done for regulatory control. For regulatory control, the pairing is selected to achieve good dynamic behavior of the closed-loop system (operating cost is not considered), while the method presented above selects a pairing purely for utility cost minimization. Dynamic behavior has not yet been considered. It is clear that neglecting dynamics can result in undesired control configurations, e.g. the closed-loop system (or parts of it) may become unstable if a control loop becomes inactive due to saturation or is switched to manual operating mode. How to implement controllability issues is the topic of the next section. The remaining part of this section will draw some parallels to the sign method in the previous chapter.

In section 5.7, multiple bypass controlled outputs were briefly discussed for the sign method. Numerical values were assigned to the P matrix in equation (4.9), however, for the simple example in equation (5.10), constraints due to saturation (equations 6.13d and 6.13e) were not included. With the numerical values in P of –1, 0 and 1 (priority *i*, *ii* and *iii*), the model in equation (6.13) can be applied for the sign method when there are several bypass controlled temperatures. Hence, the possible pairings are evaluated in a simultaneous manner. The operability heuristics in step three of the sign method were used to distinguish between pairings of equal priority. This can be implemented in the automatic approach by penalizing the objective. The first heuristic rule may be implemented by increasing an element in P by e.g. 0.01 per exchanger the corresponding manipulation traverses before it affects the output. When implementing the sign method in this manner, however, it may not always be possible to use the priority table to find a simple split-range implementation of the control configuration.

6.3 How to implement controllability considerations

Model A (equations 6.13a to 6.13f) only considers operating cost. When the control configuration is selected with no regard to the dynamic properties, the closed-loop dynamic performance may be poor. This section describes how controllability aspects can be included. It is assumed that the only available information about the process is the two transfer matrices G^{BP} and G^{U} (at steady state) together with cost data for the utilities. That is, no explicit dynamic information is available.

The following two controllability requirements will be considered:

1. The plant (HEN) should remain stable when one or more control loops become inactive.
2. Among pairings with similar steady state properties, select the one with the best closed-loop performance

Requirement one is treated first, while the second is treated at the end of the section. A control loop becomes inactive if the input signal saturates. When saturation occurs, model A should be optimized and a new pairing should be found. Until the new pairing is active it is desired that the present closed-loop system is stable. A control loop may also become inactive if the controller is switched to a different operating mode (perhaps deliberately), e.g.

"tracking mode" where the control signal to the actuator is simply given a fixed value. Also, the controller may be switched to manual mode, and then the control loop is inactive if no operator actions are taken. (The loop is of course not inactive when it is operated manually; then the operator is part of the feedback loop). In cases when one or more control loops are inactive, it is highly desirable that the plant remains stable. Otherwise, plant hazard or severe product quality problems may be the result. To ensure stability, we will require that the plant is *decentralized integral controllable* (DIC). Decentralized integral controllability was introduced by Skogestad and Morari (1988), and the following definition is taken from Skogestad and Postlethwaite (1996):

Definition 6.1. *Decentralized integral controllability* (DIC).
The plant $G(s)$ (corresponding to a given pairing) is DIC if there exists a decentralized controller with integral action in each loop, such that the feedback system is stable and such that each individual loop may be detuned (reduced gain) independently by a factor ε_i ($0 \le \varepsilon_i \le 1$) without introducing instability. ■

$\varepsilon_i = 0$ corresponds to an inactive control loop. DIC is a somewhat stronger requirement than requirement 1 above since requirement 1 only concerns a detuning factor of 0 or 1 (not the intermediate values). Plants that are open-loop unstable are not DIC. This follows trivially since with all $\varepsilon_i = 0$, only the uncontrolled (and unstable) plant is left. However, all HENs are open-loop stable, as shown by Mathisen (1994, chapter 3).

No simple conditions (both necessary and sufficient) to test for DIC exists (Skogestad and Morari, 1988). Here, we need necessary conditions for DIC in order to screen pairing alternatives that result in plants that are not DIC. It is desirable with a condition that is easy and efficient to compute so the screening can be automated. Such a simple necessary condition exists and it is based on the steady state RGA. The RGA was introduced by Bristol (1966) as a measure of interactions in multivariable systems. For square systems, the RGA of a transfer matrix G is defined as:

$$RGA(G) = G \times (G^{-1})^T \tag{6.14}$$

The following theorem concerning the steady state RGA and stability was proven by Grosdidier *et al*. (1985).

Theorem 6.1 *Steady state RGA.*
Consider a stable square plant G and a diagonal controller K with integral action in all elements, and assume that the loop transfer function GK is strictly proper (always satisfied for real systems). If a pairing of outputs and manipulated inputs corresponds to a negative steady state relative gain, then the closed-loop system has at least one of the following properties:

1. The overall closed-loop system is unstable.
2. The loop with the negative relative gain is unstable by itself (other loops opened).
3. The closed-loop system is unstable if the loop with the negative relative gain is inactive.

■

This theorem implies the following necessary condition for DIC (Skogestad and Postlethwaite (1996):

A plant $G(s)$ is DIC only if $RGA_{i,j}(G(0)) \geq 0$ for the selected pairings i,j. (6.15)

That is, pairings corresponding to negative steady state RGA elements should be avoided as it will result in a plant which is not DIC. There are other necessary conditions and also sufficient conditions for DIC, see Skogestad and Morari (1988). Note that the condition given by (6.15) is necessary, meaning that even if all selected RGA elements are positive there is no guarantee that the plant is DIC. The steady state RGA, however, is a simple tool since it does not have to be recomputed for each possible pairing. Therefore, we will here only use the simple test given by (6.15) to screen plants that are not DIC. For 2×2-plants, positive steady state RGA-elements are both necessary and sufficient conditions for DIC (Skogestad and Morari, 1988).

The screening method for HENs is based on RGA computed from G^{BP}. This transfer matrix is usually not square as there are more manipulations (bypasses) than outputs (targets). The control configuration being active at any moment, however, is square. Unused manipulations are usually saturated, and each of the active manipulations are paired with one controlled outputs using single loop controllers. Hence, the RGA is computed from G^{BP} with the columns corresponding to unused inputs removed. Since the unused inputs may vary, it is necessary to consider several RGAs, and the following notation is introduced:

RGA^k is the steady state RGA computed from G^{BP} with column k removed. (6.16)

In case of more than one unused inputs, this is indicated with more indices, e.g. $RGA^{k,l}$ the steady state RGA computed from G^{BP} with columns k and l removed. In order to get consistency when addressing individual elements in RGA^k and G^{BP}, columns with only zeros are inserted in RGA^k at the locations of the unused inputs. As an example, we have

$$G^{BP} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & -5 & 6 \end{bmatrix},\ RGA^2 = \begin{bmatrix} -1 & 0 & 2 \\ 2 & 0 & -1 \end{bmatrix} \text{ since } RGA\left(\begin{bmatrix} 1 & 3 \\ 4 & 6 \end{bmatrix}\right) = \begin{bmatrix} -1 & 2 \\ 2 & -1 \end{bmatrix}. \quad (6.17)$$

Now, the pairing with negative relative gains can be forbidden by adding new constraints to model A. Recall the variable X, where $x_{i,j} = 1$ for selected pairings and $x_{i,j} = 0$ for the remaining elements. To forbid pairing of output 1 with input 1 and output 2 with input 3 for the system given by equation (6.17), we could include the constraints $x_{1,1} = 0$ and $x_{2,3} = 0$. For the remaining 2×2-system (input 2 is unused) it will be sufficient to include only one of these constraints. This method, however, will forbid each of the two pairings in all cases and this is not what we want. If input 1 or 3 is unused, we do *not* want to forbid pairing $x_{1,1}$ or $x_{2,3}$. To forbid both $x_{1,1}$ and $x_{2,3}$ to be 1 at the same time, the following constraint should be included instead:

$$x_{1,1} + x_{2,3} \leq 1 \quad (6.18)$$

With both these pairings selected, the left-hand side of (6.18) is 2 thus the constraint is violated. This constraint, however, allows for selecting e.g. the pairing $x_{1,1} = 1$ and $x_{2,2} = 1$ in the case where input 3 is unused. With 2 controlled outputs the right-hand side of the

constraint is 1 in order to prevent selection of the 2 elements in X with negative relative gains. More general, the sum of each pairing X corresponding to a negative relative gain should be less than or equal to $n_y - 1$ (n_y is the number of controlled outputs). If there are several pairings that have at least one negative relative gain, a constraint similar to equation (6.18) should be included for each of these pairings. Thus, pairings resulting in plants that are not DIC are forbidden. The next section shows how pairings resulting in negative gains (not DIC) are not strictly forbidden, but they are disregarded if there exist any possible pairing with non-negative RGA-elements.

The steady state RGA will also be used for the second controllability requirement given above (best closed-loop performance for pairings with similar steady state properties). We choose to penalize the RGA-number, which for a square plant G with rows and columns ordered such that the pairings are along the diagonal, is:

$$\text{RGA-number} = \left\| RGA(G) - I \right\|_{\text{sum}} \tag{6.19}$$

The sum norm is simply the sum of the absolute values of all elements. The RGA-number is a measure of the diagonal dominance (Skogestad and Postlethwaite, 1996, chapter 10). In our case, the rows and columns of the plant are not ordered, but in model A we have the variable X denoting the selected pairings of G^{BP}. The RGA-number when input k is unused is:

$$\text{RGA-number with input } k \text{ unused} = \left\| RGA^k - X \right\|_{\text{sum}} \tag{6.20}$$

The matrix X with column k removed can be considered as a permutation of I (in 6.19), or as the permutation matrix itself. The matrices in equation (6.20) are not square, however, the elements in the column(s) corresponding to the unused input(s) are zero, both for RGA^k and X. Hence, the unused columns do not contribute to the RGA-number. Penalizing the RGA-number given by (6.20) for the different alternatives of RGA^k, represents a method to include controllability considerations for selection among pairings with similar steady state performance. This method will be used to handle the second controllability requirement.

The next section explains in more detail how model A is extended to include the first controllability requirement presented above, and a hierarchical approach is proposed. Section 6.5 shows how both the first and the second controllability requirement can be included in a more complex hierarchical approach.

6.4 A first hierarchical strategy for operation (model B)

In the previous sections, pairings that have to result in a plant that is not DIC were forbidden by introducing new constraints to model A (equation 6.13). In cases where no DIC plants exist, this approach implies that there are no feasible solution to the pairing problem. This infeasibility is not desired as we still want to operate the plant in the best possible way. This section introduces a hierarchical approach by extending model A to also handle cases where no DIC plants exist. We want to obtain the following strategy for operation:

1. Select the pairing that results in the lowest utility consumption among configurations that are DIC.

2. If no DIC pairing exists, select the pairing that results in the lowest utility consumption (among the plants that are not DIC).

The first point corresponds to the strategy described in the previous sections. The second point cannot be implemented simply by forbidding pairings that results in plants being not DIC. Further, simply penalizing negative values of RGA-elements will not result in the desired strategy since pairings with small negative relative gains (and low utility consumption) may be preferred before plants with only positive relative gains (and high utility consumption). This is not consistent with the two items above. The strategy can be implemented by introducing logical inference in the model. This is explained in the subsequent. Consider the following inequality where α is a logical variable (binary variable, 0-1 variable) and r is a real scalar variable:

$$-M\alpha + r \leq 0 \tag{6.21}$$

M is an upper bound on the real variable r. This constraint connects the real and binary variables in the following way:

$$\text{If } \alpha = 0, \text{ then } r \leq 0. \tag{6.22a}$$
$$\text{If } \alpha = 1, \text{ then } r \text{ is free to take any value (up to } M\text{).} \tag{6.22b}$$

Such constraints are often included e.g. in optimization of process structures using mathematical programming. If r is restricted to a positive real value denoting the size (or cost) of a unit, and $\alpha = 1$ if that unit is present and $\alpha = 0$ otherwise, then constraint (6.21) forces the size (or cost) of the unit to be zero when the unit is not selected (see e.g. Biegler *et al.* (1997). Here, we shall consider r being both positive and negative and also it will be focused on the "complementary" statement of (6.22). That is, instead of using the value of α to draw conclusions about r, we will use the value of r to draw conclusions about α. If a and b are logical propositions, then the following two statements are equivalent:

$$\text{“}a \Rightarrow b\text{” is equivalent to “(not } b) \Rightarrow (\text{not } a)\text{”} \tag{6.23}$$

Using this, constraint (6.21) results in the following relationship between the binary and real variable:

$$\text{If } r > 0, \text{ then } \alpha = 1. \tag{6.24a}$$
$$\text{If } r \leq 0, \text{ then } \alpha \text{ is free to take any value (0 or 1).} \tag{6.24b}$$

In our model, we want $\alpha = 1$ if there exists at least one pairing with no relative gains, and $\alpha = 0$ if no such pairing exists (meaning that all pairings are not DIC). The first step to do this, is to establish the proper relationship between a single variable r (which may be one element of the RGA) and a single binary variable that is denoted β. We want to formulate constraints that are equivalent to the following statements:

$$\text{If } r \geq 0, \text{ then } \beta = 1. \tag{6.25a}$$
$$\text{If } r < 0, \text{ then } \beta = 0. \tag{6.25b}$$

Statement (6.24a) is close to (6.25a) but it should be modified such that the equality part ($r = 0$) forces β to 1. This can be done by changing constraint (6.21) into

$$-RGA_{\max}\beta + (r + \delta) \leq 0 \tag{6.26}$$

$RGA_{\max}$ is the largest absolute value of any element in the RGA thus representing an upper bound on r. The new variable δ is the absolute value of an upper bound (closest to 0) on negative values of r. That is, if the largest negative value r can have is –0.01, then δ must be in the interval $\delta \in (0, 0.01)$ (not including the endpoints).

To satisfy statement (6.25b), the following constraint is introduced:

$$RGA_{\max}(\beta - 1) - (r + \delta) \leq 0 \tag{6.27}$$

Here, $RGA_{\max}$ represents a bound of the largest (in absolute value) negative value of r. With a proper value of δ, constraint (6.27) is equivalent to:

If $r < 0$, then $\beta = 0$. (6.28a)
If $r \geq 0$, then β is free to take any value (0 or 1). (6.28b)

To conclude so far, we have found that the desired statements (6.25a) and (6.25b) can be satisfied by introducing the two constraints (6.26) and (6.27), and the result can be summarized as:

$$\left.\begin{aligned} -RGA_{\max}\beta + (r+\delta) \leq 0 \\ RGA_{\max}(\beta - 1) - (r+\delta) \leq 0 \end{aligned}\right\} \quad \Rightarrow \quad \begin{cases} \text{if } r \geq 0, \text{ then } \beta = 1 \\ \text{if } r < 0, \text{ then } \beta = 0 \end{cases} \tag{6.29}$$

Actually, the correct statements in (6.29) are: If $r > -\delta$, then $\beta = 1$ and if $r \leq -\delta$, then $\beta = 0$. However, with a proper value of δ, this is equivalent to the statements in (6.29). These constraints provide the required connection between one single real number r and the logical variable β. In our case, we want to distinguish whether it can be concluded from negative relative gains that all possible pairings and choice of inputs have to result in plants that are not DIC, or not. This makes the picture somewhat more complicated. β can not just be one single variable; we need one binary value for each element in the RGAs. Using the RGA for selecting pairing is efficient since it does not have to be recomputed for each pairing. It does, however, have to be recomputed for different selections of unused inputs, recall RGA^k introduced in the last section. In the model, β is a set of k logical variables, each with the same dimension as RGA^k. The following constraints provide the necessary relationship between each of the RGAs and the βs:

$$-RGA_{\max}\beta_{i,j}^k + \left(RGA_{i,j}^k + \delta\right) \leq 0 \quad \forall i, j, k \tag{6.30a}$$

$$RGA_{\max}\left(\beta_{i,j}^k - 1\right) - \left(RGA_{i,j}^k + \delta\right) \leq 0 \quad \forall i, j, k \tag{6.30b}$$

These constraints ensure that for each unused input k, the elements of β^k are 0 if the corresponding element in RGA^k is negative, and 1 if the corresponding element in RGA^k is zero or positive. As numerical values for the bounds, $RGA_{\max}$ can be set equal to the largest

element (in absolute value) found in any of the RGA^ks and δ can be set equal to the absolute value of the negative number closest to zero found in any RGA^k.

The model including controllability considerations shall reflect the hierarchical strategy given by the two requirements stated first in this section. When extending model A to incorporate this strategy, the following two items are important:

- The model size should be kept small, i.e. if it is possible to carry out some of the computations in advance of the optimization, this should be done.
- No non-linearities should be introduced as this may make the search for the optimal pairing significantly more difficult.

When an optimization is to be performed, the steady state transfer matrix G^{BP} is known. From this matrix, the RGA^ks can be computed in advance. From each RGA^k, also the β^ks may be precomputed. This may be done by implementing the constraints (6.30a) and (6.30b) *separately*, or by means of if-then statements in any conventional computer language. Hence, both the *RGA*s and the βs are computed in advance. This reduces the size of the model, but another advantage concerning the second item above is perhaps more important: With precomputed values of the *RGA*s and the βs, these become *constants* (and not variables) in the optimization procedure. This allows variables to be multiplied by for example β without introducing non-linearities and this is very convenient.

The connection between the set of the βs and the variable α is done by adding 2 new constraints. First, consider the following term:

$$\sum_{i,j}\left(x_{i,j}^k \beta_{i,j}^k\right) \tag{6.31}$$

The multiplication by $x_{i,j}$ within the summation ensures that only the selected pairings are considered (since for pairings that are not selected we have $x_{i,j} = 0$). The constraints in model A (more specifically; constraint 6.13b) ensure that the number of selected pairings is equal to the number of controlled outputs n_y. If all β-elements of the selected pairings are 1 (i.e. relative gains are non-negative) then the term in (6.31) is equal to n_y. If, however, any of the selected relative gains are negative (i.e. $\beta_{i,j} = 0$ for the selected k), then the term in (6.31) is less than n_y and this implies that the plant with this pairing is not DIC. That is, the constraints that connect α and β must be equivalent to the following statement:

For the selected k:

If $\sum_{i,j}\left(x_{i,j}^k \beta_{i,j}^k\right) < n_y$ (one or more negative RGA-elements), then $\alpha = 0$. (6.32a)

If $\sum_{i,j}\left(x_{i,j}^k \beta_{i,j}^k\right) = n_y$ (all RGA-elements are zero or positive), then $\alpha = 1$. (6.32b)

This statement is fulfilled by introducing the following two constraints:

$$n_y\alpha - \sum_{i,j}\left(x_{i,j}\beta_{i,j}^k\right) \le 0 \quad \text{for the selected } k. \tag{6.33a}$$

$$-\alpha + 1 - n_y + \sum_{i,j}\left(x_{i,j}\beta_{i,j}^{k}\right) \le 0 \quad \text{for the selected } k. \tag{6.33b}$$

Constraint (6.33a), forces α to zero if $\Sigma_{i,j}(x_{i,j}\beta_{i,j}^{k}) < n_y$ (otherwise α is free), thus requirement (6.32a) is fulfilled. Likewise, constraint (6.33b) fulfills requirement (6.32b). With the correct value of α, the only remaining task is to prevent pairings resulting in a not DIC plant when there exist DIC pairings that are feasible (fulfills the constrains in model A). This is done simply by adding the term $M_1(1-\alpha)$ to the objective function. M_1 represents an upper bound on the *change* of the objective function from model A (from equation 6.13a). That is, M_1 should be at least as large as the difference between the best and worst case solutions from model A, thus changing α from 0 to 1 will always dominate other changes in the objective function. A suitable value of M_1 can be found from adding the n_y (since n_y is the number of selected elements) largest elements in P and subtracting the n_y smallest elements in P. This should be done when P has been updated prior to an optimization, hence M_1 is a constant in the optimization algorithm.

Now, the necessary extensions to model A have been presented, and the complete extended model B is given by equations (6.34) to (6.36).

$$X = \arg\min_{X}\left(\sum_{i}\sum_{j} x_{i,j}p_{i,j} + M_1(1-\alpha)\right) \tag{6.34}$$

subject to

$$\sum_{j} x_{i,j} = 1 \qquad \forall i \tag{6.35a}$$

$$\sum_{i} x_{i,j} \le 1 \qquad \forall j \tag{6.35b}$$

$$x_{i,j} + y_{i,j} + z_j^{\text{lo}} \le 2 \qquad \forall i,j \tag{6.35c}$$

$$x_{i,j} - y_{i,j} + z_j^{\text{up}} \le 1 \qquad \forall i,j \tag{6.35d}$$

$$x_{i,j} + g_{i,j}^{0} \le 1 \qquad \forall i,j \tag{6.35e}$$

$$n_y\alpha - \sum_{i,j}\left(x_{i,j}\beta_{i,j}^{k}\right) \le 0 \quad \text{for the selected } k. \tag{6.36a}$$

$$-\alpha + 1 - n_y + \sum_{i,j}\left(x_{i,j}\beta_{i,j}^{k}\right) \le 0 \quad \text{for the selected } k. \tag{6.36b}$$

As we see, the constraints in (6.35) are the same as the constraints in model A, while the constraints in (6.36) together with the modified objective function handle the controllability requirement. The model works in the following way:

- If none of the pairings have negative RGA-elements ($\alpha = 1$), the model is identical to model A.
- If there are some pairings (that fulfills the constraints in 6.35) where $\alpha = 1$ and others where $\alpha = 0$, then the optimal pairing is that with the smallest operating cost among

those with $\alpha = 1$. That is, even the pairing with the worst case operating cost is preferred before a plant we know is not DIC.

- If all pairings that satisfy the constraints in (6.35) result in a not DIC plant, then the pairing with the smallest operating cost is selected. This last case, however, represents a *potentially* hazardous situation since the plant (or parts of it) will become unstable if one of the selected control loops becomes inactive.

Model B represents an implementation of the two requirements given early in this section, corresponding to a hierarchical strategy for operation. The extension to include the controllability considerations comprises only two additional constraints and a new term in the objective function. The use of this model will be demonstrated in the example section.

6.5 A second hierarchical strategy for operation (model C)

This section presents a hierarchical strategy for operation which is different from that in the previous section. Here, the second controllability consideration (among pairings with similar operating costs, select that with the best controllability properties) is included. The strategy differs from the previous also in the sense that if no DIC plant exists, then utility optimization is sacrificed and focus is exclusively on controllability. We want to apply the following strategy:

1. Select the pairing that results in the lowest utility consumption among configurations that are DIC. Among pairings with similar utility consumption, select that with the best controllability properties.
2. If no DIC pairing exists, give up utility optimization and select the pairing with the best controllability properties.

This strategy is obtained from modifying the objective function of the previous models and extending model B further. The second controllability consideration involves computation of the RGA-number as explained in section 4.3, see equation (6.20). As for the previous model, it is important to keep the model linear. It will become clear that implementing the strategy above, and particularly computation of the RGA-number, represents a major extension of model B.

For the new objective function, we introduce a new variable OC (change in operating costs):

$$OC = \sum_{i,j} x_{i,j} p_{i,j} \tag{6.37}$$

OC is identical to the objective function in the previous models. To disregard the operating costs when we know the plant is not DIC, it could be tempting to simply include the term αOC in the objective function. However, this is a multiplication of the binary variable α with the continuous variable OC and thus it will introduce a non-linearity. A similar term can be implemented without introducing non-linearities by means of logic inference. This is done with the variable OC^* that should have the following properties (identical to the term αOC):

$$\text{If } \alpha = 0, \text{ then } OC^* = \ 0. \tag{6.38a}$$
$$\text{If } \alpha = 1, \text{ then } OC^* = OC. \tag{6.38b}$$

This is implemented by means of constraints quite similar to constraint (6.21), but four constraints are required. They are explained below.

$$-M_2\alpha - OC^* \le 0 \tag{6.39a}$$
$$-M_2\alpha + OC^* \le 0 \tag{6.39b}$$
$$M_2(\alpha - 1) - OC^* + OC \le 0 \tag{6.39c}$$
$$M_2(\alpha - 1) + OC^* - OC \le 0 \tag{6.39d}$$

M_2 is an upper bound on the operating costs. Constraint (6.39a) is equivalent: If $\alpha = 0$, then $OC^* \ge 0$, otherwise OC^* is free. Likewise, constraint (6.39b) is equivalent: If $\alpha = 0$, then $OC^* \le 0$, otherwise OC^* is free. The intersection of these two constraints (both constraints are fulfilled) corresponds to the equality part of both. That is: If $\alpha = 0$, then $OC^* = 0$, otherwise OC^* is free. Hence, the constraints (6.39a) and (6.39b) together ensure that requirement (6.38a) is satisfied.

The argument is similar for requirement (6.38b). Constraint (6.39c) is equivalent to: If $\alpha = 1$, then $OC^* \ge OC$, otherwise OC^* is free. Likewise, constraint (6.39d) is equivalent to: If $\alpha = 1$, then $OC^* \le OC$, otherwise OC^* is free. The intersection of these two constraints corresponds to the equality part of both. Hence, the constraints (6.39c) and (6.39d) together ensure that also requirement (6.38b) is satisfied.

Note that since model C minimizes OC^*, it is not necessary to provide any upper bounds on this variable. Thus, only constraints (6.39a) and (6.39c) are actually included in the model. This corresponds to replacing the 2 equalities in (6.38) by "greater than or equal"-signs

The objective function in the extended model has the form

$$X = \arg\min_X \left(OC^* + wCM + M_1(1-\alpha) \right) \tag{6.40}$$

The first term, operating cost, is implemented by the constraints (6.37), (6.39a) and (6.39c). The second term involves the variable *CM* (controllability measure) and the weight w that scales the controllability relative to OC^* when this is not zero. As controllability measure, we have decided to use the RGA-number, see equation (6.20). This measure is not computed in advance since it is not only a function of the selected RGA^k, but also on the pairing X (that would require computation of the RGA-number for a large number of pairings). To compute the RGA-number, a new variable γ is introduced. This is a binary variable with same dimension as RGA^k (and X). For each element, $\gamma_{i,j} = 1$ if the corresponding element in $(RGA^k - X)$ is positive and $\gamma_{i,j} = 0$ if the corresponding element in $(RGA^k - X)$ is negative. For elements in $(RGA^k - X)$ that are zero, it does not matter which value $\gamma_{i,j}$ takes. For simplicity we choose $\gamma_{i,j} = 0$ in such cases. That is, we want to obtain:

$$\text{If } \left(RGA^k_{i,j} - x_{i,j} \right) > 0 \text{, then } \gamma_{i,j} = 1$$
$$\text{If } \left(RGA^k_{i,j} - x_{i,j} \right) \le 0 \text{, then } \gamma_{i,j} = 0$$

This is achieved by the following constraints:

$$-M_3\gamma_{i,j} + \left(RGA^k_{i,j} - x_{i,j}\right) \le 0 \quad \forall i, j \tag{6.41a}$$

$$M_3\left(\gamma_{i,j} - 1\right) + \left(RGA^k_{i,j} - x_{i,j}\right) \le 0 \quad \forall i, j \tag{6.41b}$$

M_3 is and upper bound on the absolute value of $RGA^k_{i,j} - x_{i,j}$. A simple expression for M_3 is $M_3 = RGA_{\max} + 1$. The next step is to use γ to create a new continuous variable *RGAn* of the same size as γ: The purpose of *RGAn* is that the RGA-number can be found simply from adding all elements in *RGAn*. That is, *RGAn* should have the following definition:

$$\text{If } \gamma_{i,j} = 1, \text{ then } RGAn_{i,j} = \left(RGA^k_{i,j} - x_{i,j}\right)$$

$$\text{If } \gamma_{i,j} = 0, \text{ then } RGAn_{i,j} = -\left(RGA^k_{i,j} - x_{i,j}\right)$$

This is satisfied by introducing the following constraints (similar as 6.39):

$$-M_3\gamma_{i,j} - RGAn_{i,j} + \left(RGA^k_{i,j} - x_{i,j}\right) \le 0 \quad \forall i, j \tag{6.42a}$$

$$-M_3\gamma_{i,j} + RGAn_{i,j} - \left(RGA^k_{i,j} - x_{i,j}\right) \le 0 \quad \forall i, j \tag{6.42b}$$

$$M_3\left(\gamma_{i,j} - 1\right) - RGAn_{i,j} + \left(RGA^k_{i,j} - x_{i,j}\right) \le 0 \quad \forall i, j \tag{6.42c}$$

$$M_3\left(\gamma_{i,j} - 1\right) + RGAn_{i,j} - \left(RGA^k_{i,j} - x_{i,j}\right) \le 0 \quad \forall i, j \tag{6.42d}$$

The intersection (equality parts) of (6.42a) and (6.42b) ensures correct value of *RGAn* when $\gamma_{i,j} = 0$, while constraints (6.42c) and (6.42d) handles the elements where $\gamma_{i,j} = 1$. Apparently, the computation of the controllability measure causes a significant extension of the size of the model. With the values of *RGAn* from constraint (6.42), the RGA-number (*CM*) is simply given by a summation as shown by equation (6.45g) below.

Now, the extensions to include controllability are explained, and the complete extended model is given by equations (6.43) to (6.45). Note that with the introduction of the continuous variables *OC*, OC^* and *CM*, the model has become a MILP-formulation.

$$X = \arg\min_X \left(OC^* + wCM + M_1(1-\alpha)\right) \tag{6.43}$$

subject to

All constraints in model B. That is, equations (6.35) and (6.36).
In addition:

$$OC = \sum_{i,j} x_{i,j}\, p_{i,j} \tag{6.44a}$$

$$-M_2\alpha - OC^* \le 0 \tag{6.44b}$$

$$M_2\,(\alpha - 1) - OC^* + OC \le 0 \tag{6.44c}$$

$$-M_3\gamma_{i,j} + \left(RGA^k_{i,j} - x_{i,j}\right) \le 0 \quad \forall i, j \tag{6.45a}$$

$$M_3\left(\gamma_{i,j}-1\right)+\left(RGA_{i,j}^k-x_{i,j}\right)\le 0 \quad \forall i,j \tag{6.45b}$$

$$-M_3\gamma_{i,j}-RGAn_{i,j}+\left(RGA_{i,j}^k-x_{i,j}\right)\le 0 \quad \forall i,j \tag{6.45c}$$

$$-M_3\gamma_{i,j}+RGAn_{i,j}-\left(RGA_{i,j}^k-x_{i,j}\right)\le 0 \quad \forall i,j \tag{6.45d}$$

$$M_3\left(\gamma_{i,j}-1\right)-RGAn_{i,j}+\left(RGA_{i,j}^k-x_{i,j}\right)\le 0 \quad \forall i,j \tag{6.45e}$$

$$M_3\left(\gamma_{i,j}-1\right)+RGAn_{i,j}-\left(RGA_{i,j}^k-x_{i,j}\right)\le 0 \quad \forall i,j \tag{6.45f}$$

$$CM=\sum_{i,j}RGAn_{i,j} \tag{6.45g}$$

The constraints in (6.44) concern operating costs and except for (6.44a) they concern how to disregard the operating costs when the pairing gives a plant that is not DIC. The constrains in (6.45) concern computation of the RGA-number (second controllability requirement).

Model C represents an implementation of the two requirements given early in this section, corresponding to a more complex hierarchical strategy for operation than model B. As it has become evident, model B had to be extended quite extensively to incorporate the second controllability requirement.

6.6 Examples

This section includes two examples to demonstrate the application of the method described in this chapter. Example 6.1 is a very simple and academic example where model B is demonstrated. Example 6.2 shows how model C may be significantly simplified for a seven stream HEN.

Example 6.1 *Simple demonstration example.*
Consider a plant with 3 manipulable inputs and 2 outputs given by the following transfer function:

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}=\frac{1}{s+1}\underbrace{\begin{bmatrix} 3 & -1 & 3 \\ 4 & -2 & -6 \end{bmatrix}}_{G^{\mathrm{BP}}}\begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}+\frac{1}{s+1}\begin{bmatrix} 0 \\ 1 \end{bmatrix}d \tag{6.47}$$

where $u_1, u_2, u_3 \in [0, 1]$ and the objective function is $J = u_1 + u_2 + u_3$ ($G^{\mathrm{U\$}} = [1\ \ 1\ \ 1]$). Even if the transfer function in (6.47) may not represent a HEN, we will let superscript U\$ represent operating costs and G^{BP} denotes the steady state transfer function from the manipulated inputs to the controlled outputs. Only the steady state data is known. The transfer function from the disturbance as well as the disturbance d itself is unknown; we can only register the effect on the measurements in terms of control errors.

The problem is to control both outputs to setpoints while minimizing the objective J. The control configuration should result in a DIC plant, and if no DIC plants are possible when the outputs are controlled, then optimization of J is sacrificed and the best control configuration should be selected.

The three RGA^ks computed from G^{BP} (input k unused) are:

$$RGA^1 = \begin{bmatrix} 0 & 0.5 & 0.5 \\ 0 & 0.5 & 0.5 \end{bmatrix}, \quad RGA^2 = \begin{bmatrix} 0.6 & 0 & 0.4 \\ 0.4 & 0 & 0.6 \end{bmatrix} \quad \text{and} \quad RGA^3 = \begin{bmatrix} 3 & -2 & 0 \\ -2 & 3 & 0 \end{bmatrix}$$

For each of the three cases there are two pairings, but only one pairing has negative relative gains. However, it may not always be possible to select the pairing with positive relative gains due to the constraints in the manipulations. The corresponding precomputed values of the βs are:

$$\beta^1 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \quad \beta^2 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad \text{and} \quad \beta^3 = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}.$$

The initial values are $u_1 = u_2 = u_3 = 0$ and also $d = 0$. The result using model B, showing the time responses for different disturbances when the pairing changes according to the optimal pairing is shown. Initially, we have all $z^{lo} = 1$, i.e. all manipulations are saturated at zero.

At time zero, both setpoints are changed from 0 to 0.5. At the same time, an optimization is done and this results in controlling y_1 using u_3 and controlling y_2 using u_1. The different events in terms of setpoint changes and disturbance are as follows:

At time = 0: Setpoints r_1 and r_2 are changed from 0 to 0.5.
At time = 4: Setpoint r_2 is changed from 0.5 to 0.
At time = 8: Setpoint r_1 is changed from 0.5 to 0.
At time = 14: Disturbance d is changed from 0 to 1.

This is shown in figure 6.1 where the thin solid lines in 6.1a and 6.1b are the setpoints r_1 and r_2, respectively. The disturbance d is shown in figure 6.1c. Optimization to find the control structure (step 3 in the procedure on page 95) is done every fifth minute, regardless of saturation or change of sign for control errors. The resulting pairing matrix X in each period became:

Time: 0-5 min. $\quad X = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$

Time: 5-10 min. $\quad X = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$

Time: 10-15 min. $\quad X = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

Time: 15-20 min. $\quad X = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$

The result from dynamic simulations when the system is controlled according to these control structures is shown in figure 6.1. The thick lines in figures 6.1a and 6.1b shows the controlled outputs y_1 and y_2. The manipulations are shown in the 3 figures in the right column. Simple PI-controllers are used for the simulations. In all cases, the controller gain is 1 or –1 (depending on the sign of the transfer function) and the integral time is 1 minute.

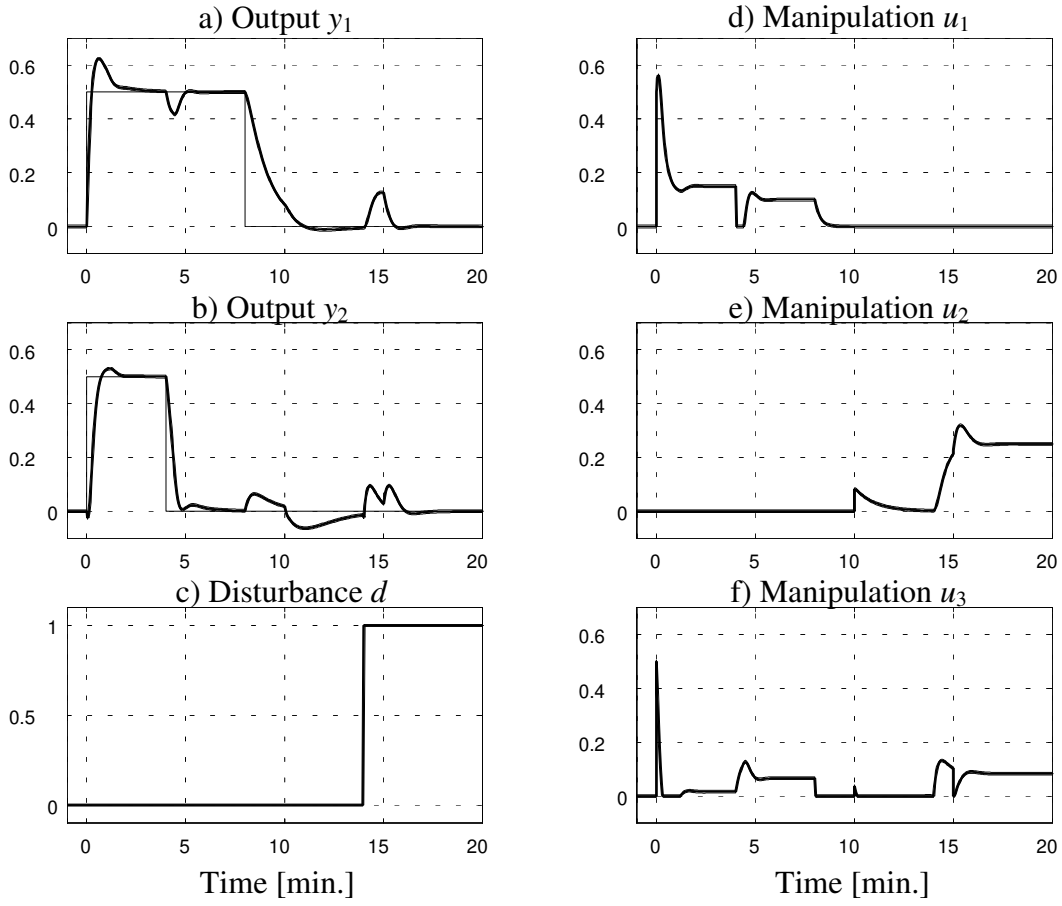

Figure 6.1 Result for example 6.1.

In addition to demonstrate how the method works, this example has also shown that the approach may be applied to other processes than HENs. As long as the process is represented by one steady state transfer function G^{BP} from the manipulations to the controlled outputs, and one transfer function from the manipulations to a scalar objective, the method may be applied. ■

Example 6.2 *Seven-stream HEN example.*
The HEN in this example has the same structure as the largest subnetwork of the aromatics plant studied by Linnhoff *et al.* (1982). The HEN is shown in the figure below.

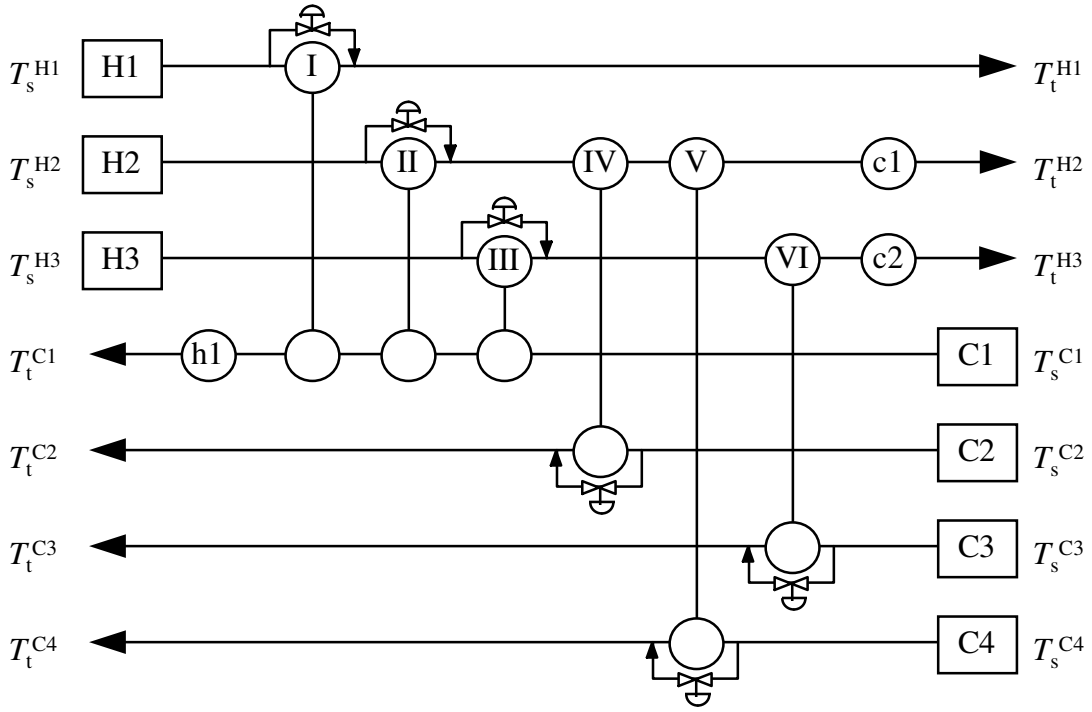

Figure 6.2 HEN in example 6.2.

The stream data has been modified slightly such that the *CP*-values are constant (segments have been removed). As usual, the HEN is decomposed according to rule 2.1 (in section 2.3) and the bypassed controlled temperatures are T_o^{H1}, T_o^{C2}, T_o^{C3} and T_o^{C4}. The transfer functions G^{BP} and G^{U} have been found from small positive perturbations of the nonlinear process model around the operating point $u_I = u_{II} = u_{III} = u_{IV} = u_V = u_{VI} = 0$.

$$\begin{bmatrix} T_o^{H1} \\ T_o^{C2} \\ T_o^{C3} \\ T_o^{C4} \end{bmatrix} = \underbrace{\begin{bmatrix} 58.8 & -13.5 & -1.2 & 0 & 0 & 0 \\ 0 & 14.9 & -2.1 & -3.6 & 0 & 0 \\ 0 & 0 & 12.3 & 0 & 0 & -38.4 \\ 0 & 14.9 & -0.1 & 0.9 & -5.4 & 0 \end{bmatrix}}_{G^{BP}} \begin{bmatrix} u_I \\ u_{II} \\ u_{III} \\ u_{IV} \\ u_V \\ u_{VI} \end{bmatrix} \tag{6.48}$$

$$\begin{bmatrix} Q_{c1} \\ Q_{c2} \\ Q_{h1} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 2.9 & -0.40 & 0.17 & 0.50 & 0 \\ 0 & 0 & 0.12 & 0 & 0 & 2.1 \\ 6.8 & 0.88 & 0.08 & 0 & 0 & 0 \end{bmatrix}}_{G^{U}} \begin{bmatrix} u_I \\ u_{II} \\ u_{III} \\ u_{IV} \\ u_V \\ u_{VI} \end{bmatrix} \tag{6.49}$$

The zeros in these transfer functions are given by the structural properties of the HEN. In this example, we simply assume that the cost data are equal to unity ($c^T = [1 \quad 1 \quad 1]$), thus we actually minimize the total utility consumption rather than the utility cost. This gives

$$g^{\mathrm{U\$}} = [1 \;\; 1 \;\; 1]G^{\mathrm{U}} = [6.8 \;\; 3.78 \;\; -0.20 \;\; 0.17 \;\; 0.50 \;\; 2.1] \tag{6.50}$$

Defining $e_1 = T_{\mathrm{o}}^{\mathrm{H1}} - T_{\mathrm{t}}^{\mathrm{H1}}$, $e_2 = T_{\mathrm{o}}^{\mathrm{C2}} - T_{\mathrm{t}}^{\mathrm{C2}}$ and so on, the priority matrix ($p_{i,j} = -\frac{g_j^{\mathrm{U\$}}}{g_{i,j}^{\mathrm{BP}}} e_i$) becomes

$$P = \begin{bmatrix} -0.12e_1 & 0.28e_1 & -0.16e_1 & \times & \times & \times \\ \times & -0.25e_2 & -0.10e_2 & 0.05e_2 & \times & \times \\ \times & \times & 0.02e_3 & \times & \times & 0.05e_3 \\ \times & -0.25e_4 & -2.0e_4 & -0.19e_4 & 0.09e_4 & \times \end{bmatrix} \tag{6.51}$$

For this example, there are 2 excess manipulations thus 2 inputs are unused. This yields 15 combinations of using 4 of the 6 bypasses. The RGAs for these 15 possible candidates are listed below. The columns with the unused inputs are marked with × instead of 0 such that they are easily identified and the zeros in the other columns are not filled in.

$$RGA^{1,2} = \begin{bmatrix} \times & \times & 1 & & \\ \times & \times & & 1 & \\ \times & \times & & & 1 \\ \times & \times & & 1 & \end{bmatrix} \quad RGA^{1,3} = \begin{bmatrix} \times & 1 & \times & & \\ \times & & \times & 1 & \\ \times & & \times & & 1 \\ \times & & \times & 1 & \end{bmatrix} \quad RGA^{1,4} = \begin{bmatrix} \times & 0.6 & 0.4 & \times & \\ \times & 0.4 & 0.6 & \times & \\ \times & & & \times & 1 \\ \times & & & \times & 1 \end{bmatrix}$$

$$RGA^{1,5} = \begin{bmatrix} \times & 0.27 & 0.73 & & \times \\ \times & 0.15 & 0.23 & 0.62 & \times \\ \times & & & & \times \; 1 \\ \times & 0.58 & 0.04 & 0.38 & \times \end{bmatrix} \quad RGA^{1,6} = \begin{bmatrix} \times & 1 & & & \times \\ \times & & & 1 & \times \\ \times & & 1 & & \times \\ \times & & & 1 & \times \end{bmatrix} \quad RGA^{2,3} = \begin{bmatrix} 1 & \times & \times & & \\ & \times & \times & 1 & \\ & \times & \times & & 1 \\ & \times & \times & 1 & \end{bmatrix}$$

$$RGA^{2,4} = \begin{bmatrix} 1 & \times & & \times & \\ & \times & 1 & \times & \\ & \times & & \times & 1 \\ & \times & & \times & 1 \end{bmatrix} \quad RGA^{2,5} = \begin{bmatrix} 1 & \times & & & \times \\ & \times & 0.84 & 0.16 & \times \\ & \times & & & \times \; 1 \\ & \times & 0.16 & 0.84 & \times \end{bmatrix} \quad RGA^{2,6} = \begin{bmatrix} 1 & \times & & & \times \\ & \times & & 1 & \times \\ & \times & 1 & & \times \\ & \times & & 1 & \times \end{bmatrix}$$

$$RGA^{3,4} = \begin{bmatrix} 1 & & \times & \times & \\ & 1 & \times & \times & \\ & & \times & \times & 1 \\ & & \times & \times & 1 \end{bmatrix} \quad RGA^{3,5} = \begin{bmatrix} 1 & & \times & & \times \\ & 0.2 & \times & 0.8 & \times \\ & & \times & & \times \; 1 \\ & 0.8 & \times & 0.2 & \times \end{bmatrix} \quad RGA^{3,6} = \text{Infeasible}$$

$$RGA^{4,5} = \begin{bmatrix} 1 & & & \times & \times \\ & -0.05 & 1.05 & \times & \times \\ & & & \times & \times \; 1 \\ & 1.05 & -0.05 & \times & \times \end{bmatrix} \quad RGA^{4,6} = \begin{bmatrix} 1 & & \times & & \times \\ & 1 & & \times & \times \\ & & 1 & \times & \times \\ & & & \times \; 1 & \times \end{bmatrix} \quad RGA^{5,6} = \begin{bmatrix} 1 & & & \times & \times \\ & 0.2 & & 0.8 & \times \; \times \\ & & 1 & & \times \; \times \\ & 0.8 & & 0.2 & \times \; \times \end{bmatrix}$$

The RGAs can be divided into the following three groups:

8 RGAs are equal to a permutation of I, i.e. $RGA = X$. (only 1 pairing is possible).
6 RGAs are not equal to X, and only 1 of these has negative relative gains.
1 RGA ($RGA^{3,6}$) is infeasible. (With inputs u_{III} and u_{VI} unused, no input affects T_o^{C3}).

For the first group of 8 RGAs equal to X, the rows and columns in G^{BP} (disregarding columns that correspond to unused inputs) may be ordered to give a triangular matrix. For each of these 8 possible selections, only one pairing is feasible. Further, these 8 pairings give an RGA-number of zero. That is, the second controllability requirement (minimize RGA-number) is useless to distinguish between these 8 candidates.

Among the 6 in the second group, 5 of these have only 2 possible pairings while 1 RGA ($RGA^{1,5}$) yields 4 possible pairings. This yields a total of 14 possible pairings for this group. Only 1 pairing has negative relative gains and this is the pairing $x_{1,1} = x_{2,2} = x_{4,3} = x_{3,6} = 1$ (see $RGA^{4,5}$). With this information, model C may be significantly simplified:

First, the constraints (6.36) in model B will be replaced by a simpler expression. To force α to zero when the pairing with negative gains is selected, (6.36) can be replaced by

$$\alpha - 4 + (x_{1,1} + x_{2,2} + x_{4,3} + x_{3,6}) \leq 0 \tag{6.52}$$

If another pairing is selected (one or more of the x-elements in (6.52) are 0), then α is free. It is not required to force α to 1 if another pairing is selected since the objective function in model C handles this by penalizing $\alpha = 0$.

The other simplification is more significant. For this example with 4 bypass controlled temperatures and 6 manipulations each of the constraints in (6.45a) to (6.45f) actually represents 24 single constraints. In total, the constraint (6.45) includes 145 single constraints. These will now be replaced by a considerably smaller number of constraints.

As mentioned above, all 8 RGAs in the first group have RGA-number = 0. The 14 possible pairings from the second group imply that there are only 14 RGA-numbers different from zero. That is, all RGA-numbers may be computed *in advance* and each RGA-number is associated with the corresponding pairing. No pairing can have RGA-number less than zero, and the following constraint is included:

$$CM \geq 0 \tag{6.53}$$

This constraint handles all 8 pairings in the first group. The infeasibility when inputs 3 and 6 are unused do not need any further consideration as this pairing will violate other constraints in the model (constraints 6.13b and 6.13f in model A). For the 14 possible pairings (in the second group) that have a positive RGA-number, we consider the 2 possible pairings in $RGA^{1,4}$ as an example. The 2 possible pairings are $x_{1,2} = x_{2,3} = x_{3,6} = x_{4,5} = 1$ and $x_{1,3} = x_{2,2} = x_{3,6} = x_{4,5} = 1$. For the first of these pairings we have a RGA-number of 1.6. That is, we want a constraint that fulfills:

$$\text{If } x_{1,2} = x_{2,3} = x_{3,6} = x_{4,5} = 1, \text{ then RGA-number } (CM) \geq 1.6$$

We do not need to have $CM = 1.6$ since CM is penalized in the objective. Thus, the optimal value for CM is the smallest being feasible. The following inequality provides the necessary constraint on CM for this pairing.

$$1.6[(x_{1,2} + x_{2,3} + x_{3,6} + x_{4,5}) - 4] + 1.6 - CM \leq 0 \tag{6.54}$$

If all 4 x-elements are 1, then (5.54) is equivalent to $CM \geq 1.6$. If one or more x-elements are zero, then $CM \geq 1.6$ by at least one of the constraints (6.53) and (6.54). For the second possible pairing in $RGA^{1,4}$ the RGA-number is 2.4. To make this RGA-number active when this pairing ($x_{1,3} = x_{2,2} = x_{3,6} = x_{4,5} = 1$) is selected, the following constraint is added:

$$2.4[(x_{1,3} + x_{2,2} + x_{3,6} + x_{4,5}) - 4] + 2.4 - CM \leq 0 \tag{6.55}$$

Similar constraints are added for all 14 pairings in the second group. That is, the 145 constraints in the general formulation of model C (equation 6.45) can be replaced 15 constraints (constraint 6.53 plus 14 constraints similar to 6.54 and 6.55) for this particular example.

From this example, two points have become evident:

- The second controllability requirement may not always be well suited to distinguish between different pairings because many pairings may have RGA-number =0.
- In a given case, it may be possible to significantly simplify the general formulations of model B and C.

As a conclusion to the example, one should always investigate the characteristics of the particular HEN/plant to be operated instead of applying the general models directly. ■

6.7 Conclusions

The chapter has presented a parametric approach for optimal operation of HENs. The approach may be regarded as a quantified version of the sign method discussed in the previous chapter. The idea may also be utilized to extend the sign method to multiple bypass controlled outlet temperatures.

An ILP (or MILP) problem to find the optimal control configuration (pairing) is solved during operation. This optimization can be carried out periodically, or it may be performed when e.g. a new manipulation has saturated such that a new pairing has to be applied. The optimal control configuration is the pairing that gives the largest reduction (or smallest increase) in operating cost after the present control errors have been canceled. That is, for a given process state with given control errors, the method will find which bypasses should be manipulated in order to move the process towards the optimal state. The actual manipulations of the bypass fractions are done by SISO controllers.

It is emphasized here, that interactions have been disregarded in the formulation of the optimization problem. This may impose limitations in using the presented approach on

MIMO systems (i.e. multiple bypass controlled outlet temperatures for HENs). To what extent the method may fail to find the optimal operating state for MIMO plants should be thoroughly investigated. This investigation, probably together with further improvements, are topics for future research that need to be done before the approach can be recommended for practical use on MIMO plants.

The approach comprises three methods that differ from each other in how controllability considerations are implemented. No controllability considerations are made in model A. In model B, pairings resulting in plants that are not DIC are disregarded when there are alternatives giving DIC plants. Model C optimizes operating cost only when DIC plants exits, and also controllability considerations are used to distinguish between pairings with similar steady state properties. If no DIC plants exist, model C focuses solely on finding the pairing with the best controllability properties. Model C represents a major extension compared to the size of model A and model B. An example has shown that model C may be significantly simplified for a practical HEN.

For processes where the optimal operating state lies at the intersection of constraints, the approach presented in this chapter is an alternative to optimizing control schemes such as MPC (model predictive control) or periodic optimization of a steady state process model to find optimal setpoints. When the optimal operating state is at constraints, other strategies may led to computing an infeasible setpoint due to model errors or unknown disturbances. Alternatively, significant safety margins may be imposed in order to avoid infeasibility and thereby the operating state will be moved away from the optimal one. Of course, other strategies may be combined with on-line identification to improve the process model, but this will increase the complexity of the applied method. Since the optimization problem included in the approach presented here results in a control *configuration*, only errors leading to a different configuration will have any influence. Hence, the approach may be expected to be robust to model errors and unknown disturbances. The robustness to errors is, however, not investigated and a suggestion to further work is to verify this.

A disadvantage with the proposed approach is that the resulting control configuration may have bad control properties, e.g. an output may be controlled by a manipulation with only a small and slow effect. This may happen despite the controllability considerations included in two of the methods. Further, the pairing may vary during operation and this may be confusing for the operators, particularly if it should be necessary to operate the plant (or parts of it) manually.

Two of the proposed methods represent hierarchical strategies for operation. It has been demonstrated that such strategies can be implemented within the framework of mathematical programming. Discrete decisions and logic inference required for the hierarchical strategies are formulated as mathematical inequalities combining continuous and binary (logic) variables.

Notation

c	Vector containing utility costs.
CM	Controllability measure (i.e. value of RGA-number).
CP	Heat capacity flowrate.
d	Disturbance input.
e	Control error.
g	Element in transfer matrix.
G	Transfer matrix.
M	Large number (upper bound on continuous variable).
n	Number of . . . (e.g. n_y is the number of controlled outputs).
OC	Operating cost.
P	Priority matrix.
Q	Heat load (duty) of heat exchanger.
RGA	Relative gain array.
$RGAn$	Absolute value of each element in ($RGA_{i,j} - x_{i,j}$).
T	Temperature.
u	Manipulated input (bypass fraction).
X	Binary variable ($x_{i,j} = 1$ for selected pairing, $x_{i,j} = 0$ for not selected pairing).
y	Binary constant ($y_{i,j} = 1$ for pairing requiring reduction of u_j, otherwise $y_{i,j} = 0$.
z	Binary constant to denote if u is saturated at upper or lower bound.

Greek:

α	Global binary variable to denote whether selected pairing is DIC.
β	Set of binary constants to denote if the corresponding RGA-element is negative.
δ	Upper bound on negative values of RGA-elements.
ε	Detuning factor.
γ	Logical variable for computation of RGA-number.

Superscripts:

BP	Bypass.
k	Index for unused input.
lo	Lower.
up	Upper.
U	Utility (concerns utility consumption in each utility exchanger).
U$	Utility cost (total utility cost in money per time unit).

Subscripts:

max	Maximum.
o	Actual output or outlet (temperature).
s	Supply (temperature).
t	Target or reference (temperature).
y	Controlled outputs.

References

Biegler, L.T., I.E. Grossmann and A.W. Westerberg (1997*). Systematic Methods for Chemical Process Design*. Prentice-Hall Inc. Upper Saddle River, NJ.

Grosdidier, P., M. Morari and B.R. Holt (1985). Closed-loop Properties from Steady state Gain Information. *Ind. Eng. Chem. Fund.*, **24**, 221-235.

Linnhoff, B. *et al*. (1982). *User Guide on Process Integration for the Efficient Use of Energy*. Inst. Chem. Engrs., Rugby, UK.

Mathisen, K.W. (1994). Integrated Design and Control of Heat Exchanger Networks. *Ph.D. Thesis*. University of Trondheim - NTH, Norway.

Skogestad, S. and M. Morari (1988). Variable Selection for Decentralized Control. *AIChE Annual Meeting*, Washington DC. Paper 126f. Reprinted in *Modeling, Identification and Control*, 1992, **13** No. 2, 113

Skogestad, S. and I. Postlethwaite (1996*). Multivariable Feedback Control, Analysis and Design*. John Wiley & Sons, Chichester, UK.