quantify the degree of directionality and the level of (two-way) interactions in MIMO systems are the condition number and the relative gain array (RGA), respectively. We £rst consider the *condition number* of a matrix which is de£ned as the ratio between the maximum and minimum singular values,

$$\gamma(G) \overset{\triangle}{=} \bar{\sigma}(G)/\underline{\sigma}(G) \tag{3.52}$$

A matrix with a large condition number is said to be *ill-conditioned*. For a non-singular (square) matrix $\underline{\sigma}(G) = 1/\bar{\sigma}(G^{-1})$, so $\gamma(G) = \bar{\sigma}(G)\bar{\sigma}(G^{-1})$. It then follows from (A.120) that the condition number is large if both G and G^{-1} have large elements.

The condition number depends strongly on the scaling of the inputs and outputs. To be more speci£c, if D_1 and D_2 are diagonal scaling matrices, then the condition numbers of the matrices G and D_1GD_2 may be arbitrarily far apart. In general, the matrix G should be scaled on physical grounds, e.g. by dividing each input and output by its largest expected or desired value as discussed in Section 1.4.

One might also consider minimizing the condition number over all possible scalings. This results in the *minimized or optimal condition number* which is de£ned by

$$\gamma^*(G) = \min_{D_1,D_2} \gamma(D_1GD_2) \tag{3.53}$$

and can be computed using (A.74).

The condition number has been used as an input–output controllability measure, and in particular it has been postulated that a large condition number indicates sensitivity to uncertainty. This is not true in general, but the reverse holds: if the condition number is small, then the multivariable effects of uncertainty are not likely to be serious (see (6.89)).

If the condition number is large (say, larger than 10), then this may *indicate* control problems:

1. A large condition number $\gamma(G) = \bar{\sigma}(G)/\underline{\sigma}(G)$ may be caused by a small value of $\underline{\sigma}(G)$, which is generally undesirable (on the other hand, a large value of $\bar{\sigma}(G)$ need not necessarily be a problem).
2. A large condition number may mean that the plant has a large minimized condition number, or equivalently, it has large RGA elements which indicate fundamental control problems; see below.
3. A large condition number *does* imply that the system is sensitive to "unstructured" (full-block) input uncertainty (e.g. with an inverse-based controller, see (8.136)), but this kind of uncertainty often does not occur in practice. We therefore *cannot* generally conclude that a plant with a large condition number is sensitive to uncertainty, e.g. see the diagonal plant in Example 3.12 (page 89).

3.4 Relative gain array (RGA)

The RGA (Bristol, 1966) of a non-singular square complex matrix G is a square complex matrix de£ned as

$$\text{RGA}(G) = \Lambda(G) \triangleq G \times (G^{-1})^T \tag{3.54}$$

where × denotes element-by-element multiplication (the Hadamard or Schur product). With Matlab, we write[5]

```
RGA = G.*pinv(G).'
```

The RGA of a transfer matrix is generally computed as a function of frequency (see Matlab program in Table 3.1). For a 2×2 matrix with elements g_{ij} the RGA is

$$\Lambda(G) = \begin{bmatrix} \lambda_{11} & \lambda_{12} \\ \lambda_{21} & \lambda_{22} \end{bmatrix} = \begin{bmatrix} \lambda_{11} & 1-\lambda_{11} \\ 1-\lambda_{11} & \lambda_{11} \end{bmatrix}; \quad \lambda_{11} = \frac{1}{1 - \frac{g_{12}g_{21}}{g_{11}g_{22}}} \tag{3.55}$$

The RGA is a very useful tool in practical applications. The RGA is treated in detail at three places in this book. First, we give a general introduction in this section (pages 82–91). The use of the RGA for decentralized control is discussed in more detail in Section 10.6 (pages 442–454). Finally, its algebraic properties and extension to non-square matrices are considered in Appendix A.4 (pages 526–529).

3.4.1 Original interpretation: RGA as an interaction measure

We follow Bristol (1966) here, and show that the RGA provides a measure of interactions. Let u_j and y_i denote a particular input–output pair for the multivariable plant $G(s)$, and assume that our task is to use u_j to control y_i. Bristol argued that there will be two extreme cases:

- All other loops open: $u_k = 0, \forall k \neq j$.
- All other loops closed with perfect control: $y_k = 0, \forall k \neq i$.

Perfect control is only possible at steady-state, but it is a good approximation at frequencies within the bandwidth of each loop. We now evaluate "our" gain $\partial y_i/\partial u_j$ for the two extreme cases:

$$\text{Other loops open:} \quad \left(\frac{\partial y_i}{\partial u_j}\right)_{u_k=0, k\neq j} = g_{ij} \tag{3.56}$$

$$\text{Other loops closed:} \quad \left(\frac{\partial y_i}{\partial u_j}\right)_{y_k=0, k\neq i} \triangleq \widehat{g}_{ij} \tag{3.57}$$

Here $g_{ij} = [G]_{ij}$ is the ij'th element of G, whereas $\widehat{g}_{ij}$ is the inverse of the ji'th element of G^{-1}

$$\widehat{g}_{ij} = 1/[G^{-1}]_{ji} \tag{3.58}$$

To derive (3.58) we note that

$$y = Gu \quad \Rightarrow \quad \left(\frac{\partial y_i}{\partial u_j}\right)_{u_k=0, k\neq j} = [G]_{ij} \tag{3.59}$$

and interchange the roles of G and G^{-1}, of u and y, and of i and j to get

$$u = G^{-1}y \quad \Rightarrow \quad \left(\frac{\partial u_j}{\partial y_i}\right)_{y_k=0, k\neq i} = [G^{-1}]_{ji} \tag{3.60}$$

[5] The symbol `'` in Matlab gives the conjugate transpose (A^H), and we must use `.'` to get the "regular" transpose (A^T).

and (3.58) follows. Bristol argued that the ratio between the gains in (3.56) and (3.57) is a useful measure of interactions, and de£ned the ij'th "relative gain" as

$$\lambda_{ij} \triangleq \frac{g_{ij}}{\widehat{g}_{ij}} = [G]_{ij}[G^{-1}]_{ji} \tag{3.61}$$

The RGA is the corresponding matrix of relative gains. From (3.61) we see that $\Lambda(G) = G \times (G^{-1})^T$ where × denotes element-by-element multiplication (the Schur product). This is identical to our de£nition of the RGA matrix in (3.54).

Remark. The assumption of $y_k = 0$ ("perfect control of y_k") in (3.57) is satis£ed at steady-state ($\omega = 0$) provided we have integral action in the loop, but it will generally not hold exactly at other frequencies. Unfortunately, this has led many authors to dismiss the RGA as being "only useful at steady-state" or "only useful if we use integral action". On the contrary, in most cases it is the value of the RGA at frequencies close to crossover which is most important, and both the gain and the phase of the RGA elements are important. The derivation of the RGA in (3.56) to (3.61) was included to illustrate one useful interpretation of the RGA, but note that our de£nition of the RGA in (3.54) is purely algebraic and makes no assumption about "perfect control". The general usefulness of the RGA is further demonstrated by the additional general algebraic and control properties of the RGA listed on page 88.

Example 3.8 **RGA for 2×2 system.** *Consider a 2×2 system with the plant model*

$$y_1 = g_{11}(s)u_1 + g_{12}(s)u_2 \tag{3.62}$$

$$y_2 = g_{21}(s)u_1 + g_{22}(s)u_2 \tag{3.63}$$

Assume that "our" task is to use u_1 to control y_1. First consider the case when the other loop is open, i.e. u_2 is constant or equivalently $u_2 = 0$ in terms of deviation variables. We then have

$$u_2 = 0: \quad y_1 = g_{11}(s)u_1$$

Next consider the case when the other loop is closed with perfect control, i.e. $y_2 = 0$. In this case, u_2 will also change when we change u_1, due to interactions. More precisely, setting $y_2 = 0$ in (3.63) gives

$$u_2 = -\frac{g_{21}(s)}{g_{22}(s)}u_1$$

Substituting this into (3.62) gives

$$y_2 = 0: \quad y_1 = \underbrace{\left(g_{11} - \frac{g_{21}}{g_{22}}g_{12}\right)}_{\widehat{g}_{11}(s)} u_1$$

This means that "our gain" changes from $g_{11}(s)$ to $\widehat{g}_{11}(s)$ as we close the other loop, and the corresponding RGA element becomes

$$\lambda_{11}(s) = \frac{\text{"open-loop gain (with } u_2 = 0)\text{"}}{\text{"closed-loop gain (with } y_2 = 0)\text{"}} = \frac{g_{11}(s)}{\widehat{g}_{11}(s)} = \frac{1}{1 - \frac{g_{12}(s)g_{21}(s)}{g_{11}(s)g_{22}(s)}}$$

Intuitively, for decentralized control, we *prefer* to pair variables u_j and y_i so that λ_{ij} is close to 1 at all frequencies, because this means that the gain from u_j to y_i is unaffected by closing the other loops. More precisely, we have:

> **Pairing rule 1** (page 450): *Prefer pairings such that the rearranged system, with the selected pairings along the diagonal, has an RGA matrix close to identity at frequencies around the closed-loop bandwidth.*

However, one should avoid pairings where the sign of the steady-state gain from u_j to y_i may change depending on the control of the other outputs, because this will yield instability with integral action in the loop. Thus, $g_{ij}(0)$ and $\widehat{g}_{11}(0)$ should have the same sign, and we have:

> **Pairing rule 2** (page 450): *Avoid (if possible) pairing on negative steady-state RGA elements.*

The reader is referred to Section 10.6.4 (page 438) for derivation and further discussion of these pairing rules.

3.4.2 Examples: RGA

Example 3.9 **Blending process.** *Consider a blending process where we mix sugar (u_1) and water (u_2) to make a given amount ($y_1 = F$) of a soft drink with a given sugar fraction ($y_2 = x$). The balances "mass in = mass out" for total mass and sugar mass are*

$$F_1 + F_2 = F$$

$$F_1 = xF$$

Note that the process itself has no dynamics. Linearization yields

$$dF_1 + dF_2 = dF$$

$$dF_1 = x^* dF + F^* dx$$

With $u_1 = dF_1, u_2 = dF_2, y_1 = dF$ and $y_2 = dx$ we then get the model

$$y_1 = u_1 + u_2$$

$$y_2 = \frac{1-x^*}{F^*}u_1 - \frac{x^*}{F^*}u_2$$

where $x^* = 0.2$ is the nominal steady-state sugar fraction and $F^* = 2$ kg/s is the nominal amount. The transfer matrix then becomes

$$G(s) = \begin{bmatrix} 1 & 1 \\ \frac{1-x^*}{F^*} & -\frac{x^*}{F^*} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0.4 & -0.1 \end{bmatrix}$$

and the corresponding RGA matrix is (at all frequencies)

$$\Lambda = \begin{bmatrix} x^* & 1-x^* \\ 1-x^* & x^* \end{bmatrix} = \begin{bmatrix} 0.2 & 0.8 \\ 0.8 & 0.2 \end{bmatrix}$$

For decentralized control, it then follows from pairing rule 1 ("prefer pairing on RGA elements close to 1") that we should pair on the off-diagonal elements; that is, use u_1 to control y_2 and use u_2 to control y_1. This corresponds to using the largest stream (water, u_2) to control the amount ($y_1 = F$), which is reasonable from a physical point of view. Pairing rule 2 is also satis£ed for this choice.

Example 3.10 Steady-state RGA. *Consider a 3×3 plant for which we have at steady-state*

$$G = \begin{bmatrix} 16.8 & 30.5 & 4.30 \\ -16.7 & 31.0 & -1.41 \\ 1.27 & 54.1 & 5.40 \end{bmatrix}, \ \Lambda(G) = \begin{bmatrix} 1.50 & 0.99 & -1.48 \\ -0.41 & 0.97 & 0.45 \\ -0.08 & -0.95 & 2.03 \end{bmatrix} \tag{3.64}$$

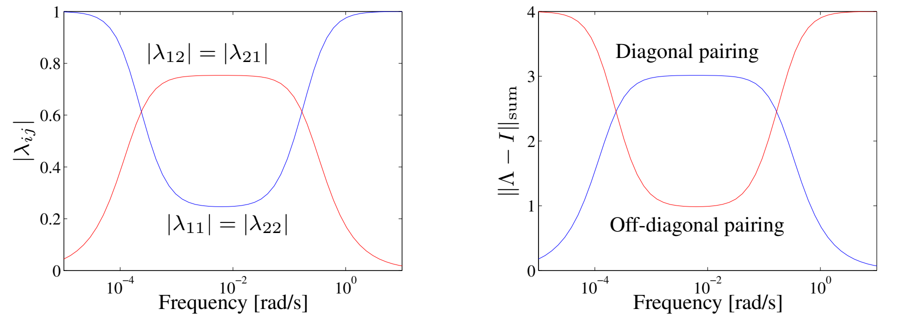
For decentralized control, we need to pair on one element in each column or row. It is then clear that the only choice that satis£es pairing rule 2 ("avoid pairing on negative RGA elements") is to pair on the diagonal elements; that is, use u_1 to control y_1, u_2 to control y_2 and u_3 to control y_3.

Remark. *The plant in (3.64) represents the steady-state model of a ¤uid catalytic cracking (FCC) process. A dynamic model of the FCC process in (3.64) is given in Exercise 6.17 (page 257).*

Some additional examples and exercises, that further illustrate the effectiveness of the steady-state RGA for selecting pairings, are given on page 443.

Example 3.11 Frequency-dependent RGA. *The following model describes a a large pressurized vessel (Skogestad and Wolff, 1991), for example, of the kind found in offshore oil-gas separations. The inputs are the valve positions for liquid (u_1) and vapour (u_2) ¤ow, and the outputs are the liquid volume (y_1) and pressure (y_2).*

$$G(s) = \frac{0.01e^{-5s}}{(s + 1.72 \cdot 10^{-4})(4.32s + 1)} \begin{bmatrix} -34.54(s + 0.0572) & 1.913 \\ -30.22s & -9.188(s + 6.95 \cdot 10^{-4}) \end{bmatrix} \tag{3.65}$$



(a) Magnitude of RGA elements

(b) RGA number

Figure 3.8: Frequency-dependent RGA for $G(s)$ in (3.65)

The RGA matrix $\Lambda(s)$ depends on frequency. At steady-state ($s = 0$) the 2,1 element of $G(s)$ is zero, so $\Lambda(0) = I$. Similarly, at high frequencies the 1,2 element is small relative to the other elements, so $\Lambda(j\infty) = I$. This seems to suggest that the diagonal pairing should be used. However, at intermediate frequencies, the off-diagonal RGA elements are closest to 1, see Figure 3.8(a). For example, at frequency $\omega = 0.01$ rad/s the RGA matrix becomes (see Table 3.1)

$$\Lambda = \begin{bmatrix} 0.2469 + 0.0193i & 0.7531 - 0.0193i \\ 0.7531 - 0.0193i & 0.2469 + 0.0193i \end{bmatrix} \tag{3.66}$$

Thus, from pairing rule 1, the reverse pairings is probably best if we use decentralized control and the closed-loop bandwidth is around 0.01 rad/s. From a physical point of view the use of the reverse pairings is quite surprising, because it involves using the vapour ¤ow (u_2) to control liquid level (y_1). and the liquid ¤ow (u_1) to control pressure (y_2).

Table 3.1: **Matlab program to calculate frequency-dependent RGA**

```
% Plant model (3.65)
s = tf('s');
G = (0.01/(s+1.72e-4)/(4.32*s + 1))*[-34.54*(s+0.0572),....
omega = logspace(-5,2,61);
% RGA
for i = 1:length(omega)
    Gf = freqresp(G,omega(i));                             % G(jω)
    RGAw(:,:,i) = Gf.*inv(Gf).';                           % RGA at frequency omega
    RGAno(i) = sum(sum(abs(RGAw(:,:,i) - eye(2))));        % RGA number
end
RGA = frd(RGAw,omega);
```

Remark. *Although it is possible to use decentralized control for this interactive process, see the following exercise, one may achieve much better performance with multivariable control. If one insists on using decentralized control, then it is recommended to add a liquid flow measurement and use an "inner" (lower layer) flow controller. The resulting u_1 is then the liquid flow rate rather than the valve position. Then u_2 (vapour flow) has no effect on y_1 (liquid volume), and the plant is triangular with $g_{12} = 0$. In this case the diagonal pairing is clearly best.*

Exercise 3.7* *Design decentralized single-loop controllers for the plant (3.65) using (a) the diagonal pairings and (b) the off-diagonal pairings. Use the delay θ (which is nominally 5 seconds) as a parameter. Use PI controllers independently tuned with the SIMC tuning rules (based on the paired elements).*

Outline of solution: For tuning purposes the elements in $G(s)$ are approximated using the half rule to get

$$G(s) \approx \begin{bmatrix} -0.0823\frac{e^{-\theta s}}{s} & 0.01913\frac{e^{-(\theta+2.16)s}}{s} \\ -0.3022\frac{e^{-\theta s}}{4.32s+1} & -0.09188\frac{e^{-\theta s}}{4.32s+1} \end{bmatrix}$$

For the diagonal pairings this gives the PI settings

$$K_{c1} = -12.1/(\tau_{c1}+\theta), \tau_{I1} = 4(\tau_{c1}+\theta); K_{c2} = -47.0/(\tau_{c2}+\theta), \tau_{I2} = 4.32$$

and for the off-diagonal pairings (the index refers to the output)

$$K_{c1} = 52.3/(\tau_{c1}+\theta+2.16), \tau_{I1} = 4(\tau_{c1}+\theta+2.16); K_{c2} = -14.3/(\tau_{c2}+\theta), \tau_{I2} = 4.32$$

For improved robustness, the level controller (y_1) is tuned about 3 times slower than the pressure controller (y_2), i.e. use $\tau_{c1} = 3\theta$ and $\tau_{c2} = \theta$. This gives a crossover frequency of about $0.5/\theta$ in the fastest loop. With a delay of about 5 s or larger you should find, as expected from the RGA at crossover frequencies (pairing rule 1), that the off-diagonal pairing is best. However, if the delay is decreased from 5 s to 1 s, then the diagonal pairing is best, as expected since the RGA for the diagonal pairing approaches 1 at frequencies above 1 rad/s.

3.4.3 RGA number and iterative RGA

Note that in Figure 3.8(a) we plot only the magnitudes of λ_{ij}, but this may be misleading when selecting pairings. For example, a magnitude of 1 (seemingly a desirable pairing) may correspond to an RGA element of -1 (an undesirable pairing). The phase of the RGA elements should therefore also be considered. An alternative is to compute the RGA number, as defined next.

RGA number. A simple measure for selecting pairings according to rule 1 is to prefer pairings with a small *RGA number*. For a diagonal pairing,

$$\text{RGA number} \triangleq \|\Lambda(G) - I\|_{\text{sum}} \qquad (3.67)$$

where we have (somewhat arbitrarily) chosen the sum norm, $\|A\|_{\rm sum} = \sum_{i,j} |a_{ij}|$. The RGA number for other pairings is obtained by subtracting 1 for the selected pairings; for example, $\Lambda(G) - \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ for the off-diagonal pairing for a 2 × 2 plant. The disadvantage with the RGA number, at least for larger systems, is that it needs to be recomputed for each alternative pairing. On the other hand, the RGA elements need to be computed only once.

Example 3.11 continued. *The RGA number for the plant $G(s)$ in (3.65) is plotted for the two alternative pairings in Figure 3.8(b). As expected, we see that the off-diagonal pairing is preferred at intermediate frequencies.*

Exercise 3.8 *Compute the RGA number for the six alternate pairings for the plant in (3.64). Which pairing would you prefer?*

Remark. Diagonal dominance. A more precise statement of pairing rule 1 (page 85) would be to prefer pairings that have "diagonal dominance" (see de£nition on page 10.6.4). There is a close relationship between a small RGA number and diagonal dominance, but unfortunately there are exceptions for plants of size 4 × 4 or larger, so a small RGA number does not always guarantee diagonal dominance; see Example 10.18 on page 441.

Iterative RGA. An iterative evaluation of the RGA, $\Lambda^2(G) = \Lambda(\Lambda(G))$ etc., is very useful for choosing pairings with diagonal dominance for large systems. Wolff (1994) found numerically that

$$\Lambda^\infty \triangleq \lim_{k\to\infty} \Lambda^k(G) \tag{3.68}$$

is a permuted identity matrix (except for "borderline" cases). More importantly, Johnson and Shapiro (1986, Theorem 2) have proven that Λ^∞ *always converges to the identity matrix if G is a generalized diagonally dominant matrix* (see de£nition in Remark 10.6.4 on page 440) . Since permuting the matrix G causes similar permutations of $\Lambda(G)$, Λ^∞ may then be used as a candidate pairing choice. Typically, Λ^k approaches Λ^∞ for k between 4 and 8. For example, for $G = \begin{bmatrix} 1 & 2 \\ -1 & 1 \end{bmatrix}$ we get $\Lambda = \begin{bmatrix} 0.33 & 0.67 \\ 0.67 & 0.33 \end{bmatrix}$, $\Lambda^2 = \begin{bmatrix} -0.33 & 1.33 \\ 1.33 & -0.33 \end{bmatrix}$, $\Lambda^3 = \begin{bmatrix} -0.07 & 1.07 \\ 1.07 & -0.07 \end{bmatrix}$ and $\Lambda^4 = \begin{bmatrix} 0.00 & 1.00 \\ 1.00 & 0.00 \end{bmatrix}$, which indicates that the off-diagonal pairing is diagonally dominant. Note that Λ^∞ may sometimes "recommend" a pairing on negative RGA elements, even if a positive pairing is possible.

Exercise 3.9 *Test the iterative RGA method on the plant (3.64) and con£rm that it gives the diagonally dominant pairing (as it should according to the theory).*

3.4.4 Summary of algebraic properties of the RGA

The (complex) RGA matrix has a number of interesting *algebraic properties*, of which the most important are (see Appendix A.4, page 526, for more details):

A1. It is independent of input and output scaling.
A2. Its rows and columns sum to 1.
A3. The RGA is the identity matrix if G is upper or lower triangular.
A4. A relative change in an element of G equal to the negative inverse of its corresponding RGA element, $g'_{ij} = g_{ij}(1 - 1/\lambda_{ij})$, yields singularity.

A5. From (A.80), plants with large RGA elements are always ill-conditioned (with a large value of $\gamma(G)$), but the reverse may not hold (i.e. a plant with a large $\gamma(G)$ may have small RGA elements).

From property A3, it follows that the RGA (or more precisely $\Lambda - I$) provides a measure of *two-way interaction*.

Example 3.12 *Consider a diagonal plant for which we have*

$$G = \begin{bmatrix} 100 & 0 \\ 0 & 1 \end{bmatrix}, \; \Lambda(G) = I, \; \gamma(G) = \frac{\bar{\sigma}(G)}{\underline{\sigma}(G)} = \frac{100}{1} = 100, \; \gamma^*(G) = 1 \qquad (3.69)$$

Here the condition number is 100 *which means that the plant gain depends strongly on the input direction. However, since the plant is diagonal there are no interactions so* $\Lambda(G) = I$ *and the minimized condition number* $\gamma^*(G) = 1$.

Example 3.13 *Consider a triangular plant G for which we get*

$$G = \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix}, \; G^{-1} = \begin{bmatrix} 1 & -2 \\ 0 & 1 \end{bmatrix}, \; \Lambda(G) = I, \; \gamma(G) = \frac{2.41}{0.41} = 5.83, \; \gamma^*(G) = 1 \qquad (3.70)$$

Note that for a triangular matrix, there is one-way interaction, but no two-way interaction, and the RGA is always the identity matrix.

Example 3.14 *Consider again the distillation process in (3.45) for which we have at steady-state*

$$G = \begin{bmatrix} 87.8 & -86.4 \\ 108.2 & -109.6 \end{bmatrix}, \; G^{-1} = \begin{bmatrix} 0.399 & -0.315 \\ 0.394 & -0.320 \end{bmatrix}, \; \Lambda(G) = \begin{bmatrix} 35.1 & -34.1 \\ -34.1 & 35.1 \end{bmatrix} \qquad (3.71)$$

In this case $\gamma(G) = 197.2/1.391 = 141.7$ *is only slightly larger than* $\gamma^*(G) = 138.268$. *The magnitude sum of the elements in the RGA matrix is* $\|\Lambda\|_{\rm sum} = 138.275$. *This confirms property A5 which states that, for* 2×2 *systems,* $\|\Lambda(G)\|_{\rm sum} \approx \gamma^*(G)$ *when* $\gamma^*(G)$ *is large. The condition number is large, but since the minimum singular value* $\underline{\sigma}(G) = 1.391$ *is larger than* 1 *this does not by itself imply a control problem. However, the large RGA elements indicate problems, as discussed below (control property C1).*

Example 3.15 *Consider again the FCC process in (3.64) with* $\gamma = 69.6/1.63 = 42.6$ *and* $\gamma^* = 7.80$. *The magnitude sum of the elements in the RGA is* $\|\Lambda\|_{\rm sum} = 8.86$ *which is close to* γ^* *as expected from property A5. Note that the rows and the columns of* Λ *in (3.64) sums to* 1. *Since* $\underline{\sigma}(G)$ *is larger than* 1 *and the RGA elements are relatively small, this steady-state analysis does not indicate any particular control problems for the plant.*

3.4.5 Summary of control properties of the RGA

In addition to the algebraic properties listed above, the RGA has a surprising number of useful *control properties*:

C1. *Large RGA elements (typically,* $5 - 10$ *or larger) at frequencies important for control indicate that the plant is fundamentally difficult to control due to strong interactions and sensitivity to uncertainty.*

(a) *Uncertainty in the input channels (diagonal input uncertainty).* Plants with large RGA elements (at crossover frequency) are fundamentally difficult to control because of sensitivity to input uncertainty, e.g. caused by uncertain or neglected actuator dynamics. In particular, decouplers or other inverse-based controllers should not be used for plants with large RGA elements (see page 251).

(b) *Element uncertainty.* As implied by algebraic property A4 above, large RGA elements imply sensitivity to element-by-element uncertainty. However, this kind of uncertainty may not occur in practice due to physical couplings between the transfer function elements. Therefore, diagonal input uncertainty (which is always present) is usually of more concern for plants with large RGA elements.

C2. *RGA and RHP-zeros.* If the sign of an RGA element changes as we go from $s = 0$ to $s = \infty$, then there is a RHP-zero in G or in some subsystem of G (see Theorem 10.7, page 446).

C3. *Non-square plants*. The de£nition of the RGA may be generalized to non-square matrices by using the pseudo-inverse; see Appendix A.4.2. Extra inputs: If the sum of the elements in a column of RGA is small ($\ll 1$), then one may consider deleting the corresponding input. Extra outputs: If all elements in a row of RGA are small ($\ll 1$), then the corresponding output cannot be controlled.

C4. *RGA and decentralized control.* The usefulness of the RGA is summarized by the two pairing rules on page 85.

Example 3.14 continued. *For the steady-state distillation model in (3.71), the large RGA element of* 35.1 *indicates a control problem. More precisely, fundamental control problems are expected if analysis shows that* $G(j\omega)$ *has large RGA elements also in the crossover frequency range. Indeed, with the idealized dynamic model (3.93) used below, the RGA elements are large at all frequencies, and we will con£rm in simulations that there is a strong sensitivity to input channel uncertainty with an inverse-based controller, see page 100. For decentralized control, we should, according to rule 2, avoid pairing on the negative RGA elements. Thus, the diagonal pairing is preferred.*

Example 3.16 *Consider the plant*

$$G(s) = \frac{1}{5s+1}\begin{pmatrix} s+1 & s+4 \\ 1 & 2 \end{pmatrix} \tag{3.72}$$

We £nd that $\lambda_{11}(\infty) = 2$ *and* $\lambda_{11}(0) = -1$ *have different signs. Since none of the diagonal elements have RHP-zeros we conclude from property C2 that* $G(s)$ *must have a RHP-zero. This is indeed true and* $G(s)$ *has a zero at* $s = 2$.

Let us elaborate a bit more on the use of RGA for decentralized control (control property C4). Assume we use decentralized control with integral action in each loop, and want to pair on one or more negative steady-state RGA elements. This may happen because this pairing is preferred for dynamic reasons or because there exists no pairing choice with only positive RGA elements, e.g. see the system in (10.81) on page 444. What will happen? Will the system be unstable? No, not necessarily. We may, for example, tune one loop at a time in a sequential manner (usually starting with the fastest loops), and we will end up with a stable overall system. However, due to the negative RGA element there will be some hidden problem, because the system is not *decentralized integral controllable* (DIC); see page 443. The stability of the overall system then depends on the individual loops being in service. This means that detuning one or more of the individual loops may result in instability for the overall system. Instability may also occur if an input saturates, because the corresponding loop is then effectively out of service. In summary, pairing on negative steady-state RGA elements should be avoided, and if it cannot be avoided then one should make sure that the loops remain in service.

For a detailed analysis of achievable performance of the plant (input–output controllability analysis), one must consider the singular values, as well as the RGA and condition number as functions of frequency. In particular, the crossover frequency range is important. In addition, disturbances and the presence of unstable (RHP) plant poles and zeros must be considered. All these issues are discussed in much more detail in Chapters 5 and 6 where we address achievable performance and input–output controllability analysis for SISO and MIMO plants, respectively.

3.5 Control of multivariable plants

3.5.1 Diagonal controller (decentralized control)

The simplest approach to multivariable controller design is to use a diagonal or block-diagonal controller $K(s)$. This is often referred to as decentralized control. Decentralized control works well if $G(s)$ is close to diagonal, because then the plant to be controlled is essentially a collection of independent sub-plants. However, if the off-diagonal elements in $G(s)$ are large, then the performance with decentralized diagonal control may be poor because no attempt is made to counteract the interactions. There are three basic approaches to the design of decentralized controllers:

- Fully coordinated design
- Independent design
- Sequential design

Decentralized control is discussed in more detail in Chapter 10 on page 429.

3.5.2 Two-step compensator design approach

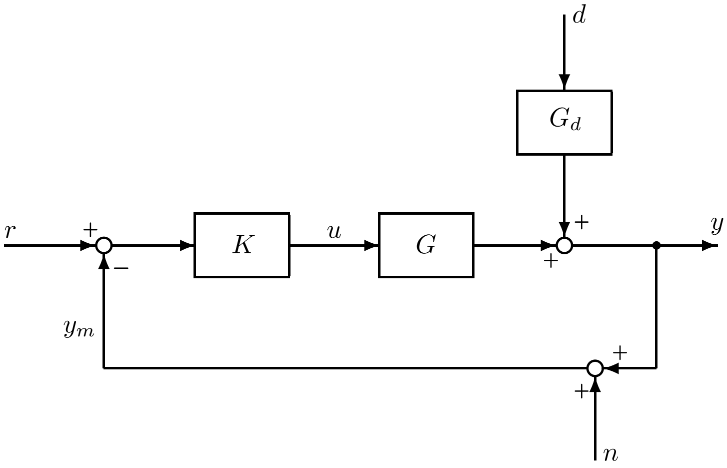


Figure 3.9: One degree-of-freedom feedback control con£guration

Consider the simple feedback system in Figure 3.9. A conceptually simple approach to multivariable control is given by a two-step procedure in which we £rst design a "compensator" to deal with the interactions in G, and then design a *diagonal* controller using methods similar to those for SISO systems in Chapter 2. Several such approaches are discussed below.

The most common approach is to use a pre-compensator, $W_1(s)$, which counteracts the interactions in the plant and results in a "new" shaped plant:

$$G_s(s) = G(s)W_1(s) \tag{3.73}$$

which is more diagonal and easier to control than the original plant $G(s)$. After £nding a suitable $W_1(s)$ we can design a *diagonal* controller $K_s(s)$ for the shaped plant $G_s(s)$. The overall controller is then

$$K(s) = W_1(s)K_s(s) \tag{3.74}$$

In many cases effective compensators may be derived on physical grounds and may include nonlinear elements such as ratios.

Remark 1 Some design approaches in this spirit are the Nyquist array technique of Rosenbrock (1974) and the characteristic loci technique of MacFarlane and Kouvaritakis (1977).

Remark 2 The $\mathcal{H}_\infty$ loop-shaping design procedure, described in detail in Section 9.4, is similar in that a pre-compensator is £rst chosen to yield a shaped plant, $G_s = GW_1$, with desirable properties, and then a controller $K_s(s)$ is designed. The main difference is that in $\mathcal{H}_\infty$ loop shaping, $K_s(s)$ is a full multivariable controller, designed and based on optimization (to optimize $\mathcal{H}_\infty$ robust stability).

3.5.3 Decoupling

Decoupling control results when the compensator W_1 is chosen such that $G_s = GW_1$ in (3.73) is diagonal at a selected frequency. The following different cases are possible:

1. **Dynamic decoupling:** $G_s(s)$ is diagonal at all frequencies. For example, with $G_s(s) = I$ and a square plant, we get $W_1 = G^{-1}(s)$ (disregarding the possible problems involved in realizing $G^{-1}(s)$). If we then select $K_s(s) = l(s)I$ (e.g. with $l(s) = k/s$), the overall controller is

$$K(s) = K_{\text{inv}}(s) \triangleq l(s)G^{-1}(s) \tag{3.75}$$

   We will later refer to (3.75) as an *inverse-based* controller. It results in a decoupled nominal system with identical loops, i.e. $L(s) = l(s)I$, $S(s) = \frac{1}{1+l(s)}I$ and $T(s) = \frac{l(s)}{1+l(s)}I$.

   Remark. In some cases we may want to keep the diagonal elements in the shaped plant unchanged by selecting $W_1 = G^{-1}G_{diag}$. In other cases we may want the diagonal elements in W_1 to be 1. This may be obtained by selecting $W_1 = G^{-1}((G^{-1})_{diag})^{-1}$, and the off-diagonal elements of W_1 are then called "decoupling elements".

2. **Steady-state decoupling:** $G_s(0)$ is diagonal. This may be obtained by selecting a constant pre-compensator $W_1 = G^{-1}(0)$ (and for a non-square plant we may use the pseudo-inverse provided $G(0)$ has full row (output) rank).
3. **Approximate decoupling at frequency** w_o**:** $G_s(j\omega_o)$ is as diagonal as possible. This is usually obtained by choosing a constant pre-compensator $W_1 = G_o^{-1}$ where G_o is a real approximation of $G(j\omega_o)$. G_o may be obtained, for example, using the align algorithm of Kouvaritakis (1974) (see £le `align.m` available at the book's home page). The bandwidth frequency is a good selection for ω_o because the effect on performance of reducing interaction is normally greatest at this frequency.

The idea of decoupling control is appealing, but there are several difficulties:

1. As one might expect, decoupling may be very sensitive to modelling errors and uncertainties. This is illustrated below in Section 3.7.2 (page 100).
2. The requirement of decoupling and the use of an inverse-based controller may not be desirable for disturbance rejection. The reasons are similar to those given for SISO systems in Section 2.6.4, and are discussed further below; see (3.79).
3. If the plant has RHP-zeros then the requirement of decoupling generally introduces extra RHP-zeros into the closed-loop system (see Section 6.6.1, page 236).

Even though decoupling controllers may not always be desirable in practice, they are of interest from a theoretical point of view. They also yield insights into the limitations imposed by the multivariable interactions on achievable performance. One popular design method, which essentially yields a decoupling controller, is the internal model control (IMC) approach (Morari and Zafiriou, 1989).

Another common strategy, which avoids most of the problems just mentioned, is to use *partial (one-way) decoupling* where $G_s(s)$ in (3.73) is upper or lower triangular.

3.5.4 Pre- and post-compensators and the SVD controller

The above pre-compensator approach may be extended by introducing a post-compensator $W_2(s)$, as shown in Figure 3.10. One then designs a *diagonal* controller K_s for the shaped plant $W_2 G W_1$. The overall controller is then

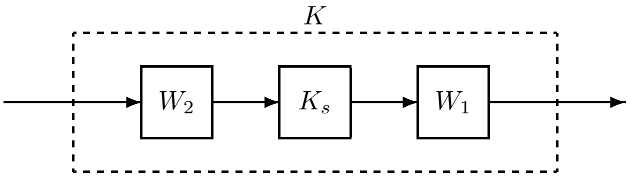


Figure 3.10: Pre- and post-compensators, W_1 and W_2. K_s is diagonal.

$$K(s) = W_1 K_s W_2 \tag{3.76}$$

The *SVD controller* is a special case of a pre- and post-compensator design. Here

$$W_1 = V_o \quad \text{and} \quad W_2 = U_o^T \tag{3.77}$$

where V_o and U_o are obtained from the SVD of $G_o = U_o \Sigma_o V_o^T$, where G_o is a real approximation of $G(j\omega_o)$ at a given frequency w_o (often around the bandwidth). SVD controllers are studied by Hung and MacFarlane (1982), and by Hovd et al. (1997) who found that the SVD-controller structure is optimal in some cases, e.g. for plants consisting of symmetrically interconnected subsystems.

In summary, the SVD controller provides a useful class of controllers. By selecting $K_s = l(s)\Sigma_o^{-1}$ a decoupling design is achieved, and selecting a diagonal K_s with a low condition number ($\gamma(K_s)$ small) generally results in a robust controller (see Section 6.10).

10

CONTROL STRUCTURE DESIGN

Most (if not all) available control theories assume that a control structure is given at the outset. They therefore fail to answer some basic questions, which a control engineer regularly meets in practice. Which variables should be controlled, which variables should be measured, which inputs should be manipulated, and which links should be made between them? The objective of this chapter is to describe the main issues involved in control structure design and to present some of the quantitative methods available, for example, for selection of controlled variables and for decentralized control.

10.1 Introduction

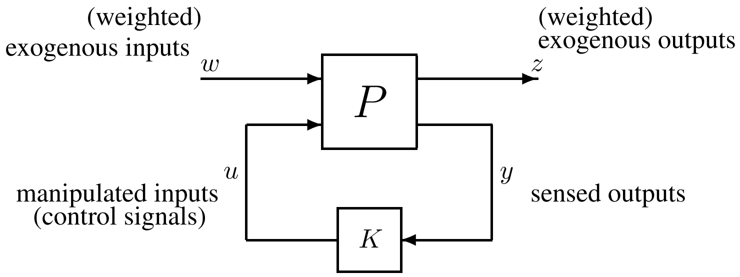


Figure 10.1: General control configuration

In much of this book, we consider the general control problem formulation shown in Figure 10.1, where the *controller design* problem is to

- Find a stabilizing controller K, which, based on the information in y, generates a control signal u, which counteracts the influence of w on z, thereby minimizing the closed-loop norm from w to z.

We presented different techniques for controller design in Chapters 2, 8 and 9. However, if we go back to Chapter 1 (page 1), then we see that controller design is only one step, step 9, in the overall process of designing a control system. In this chapter, we are concerned with the structural decisions of *control structure design*, which are the steps necessary to get to Figure 10.1:

Multivariable Feedback Control: Analysis and Design. Second Edition
S. Skogestad and I. Postlethwaite

Step 4 on page 1: The selection of controlled outputs (a set of variables which are to be controlled to achieve a set of specific objectives).
See Sections 10.2 and 10.3: *What are the variables z in Figure 10.1?*

Step 5 on page 1: The selection of manipulated inputs and measurements (sets of variables which can be manipulated and measured for control purposes).
See Section 10.4: *What are the variable sets u and y in Figure 10.1?*

Step 6 on page 1: The selection of a *control configuration* (a structure of interconnecting measurements/commands and manipulated variables).
See Sections 10.5 and 10.6: *What is the structure of K in Figure 10.1; that is, how should we "pair" the variable sets u and y?*

The distinction between the words control *structure* and control *configuration* may seem minor, but note that it is significant within the context of this book. The *control structure* (or control strategy) refers to all structural decisions included in the design of a control system (steps 4, 5 and 6). On the other hand, the *control configuration* refers only to the structuring (decomposition) of the controller K itself (step 6) (also called the measurement/manipulation partitioning or input/output pairing). Control configuration issues are discussed in more detail in Section 10.5. The selection of controlled outputs, manipulations and measurements (steps 4 and 5 combined) is sometimes called *input/output selection*.

One important reason for decomposing the control system into a specific *control configuration* is that it may allow for simple tuning of the subcontrollers without the need for a detailed plant model describing the dynamics and interactions in the process. Multivariable centralized controllers can always outperform decomposed (decentralized) controllers, but this performance gain must be traded off against the cost of obtaining and maintaining a sufficiently detailed plant model and the additional hardware.

The number of possible control structures shows a combinatorial growth, so for most systems a careful evaluation of all alternative control structures is impractical. Fortunately, we can often obtain a reasonable choice of controlled outputs, measurements and manipulated inputs from physical insight. In other cases, simple controllability measures as presented in Chapters 5 and 6 may be used for quickly evaluating or screening alternative control structures. Additional tools are presented in this chapter.

From an engineering point of view, the decisions involved in designing a complete control system are taken sequentially: first, a "top-down" selection of controlled outputs, measurements and inputs (steps 4 and 5) and then a "bottom-up" design of the control system (in which step 6, the selection of the control configuration, is the most important decision). However, the decisions are closely related in the sense that one decision directly influences the others, so the procedure may involve iteration. Skogestad (2004a) has proposed a procedure for control structure design for complete chemical plants, consisting of the following *structural* decisions:

"Top-down" (mainly step 4)

(i) Identify operational constraints and identify a scalar cost function J that characterizes optimal operation.

(ii) Identify degrees of freedom (manipulated inputs u) and in particular identify the ones that affect the cost J (in process control, the cost J is usually determined by the steady-state).

(iii) Analyze the solution of optimal operation for various disturbances, with the aim of finding primary controlled variables ($y_1 = z$) which, when kept constant, indirectly minimize the cost (“self-optimizing control”). (Section 10.3)

(iv) Determine where in the plant to set the production rate.

“Bottom-up” (steps 5 and 6)

(v) *Regulatory/base control layer*: Identify additional variables to be measured and controlled (y_2), and suggest how to pair these with manipulated inputs. (Section 10.4)

(vi) *“Advanced”/supervisory control layer* configuration: Should it be decentralized or multivariable? (Sections 10.5.1 and 10.6)

(vii) *On-line optimization layer*: Is this needed or is a constant setpoint policy sufficient (“self-optimizing control”)? (Section 10.3)

Except for decision (iv), which is specific to process control, this procedure may be applied to any control problem.

Control structure design was considered by Foss (1973) in his paper entitled “Critique of chemical process control theory” where he concluded by challenging the control theoreticians of the day to close the gap between theory and applications in this important area. Control structure design is clearly important in the chemical process industry because of the complexity of these plants, but the same issues are relevant in most other areas of control where we have large-scale systems. In the late 1980’s Carl Nett (Nett, 1989; Nett and Minto, 1989) gave a number of lectures based on his experience of aero-engine control at General Electric, under the title “A quantitative approach to the selection and partitioning of measurements and manipulations for the control of complex systems”. He noted that increases in controller complexity unnecessarily outpace increases in plant complexity, and that the objective should be to

> minimize control system complexity subject to the achievement of accuracy specifications in the face of uncertainty.

Balas (2003) recently surveyed the status of flight control. He states, with reference to the *Boeing* company, that “the key to the control design is selecting the variables to be regulated and the controls to perform regulation” (steps 4 and 5). Similarly, the first step in *Honeywell*’s procedure for controller design is “the selection of controlled variables (CVs) for performance and robustness” (step 4).

Surveys on control structure design and input–output selection are given by Van de Wal (1994) and Van de Wal and de Jager (2001), respectively. A review of control structure design in the chemical process industry (plantwide control) is given by Larsson and Skogestad (2000). The reader is referred to Chapter 5 (page 164) for an overview of the literature on input–output controllability analysis.

10.2 Optimal operation and control

The overall control objective is to maintain acceptable operation (in terms of safety, environmental impact, load on operators, and so on) while keeping the operating conditions close to economically optimal. In Figure 10.2, we show three different implementations for optimization and control:

(a) Open-loop optimization

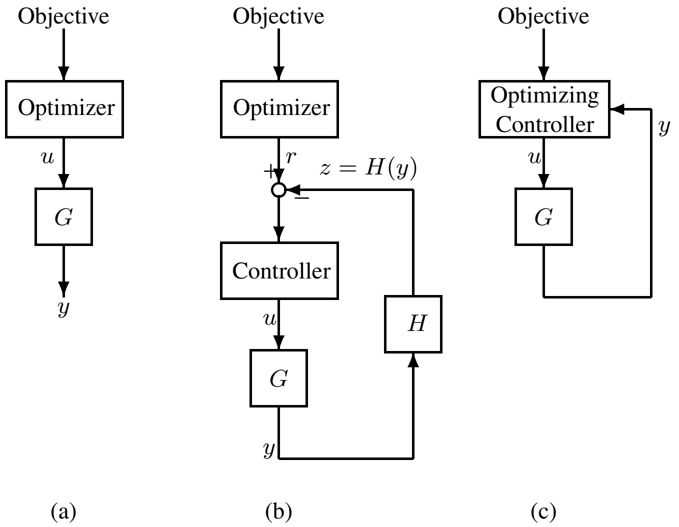


Figure 10.2: Different structures for optimization and control. (a) Open-loop optimization. (b) Closed-loop implementation with separate control layer. (c) Integrated optimization and control.

(b) Closed-loop implementation with separate control layer

(c) Integrated optimization and control ("optimizing control")

Structure (a) with open-loop optimization is usually not acceptable because of model error and unmeasured disturbances. Theoretically, optimal performance is obtained with the *centralized optimizing controller* in structure (c), which combines the functions of optimization and control in one layer. All control actions in such an ideal control system would be perfectly coordinated and the control system would use on-line dynamic optimization based on a nonlinear dynamic model of the complete plant instead of, for example, infrequent steady-state optimization. However, this solution is normally not used for a number of reasons, including: the cost of modelling, the difficulty of controller design, maintenance and modification, robustness problems, operator acceptance, and the lack of computing power.

In practice, the *hierarchical control system* in Figure 10.2(b) is used, with different tasks assigned to each layer in the hierarchy. In the simplest case we have two layers:

- *optimization layer* – computes the desired optimal reference commands r (outside the scope of this book)
- *control layer* – implements the commands to achieve $z \approx r$ (the focus of this book).

The optimization tends to be performed *open-loop* with limited use of feedback. On the other hand, the control layer is mainly based on *feedback* information. The optimization is often based on nonlinear steady-state models, whereas linear dynamic models are mainly used in the control layer (as we do throughout the book).

Additional layers are possible, as is illustrated in Figure 10.3 which shows a typical control hierarchy for a complete chemical plant. Here the control layer is subdivided into two layers: *supervisory control* ("advanced control") and *regulatory control* ("base control"). We have also included a scheduling layer above the optimization layer. Similar hierarchies

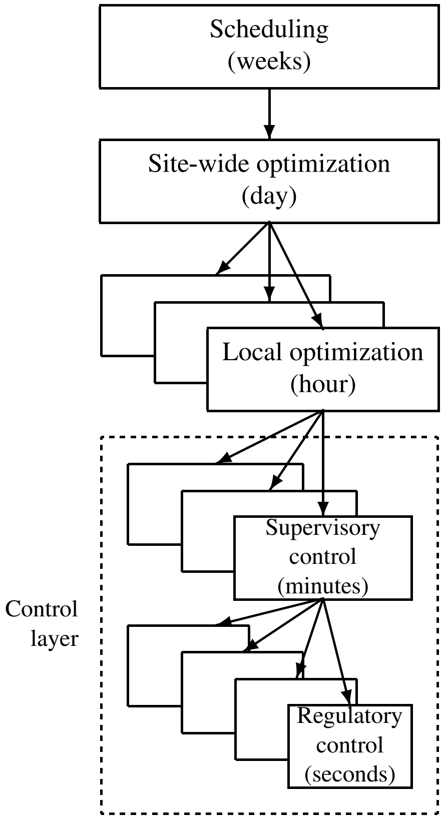


Figure 10.3: Typical control system hierarchy in a chemical plant

are found in control systems for most applications, although the time constants and names of the layers may be different. Note that we have not included any functions related to logic control (startup/ shutdown) and safety systems. These are of course important, but need not be considered during normal operation.

In general, the information ¤ow in such a control hierarchy is based on the upper layer sending *setpoints* (references, commands) to the layer below, and the lower layer reporting back any problems in achieving this. There is usually a *time scale separation* between the upper layers and the lower layers as indicated in Figure 10.3. The slower upper layer controls variables that are more important from an overall (long time scale) point of view, using as degrees of freedom the setpoints for the faster lower layer. The lower layer should take care of fast (high-frequency) disturbances and keep the system reasonably close to its optimum in the fast time scale. To reduce the need for frequent setpoint changes, we should control variables that require small setpoint changes, and this observation is the basis for Section 10.3 which deals with selecting controlled variables.

With a “reasonable” time scale separation between the layers, typically a factor of £ve or more in terms of closed-loop response time, we have the following advantages:

1. The stability and performance of a lower (faster) layer is not much in¤uenced by the presence of upper (slow) layers because the frequency of the "disturbance" from the upper layer is well inside the bandwidth of the lower layer.
2. With the lower (faster) layers in place, the stability and performance of the upper (slower) layers do not depend much on the speci£c controller settings used in the lower layers because they only effect high frequencies outside the bandwidth of the upper layers.

More generally, there are two ways of partitioning the control system:

Vertical (hiearchical) decomposition. This is the decomposition just discussed which usually results from a time scale difference between the various control objectives ("decoupling in time"). The controllers are normally designed sequentially, starting with the fast layers, and then cascaded (series interconnected) in a hierarchical manner.

Horizontal decomposition. This is used when the plant is "decoupled in space", and normally involves a set of independent decentralized controllers. Decentralized control is discussed in more detail in Section 10.6 (page 429).

Remark 1 In accordance with Lunze (1992) we have purposely used the word *layer* rather than *level* for the hierarchical decomposition of the control system. The somewhat subtle difference is that in a *multilevel* system all units contribute to satisfying the same goal, whereas in a *multilayer* system the different units have different local objectives (which preferably contribute to the overall goal). Multilevel systems have been studied in connection with the solution of optimization problems.

Remark 2 The tasks within any layer can be performed by humans (e.g. manual control), and the interaction and task sharing between the automatic control system and the human operators are very important in most cases, e.g. an aircraft pilot. However, these issues are outside the scope of this book.

Remark 3 As noted above, we may also decompose the control layer, and from now on when we talk about control con£gurations, hierarchical decomposition and decentralization, we generally refer to the control layer.

Remark 4 A fourth possible strategy for optimization and control, not shown in Figure 10.2, is (d) *extremum-seeking control*. Here the model-based block in Figure 10.2(c) is replaced by an "experimenting" controller, which, based on measurements of the cost J, perturbs the input in order to seek the extremum (minimum) of J; see e.g. Ariyur and Krstic (2003) for details. The main disadvantage with this strategy is that a fast and accurate on-line measurement of J is rarely available.

10.3 Selection of primary controlled outputs

We are concerned here with the selection of controlled outputs (controlled variables, CVs). This involves selecting the variables z to be controlled at given reference values, $z \approx r$, where r is set by some higher layer in the control hierarchy. Thus, the selection of controlled outputs (for the control layer) is usually intimately related to the hierarchical structuring of the control system shown in Figure 10.2(b). The aim of this section is to provide systematic methods for selecting controlled variables. Until recently, this has remained an unsolved problem. For example, Fisher et al. (1985) state that "Our current approach to control of a complete plant is to solve the optimal steady-state problem on-line, and then use the results of this analysis to £x the setpoints of selected controlled variables. There is no available procedure for selecting this set of controlled variables, however. Hence experience and intuition still plays a major role in the design of control systems."

The important variables in this section are:

- u – degrees of freedom (inputs)
- z – primary ("economic") controlled variables
- r – reference value (setpoint) for z
- y – measurements, process information (often including u)

In the general case, the controlled variables are selected as functions of the measurements, $z = H(y)$. For example, z can be a linear combination of measurements, i.e. $z = Hy$. In many cases, we select individual measurements as controlled variables and H is a "selection matrix" consisting of ones and zeros. Normally, we select as many controlled variables as the number of available degrees of freedom, i.e. $n_z = n_u$.

The controlled variables z are often not important variables in themselves, but are controlled in order to achieve some overall operational objective. A reasonable question is then: why not forget the whole thing about selecting controlled variables, and instead directly adjust the manipulated variables u? The reason is that an open-loop implementation usually fails because we are not able to adjust to changes (disturbances d) and errors (in the model). The following example illustrates the issues.

Example 10.1 **Cake baking.** *The overall goal is to make a cake which is well baked inside and has a nice exterior. The manipulated input for achieving this is the heat input, $u = Q$ (and we will assume that the duration of the baking is £xed, e.g. at 15 minutes).*

(a) If we had never baked a cake before, and if we were to construct the oven ourselves, we might consider directly manipulating the heat input to the oven, possibly with a watt-meter measurement. However, this open-loop implementation would not work well, as the optimal heat input depends strongly on the particular oven we use, and the operation is also sensitive to disturbances; for example, opening the oven door or whatever else might be in the oven. In short, the open-loop implementation is sensitive to uncertainty.

(b) An effective way of reducing the uncertainty is to use feedback. Therefore, in practice we use a closed-loop implementation where we control the oven temperature ($z = T$) using a thermostat. The temperature setpoint $r = T_s$ is found from a cook book (which plays the role of the "optimizer"). The (a) open-loop and (b) closed-loop implementations of the cake baking process are illustrated in Figure 10.2.

The key question is: what variables z should we control? In many cases, it is clear from a physical understanding of the process what these are. For example, if we are considering heating or cooling a room, then we should select the room temperature as the controlled variable z. Furthermore, we generally control variables that are optimally at their constraints (limits). For example, we make sure that the air conditioning is on maximum if we want to cool down our house quickly. In other cases, it is less obvious what to control, because the overall control objective may not be directly associated with keeping some variable constant.

To get an idea of the issues involved, we will consider some simple examples. Let us £rst consider two cases where implementation is obvious because the optimal strategy is to keep variables at their constraints.

Example 10.2 **Short-distance (100 m) running.** *The objective is to minimize the time T of the race ($J = T$). The manipulated input (u) is the muscle power. For a well-trained runner, the optimal solution lies at the constraint $u = u_{\max}$. Implementation is then easy: select $z = u$ and $r = u_{\max}$ or alternatively "run as fast as possible".*

Example 10.3 **Driving from A to B.** *Let y denote the speed of the car. The objective is to minimize the time T of driving from A to B or, equivalently, to maximize the speed (y), i.e. $J = -y$. If we are driving on a straight and clear road, then the optimal solution is always to stay on the speed limit constraint ($y_{\max}$). Implementation is then easy: use a feedback scheme (cruise control) to adjust the engine power (u) such that we are at the speed limit; that is, select $z = y$ and $r = y_{\max}$.*

In the next example, the optimal solution does not lie at a constraint and the selection of the controlled variable is not obvious.

Example 10.4 **Long-distance running.** *The objective is to minimize the time T of the race ($J = T$), which is achieved by maximizing the average speed. It is clear that running at maximum input power is not a good strategy. This would give a high speed at the beginning, but a slower speed towards the end, and the average speed will be lower. A better policy would be to keep constant speed ($z = y_1$ = speed). The optimization layer (e.g. the trainer) will then choose an optimal setpoint r for the speed, and this is implemented by the control layer (the runner). Alternative strategies, which may work better in a hilly terrain, are to keep a constant heart rate ($z = y_2$ = heart rate) or a constant lactate level ($z = y_3$ = lactate level).*

10.3.1 Self-optimizing control

Recall that the title of this section is selection of primary controlled outputs. In the cake baking process, we select the *oven temperature* as the controlled output z in the control layer. It is interesting to note that controlling the oven temperature in itself has no direct relation to the overall goal of making a well-baked cake. So why do we select the oven temperature as a controlled output? We now want to outline an approach for answering questions of this kind. Two distinct questions arise:

1. What variables z should be selected as the controlled variables?
2. What is the optimal reference value (z_{opt}) for these variables?

The second problem is one of optimization and is extensively studied (but not in this book). Here we want to gain some insight into the £rst problem which has been much less studied. We make the following *assumptions*:

1. The overall goal can be quanti£ed in terms of a scalar cost function J.
2. For a given disturbance d, there exists an optimal value $u_{\text{opt}}(d)$ (and corresponding value $z_{\text{opt}}(d)$), which minimizes the cost function J.
3. The reference values r for the controlled outputs z are kept constant, i.e. r is independent of the disturbances d. Typically, some average value is selected, e.g. $r = z_{\text{opt}}(\bar{d})$.

In the following, we assume that the optimally constrained variables are already controlled at their constraints ("active constraint control") and consider the "remaining" unconstrained problem with controlled variables z and remaining unconstrained degrees of freedom u.

The system behaviour is a function of the independent variables u and d, so we may formally write $J = J(u, d)$.[1] For a given disturbance d the optimal value of the cost function

[1] Note that the cost J is usually not a simple function of u and d, but is rather given by some implied relationship such as

$$\min_{u,x} J = J_0(u, x, d) \quad \text{s.t.} \quad f(x, u, d) = 0$$

where $\dim f = \dim x$ and $f(x, u, d) = 0$ represents the model equations. Formally eliminating the internal state variables x gives the problem $\min_u J(u, d)$.

is

$$J_{\mathrm{opt}}(d) \triangleq J(u_{\mathrm{opt}}(d), d) = \min_u J(u, d) \tag{10.1}$$

Ideally, we want $u = u_{\mathrm{opt}}(d)$. However, this will not be achieved in practice and we have a loss $L = J(u, d) - J_{\mathrm{opt}}(d) > 0$.

We consider the simple feedback policy in Figure 10.2(b), where we attempt to keep z constant. Note that the open-loop implementation is included as a special case by selecting $z = u$. The aim is to adjust u automatically, if necessary, when there is a disturbance d such that $u \approx u_{\mathrm{opt}}(d)$. This effectively turns the complex optimization problem into a simple feedback problem. The goal is to achieve "self-optimizing control" (Skogestad, 2000):

> *Self-optimizing control is when we can achieve an acceptable loss with constant setpoint values for the controlled variables without the need to reoptimize when disturbances occur.*

Remark. In Chapter 5, we introduced the term self-regulation, which is when acceptable dynamic control performance can be obtained with constant manipulated variables (u). Self-optimizing control is a direct generalization to the layer above where we can achieve acceptable (economic) performance with constant controlled variables (z).

The concept of self-optimizing control is inherent in many real-life scenarios including (Skogestad, 2004b):

- The *central bank* attempts to optimize the welfare of the country (J) by keeping a constant in¤ation rate (z) by varying the interest rate (u).
- The *long-distance runner* may attempt to minimize the total running time ($J = T$) by keeping a constant heart rate ($z = y_1$) or constant lactate level ($z = y_2$) by varying the muscle power (u).
- A driver attempts to minimize the fuel consumption and engine wear (J) by keeping a constant engine rotation speed (z) by varying the gear position (u).

The presence of self-optimizing control is also evident in biological systems, which have no capacity for solving complex on-line optimization problems. Here, self-optimizing control policies are the only viable solution and have developed by evolution. In business systems, the primary ("economic") controlled variables are called key performance indicators (KPIs) and their optimal values are obtained by analyzing successful businesses ("benchmarking").

The idea of self-optimizing control is further illustrated in Figure 10.4, where we see that there is a loss if we keep a constant value for the controlled variable z, rather than reoptimizing when a disturbance moves the process away from its nominal optimal operating point (denoted $\bar{d}$).

An ideal self-optimizing variable would be the gradient of the Lagrange function for the optimization problem, which should be zero. However, a direct measurement of the gradient (or a closely related variable) is rarely available, and computing the gradient generally requires knowing the value of unmeasured disturbances. We will now outline some approaches for selecting the controlled variables z. Although a model is used to £nd z, note that the goal of self-optimizing control is to eliminate the need for on-line model-based optimization.

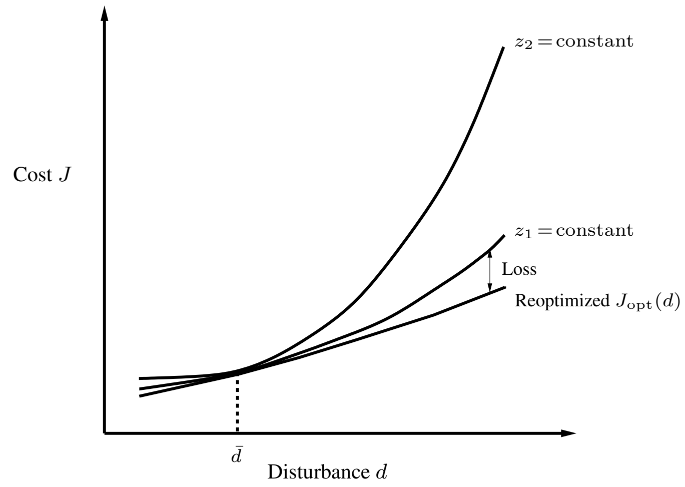


Figure 10.4: Loss imposed by keeping constant setpoint for the controlled variable. In this case z_1 is a better "self-optimizing" controlled variable than z_2.

10.3.2 Selecting controlled outputs: local analysis

We use here a local second-order accurate analysis of the loss function. From this, we derive the useful minimum singular value rule, and an exact local method; see Halvorsen et al. (2003) for further details. Note that this is a local analysis, which may be misleading; for example, if the optimum point of operation is close to infeasibility.

Consider the loss $L = J(u, d) - J_{\text{opt}}(d)$, where d is a fixed (generally non-zero) disturbance. We here make the following additional assumptions:

1. The cost function J is smooth, or more precisely twice differentiable.
2. As before, we assume that the optimization problem is unconstrained. If it is optimal to keep some variable at a constraint, then we assume that this is implemented ("active constraint control") and consider the remaining unconstrained problem.
3. The dynamics of the problem can be neglected when evaluating the cost; that is, we consider steady-state control and optimization.
4. We control as many variables z as there are available degrees of freedom, i.e. $n_z = n_u$.

For a fixed d we may then express $J(u, d)$ in terms of a Taylor series expansion in u around the optimal point. We get

$$\begin{aligned} J(u,d) \;&=\; J_{\text{opt}}(d) + \underbrace{\left(\frac{\partial J}{\partial u}\right)^T_{\text{opt}}}_{=0} (u - u_{\text{opt}}(d)) \\ &\quad + \frac{1}{2}(u - u_{\text{opt}}(d))^T \underbrace{\left(\frac{\partial^2 J}{\partial u^2}\right)_{\text{opt}}}_{=J_{uu}} (u - u_{\text{opt(d)}}) + \cdots \end{aligned} \qquad (10.2)$$

We will neglect terms of third order and higher (which assumes that we are reasonably close to the optimum). The second term on the right hand side in (10.2) is zero at the optimal point for an unconstrained problem. Equation (10.2) quanti£es how a non-optimal input $u - u_{\text{opt}}$ affects the cost function. To study how this relates to output selection we use a linearized model of the plant

$$z = Gu + G_d d \tag{10.3}$$

where G and G_d are the steady-state gain matrix and disturbance model respectively. For a £xed d, we have $z - z_{\text{opt}} = G(u - u_{\text{opt}})$. If G is invertible we then get

$$u - u_{\text{opt}} = G^{-1}(z - z_{\text{opt}}) \tag{10.4}$$

Note that G is a square matrix, since we have assumed that $n_z = n_u$. From (10.2) and (10.4) we get the second-order accurate approximation

$$L = J - J_{\text{opt}} \approx \frac{1}{2}\left(z - z_{\text{opt}}\right)^T G^{-T} J_{uu} G^{-1}\left(z - z_{\text{opt}}\right) \tag{10.5}$$

where the term $J_{uu} = (\partial^2 J/\partial u^2)_{\text{opt}}$ is independent of z. Alternatively, we may write

$$L = \frac{1}{2}\|\widetilde{z}\|_2^2 \tag{10.6}$$

where $\widetilde{z} = J_{uu}^{1/2} G^{-1}(z - z_{\text{opt}})$. These expressions for the loss L yield considerable insight. Obviously, we would like to select the controlled outputs z such that $z - z_{\text{opt}}$ is zero. However, this is not possible in practice because of (1) varying disturbances d and (2) implementation error e associated with control of z. To see this more clearly, we write

$$z - z_{\text{opt}} = z - r + r - z_{\text{opt}} = e + e_{\text{opt}}(d) \tag{10.7}$$

where

$$\text{Optimization error}: \quad e_{\text{opt}}(d) \triangleq r - z_{\text{opt}}(d)$$

$$\text{Implementation error}: \quad e \triangleq z - r$$

First, we have an optimization error $e_{\text{opt}}(d)$ because the algorithm (e.g. the cook book for cake baking) gives a desired r which is different from the optimal $z_{\text{opt}}(d)$. Second, we have a control or implementation error e because control is not perfect; either because of poor control performance or because of an incorrect measurement (steady-state bias) n^z. If we have integral action in the controller, then the steady-state control error is zero, and we have

$$e = n^z$$

If z is directly measured then n^z is its measurement error. If z is a combination of several measurements y, $z = Hy$, see Figure 10.2(b), then $n^z = Hn^y$, where n^y is the vector of measurement errors for the measurements y.

In most cases, the errors e and $e_{\text{opt}}(d)$ can be assumed independent. The maximum value of $|z - z_{\text{opt}}|$ for the expected disturbances and implementation errors, which we call the "expected optimal span", is then

$$\mathrm{span(z)} = \max_{\mathrm{d,e}} |\mathrm{z} - \mathrm{z}_{\text{opt}}| = \max_{\mathrm{d}} |\mathrm{e}_{\text{opt}}(\mathrm{d})| + \max_{\mathrm{e}} |\mathrm{e}| \tag{10.8}$$

Example 10.1 Cake baking continued. *Let us return to the question: why select the oven temperature as a controlled output? We have two alternatives: a closed-loop implementation with $z = T$ (the oven temperature) and an open-loop implementation with $z = u = Q$ (the heat input). From experience, we know that the optimal oven temperature $T_{\rm opt}$ is largely independent of disturbances and is almost the same for any oven. This means that we may always specify the same oven temperature, say $r = T_s = 190°C$, as obtained from the cook book. On the other hand, the optimal heat input $Q_{\rm opt}$ depends strongly on the heat loss, the size of the oven, etc., and may vary between, say,* 100 *W and* 5000 *W. A cook book would then need to list a different value of $r = Q_s$ for each kind of oven and would in addition need some correction factor depending on the room temperature, how often the oven door is opened, etc. Therefore, we £nd that it is much easier to get $e_{\rm opt} = T_s - T_{\rm opt}$ [$°C$] small than to get $e_{\rm opt} = Q_s - Q_{\rm opt}$ [W] small. Thus, the main reason for controlling the oven temperature is to minimize the optimization error.* In addition, the control error e is expected to be much smaller when controlling temperature.

From (10.5) and (10.7), we conclude that we should select the controlled outputs z such that:

1. G^{-1} is small (i.e. G is large); the choice of z should be such that the inputs have a large effect on z.
2. $e_{\rm opt}(d) = r - z_{\rm opt}(d)$ is small; the choice of z should be such that its optimal value $z_{\rm opt}(d)$ depends only weakly on the disturbances (and other changes).
3. $e = z - r$ is small; the choice of z should be such that it is easy to keep the control or implementation error e small.
4. G^{-1} is small, which implies that G should not be close to singular. For cases with two or more controlled variables, the variables should be selected such that they are independent of each other.

By proper scaling of the variables, these four requirements can be combined into the "maximize minimum singular value rule" as discussed next.

10.3.3 Selecting controlled outputs: maximum scaled gain method

We here derive a very simple method for selecting controlled variables in terms of the steady-state gain matrix G from inputs u (unconstrained degrees of freedom) to outputs z (candidate controlled variables).

Scalar case. In many cases we only have one unconstrained degree of freedom (u is a scalar and we want to select one z to control). Introduce the scaled gain from u to z:

$$G' = G/\text{span}(z)$$

Note form (10.8) that span$(z) = \max_{d,e} |z - z_{\rm opt}|$ includes both the optimization (setpoint) error and the implementation error. Then, from (10.5), the maximum expected loss imposed by keeping z constant is

$$L_{max} = \max_{d,e} L = \frac{|J_{uu}|}{2}\left(\frac{\max_{d,e}|z - z_{\rm opt}|}{G}\right)^2 = \frac{|J_{uu}|}{2}\frac{1}{|G'|^2} \qquad (10.9)$$

Here $|J_{uu}|$, the Hessian of the cost function, is independent of the choice for z. From (10.9), we then get that *the "scaled gain" $|G'|$ should be maximized to minimize the loss*. Note that the loss decreases with the square of the scaled gain. For an application, see Example 10.6 on page 398.

Multivariable case. Here u and z are vectors. Introduce the scaled outputs $z' \triangleq S_1 z$ and the scaled plant $G' = S_1 G$. Similar to the scalar case we scale with respect to the span,

$$S_1 = \mathrm{diag}\{\frac{1}{\mathrm{span}(z_i)}\} \tag{10.10}$$

where

$$\mathrm{span(z_i) = \max_{d,e} |z_i - z_{i,opt}| = \max_d e_{i,opt}(d) + \max_e |e_i|}$$

From (10.6), we have $L = \frac{1}{2}\|\tilde{z}\|_2^2$ where $\tilde{z} = J_{uu}^{1/2} G^{-1}(z - z_{\mathrm{opt}})$. Introducing the scaled outputs gives $\tilde{z} = J_{uu}^{1/2} G'^{-1}(z' - z'_{\mathrm{opt}})$. With the assumed scaling, the individual scaled output deviations $z'_i - z'_{i,\mathrm{opt}}$ are less than 1 in magnitude. However, the variables z_i are generally correlated, so any combinations of deviations with magnitudes less than 1 may not possible. For example, the optimal values of both z_1 and z_2 may change in the same direction when there is a disturbance. Nevertheless, we will here assume that the expected output deviations are uncorrelated by making the following assumption:

A1 The variations in $z'_i - z'_{i_{\mathrm{opt}}}$ are uncorrelated, or more precisely, the "worst-case" combination of output deviations $z'_i - z'_{i_{\mathrm{opt}}}$, with $\|z' - z'_{\mathrm{opt}}\|_2 = 1$, can occur in practice. Here $z' = S_1 z$ denotes the scaled outputs.

The reason for using the vector 2-norm, and not the max-norm, is mainly for mathematical comvenience. With assumption A1 and (A.104), we then have from (10.6) that the maximum (worst-case) loss is

$$L_{max} = \max_{\|z' - z'_{\mathrm{opt}}\|_2 \leq 1} \frac{\|\tilde{z}\|_2}{2} = \frac{1}{2}\bar{\sigma}^2(J_{uu}^{1/2} G'^{-1}) = \frac{1}{2}\frac{1}{\underline{\sigma}^2(G' J_{uu}^{-1/2})} \tag{10.11}$$

where $G' = S_1 G$ and the last equality follows from (A.40). The result may be stated as follows

> **Maximum gain (minimum singular value) rule.** *Let G denote the steady-state gain matrix from inputs u (unconstrained degrees of freedom) to outputs z (candidate controlled variables). Scale the outputs using S_1 in (10.10) and assume that A1 holds. Then to minimize the steady-state loss select controlled variables z that maximize $\underline{\sigma}(S_1 G J_{uu}^{-1/2})$.*

The rule may stated as minimizing the scaled minimum singular value, $\underline{\sigma}(G')$, of the scaled gain matrix $G' = S_1 G S_2$, where the output scaling matrix S_1 has the inverse of the spans along its diagonal, whereas the input "scaling" is generally a full matrix, $S_2 = J_{uu}^{-1/2}$. This important result was £rst presented in the £rst edition of this book (Skogestad and Postlethwaite, 1996) and proven in more detail by Halvorsen et al. (2003).

Example 10.5 *The aero-engine application in Chapter 13 (page 500) provides a nice illustration of output selection. There the overall goal is to operate the engine optimally in terms of fuel consumption, while at the same time staying safely away from instability. The optimization layer is a look-up table, which gives the optimal parameters for the engine at various operating points. Since the engine at steady-state has three degrees of freedom we need to specify three variables to keep the engine approximately at the optimal point, and six alternative sets of three outputs are given in Table 13.3.2 (page 503). For the scaled variables, the value of $\underline{\sigma}(G'(0))$ is $0.060, 0.049, 0.056, 0.366, 0.409$ and 0.342 for the six alternative sets. Based on this, the £rst three sets are eliminated. The £nal choice is then based on other considerations including controllability.*

Remark 1 In the maximum gain rule, the objective function and the magnitudes of the disturbances and measurement noise enter indirectly through the scaling S_1 of the outputs z. To obtain $S_1 = \mathrm{diag}\{\frac{1}{\mathrm{span}(z_i)}\}$ we need to obtain for each candidate output $\mathrm{span}(z_i) = \max_d |e_{i,\mathrm{opt}}(d)| + \max |e_i|$. The second contribution to the span is simply the expected measurement error, which is the measurement error plus the control error. The first contribition, $e_{i,\mathrm{opt}}$, may be obtained from a (nonlinear) model as follows: Compute the optimal values of the unconstrained z for the expected disturbances (with optimally constrained variables fixed). This yields a "look-up" table of z_{opt} for various expected disturbance combinations. From this data obtain for each candidate output, the expected variation in its optimal value, $e_{i_{\mathrm{opt}}} = (z_{i_{\mathrm{opt,max}}} - z_{i_{\mathrm{opt,min}}})/2$.

Remark 2 Our desire to have $\underline{\sigma}(G')$ large for output selection is *not* related to the desire to have $\underline{\sigma}(G)$ large to avoid input constraints as discussed in Section 6.9. In particular, the scalings, and thus the matrix G', are different for the two cases.

Remark 3 We have in our derivation assumed that the nominal operating point is optimal. However, it can be shown that the results are independent of the operating point, provided we are in the region where the cost can be approximated by a quadratic function as in (10.2) (Alstad, 2005). Thus, it is equally important to select the right controlled variables when we are nominally non-optimal.

Exercise 10.1 *Recall that the maximum gain rule requires that the minimum singular value of the (scaled) gain matrix be maximized. It is proposed that the loss can simply be minimized by selecting the controlled variables as $z = \beta y$, where β is a large number. Show that such a scaling does not affect the selection of controlled variables using the singular value method.*

10.3.4 Selecting controlled outputs: exact local method

The maximum gain rule is based on assumption A1 on page 395, which may not hold for some cases with more than one controlled variable ($n_z = n_u > 1$). This is pointed out by Halvorsen et al. (2003), who derived the following exact local method.

Let the diagonal matrix W_d contain the magnitudes of expected disturbances and the diagonal matrix W_e contain the expected implementation errors associated with the individual controlled variables. We assume that the combined disturbance and implementation error vector has norm 1, $\left\| \begin{bmatrix} d' \\ e' \end{bmatrix} \right\|_2 = 1$. Then, it may be shown that the worst-case loss is (Halvorsen et al., 2003)

$$\max_{\left\| \begin{bmatrix} d' \\ e' \end{bmatrix} \right\|_2 \le 1} L = \frac{1}{2}\bar{\sigma}([\,M_d \quad M_e\,])^2 \tag{10.12}$$

where

$$\begin{aligned} M_d &= J_{uu}^{1/2}\left(J_{uu}^{-1}J_{ud} - G^{-1}G_d\right)W_d && (10.13) \\ M_e &= J_{uu}^{1/2}G^{-1}W_e && (10.14) \end{aligned}$$

Here $J_{uu} = \left(\partial^2 J/\partial u^2\right)_{\mathrm{opt}}$, $J_{ud} = \left(\partial^2 J/\partial u \partial d\right)_{\mathrm{opt}}$ and the scaling enters through the weights W_d and W_e.

10.3.5 Selecting controlled outputs: direct evaluation of cost

The local methods presented in Sections 10.3.2-10.3.4 are very useful. However, in many practical examples nonlinear effects are important. In particular, the local methods may not be able to detect feasibility problems. For example, in marathon running, selecting a control strategy based on constant speed may be good locally (for small disturbances). However, if we encounter a steep hill (a large disturbance), then operation may not be feasible, because the selected reference value may be too high. In such cases, we may need to use a "brute force" direct evaluation of the loss and feasibility for alternative sets of controlled variables. This is done by solving the nonlinear equations, and evaluating the cost function J for various selected disturbances d and control errors e, assuming $z = r + e$ where r is kept constant (Skogestad, 2000). Here r is usually selected as the optimal value for the nominal disturbance, but this may not be the best choice and its value may also be found by optimization ("optimal back-off") (Govatsmark, 2003). The set of controlled outputs with smallest worst-case or average value of J is then preferred. This approach may be time consuming because the solution of the nonlinear equations must be repeated for each candidate set of controlled outputs.

10.3.6 Selecting controlled outputs: measurement combinations

We have so far selected z as a subset of the available measurements y. More generally, we may consider *combinations* of the measurements. We will restrict ourselves to *linear* combinations

$$z = Hy \tag{10.15}$$

where y now denotes all the available measurements, including the inputs u used by the control system. The objective is to £nd the measurement combination matrix H.

Optimal combination. Write the linear model in terms of the measurements y as $y = G^y u + G^y_d d$. Locally, the optimal linear combination is obtained by minimizing $\bar{\sigma}([\,M_d \quad M_e\,])$ in (10.12) with $W_e = HW_{n^y}$, where W_{n^y} contains the expected measurement errors associated with the individual measured variables; see Halvorsen et al. (2003). Note that H enters (10.12) indirectly, since $G = HG^y$ and $G_d = HG^y_d$ depend on H. However, (10.12) is a nonlinear function of H and numerical search-based methods need to be used.

Null space method. A simpler method for £nding H is the *null space method* proposed by Alstad and Skogestad (2004), where we neglect the implementation error, i.e., $M_e = 0$ in (10.14). Then, a constant setpoint policy ($z = r$) is optimal if $z_{\rm opt}(d)$ is independent of d, that is, when $z_{\rm opt} = 0 \cdot d$ in terms of deviation variables. Note that the optimal values of the individual measurements $y_{\rm opt}$ still depend on d and we may write

$$y_{\rm opt} = Fd \tag{10.16}$$

where F denotes the *optimal* sensitivity of y with respect to d. We would like to £nd $z = Hy$ such that $z_{\rm opt} = Hy_{\rm opt} = HFd = 0 \cdot d$ for all d. To satisfy this, we must require

$$HF = 0 \tag{10.17}$$

or that H lies in the left null space of F. This is always possible, provided $n_y \geq n_u + n_d$. This is because the null space of F has dimension $n_y - n_d$ and to make $HF = 0$, we must require that $n_z = n_u < n_y - n_d$. It can be shown that when (10.17) holds, $M_d = 0$. If there are too many disturbances, i.e. $n_y < n_u + n_d$, then one should select only the important disturbances (in terms of economics) or combine disturbances with a similar effect on y (Alstad, 2005).

In the presence of implementation errors, even when (10.17) holds such that $M_d = 0$, the loss can be large due to non-zero M_e. Therefore, the null space method does not guarantee that the loss L using a combination of measurements will be less than using the individual measurements. One practical approach is to select first the candidate measurements y, whose sensitivity to the implementation error is small (Alstad, 2005).

10.3.7 Selecting controlled outputs: examples

The following example illustrates the simple "maximize scaled gain rule" (mimimum singular value method).

Example 10.6 Cooling cycle. *A simple cooling cycle or heat pump consists of a compressor (where work W_s is supplied and the pressure is increased to p_h), a high-pressure condenser (where heat is supplied to the surroundings at high temperature), an expansion valve (where the fluid is expanded to a lower pressure p_l such that the temperature drops) and a low-pressure evaporator (where heat is removed from the surroundings at low temperature); see Figure 10.5. The compressor work is indirectly set by the amount of heating or cooling, which is assumed given. We consider a design with a flooded evaporator where there is no super-heating. In this case, the expansion valve position (u) remains as an unconstrained degree of freedom, and should be adjusted to minimize the work supplied, $J = W_s$. The question is: what variable should we control?*

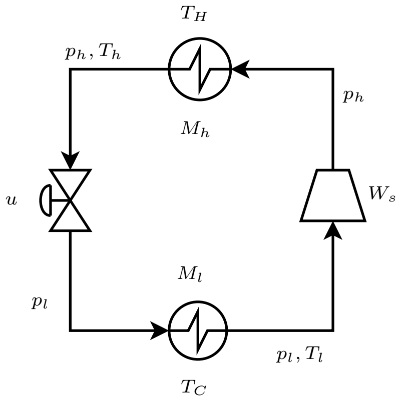


Figure 10.5: Cooling cycle

Seven alternative controlled variables are considered in Table 10.1. The data is for an ammonia cooling cycle, and we consider $\Delta y_{\rm opt}$ for a small disturbance of 0.1 K in the hot surroundings ($d_1 = T_H$). We do not consider implementation errors. Details are given in Jensen and Skogestad (2005). From (10.9), it follows that it may be useful to compute the scaled gain $G' = G/\text{span}(z(d_i))$ for the various disturbances d_i and look for controlled variables z with a large value of $|G'|$. From a physical point of view, two obvious candidate controlled variables are the high and low pressures (p_h and p_l). However, these appear to be poor choices with scaled gains $|G'|$ of 126 and 0, respectively. The

Table 10.1: Local "maximum gain" analysis for selecting controlled variable for cooling cycle

Variable (y)	$\Delta z_{\rm opt}(d_1)$	$G = \frac{\Delta z}{\Delta u}$	$\lvert G'\rvert = \frac{\lvert G\rvert}{\lvert\Delta z_{\rm opt}(d_1)\rvert}$
Condenser pressure, p_h [Pa]	3689	-464566	126
Evaporator pressure, p_l [Pa]	-167	0	0
Temperature at condenser exit, T_h [K]	0.1027	316	3074
Degree of sub-cooling, $T_h - T^{\rm sat}(p_h)$ [K]	-0.0165	331	20017
Choke valve opening, u	8.0×10^{-4}	1	1250
Liquid level in condenser, M_h $[m^3]$	6.7×10^{-6}	-1.06	157583
Liquid level in evaporator, M_l $[m^3]$	-1.0×10^{-5}	1.05	105087

zero gain is because we assume a given cooling duty $Q_C = UA(T_l - T_C)$ and further assume saturation $T_l = T^{\rm sat}(p_l)$. Keeping p_l constant is then infeasible when, for example, there are disturbances in T_C. Other obvious candidates are the temperatures at the exit of the heat exchangers, T_h and T_l. However, the temperature T_l at the evaporator exit is directly related to p_l (because of saturation) and also has a zero gain. The open-loop policy with a constant valve position u has a scaled gain of 1250*, and the temperature at the condenser exit (T_h) has a scaled gain of 3074. Even more promising is the degree of subcooling at the condenser exit with a scaled gain of* 20017*. Note that the loss decreases in proportion to $\lvert G'\rvert^2$, so the increase in the gain by a factor* $20017/1250 = 16.0$ *when we change from constant choke valve opening ("open-loop") to constant degree of subcooling, corresponds to a decrease in the loss (at least for small perturbations) by a factor* $16.0^2 = 256$*. Finally, the best single measurements seem to be the amount of liquid in the condenser and evaporator, M_h and M_l, with scaled gains of* 157583 *and* 105087*, respectively. Both these strategies are used in actual heat pump systems. A "brute force" evaluation of the cost for a (large) disturbance in the surrounding temperature ($d_1 = T_H$) of about* 10 *K, confirms the linear analysis, except that the choice $z = T_h$ turns out to be infeasible. The open-loop policy with constant valve position ($z = u$) increases the compressor work by about* 10%*, whereas the policy with a constant condenser level ($z = M_h$) has an increase of less than* 0.003%*. Similar results hold for a disturbance in the cold surroundings ($d_2 = T_C$). Note that the implementation error was not considered, so the actual losses will be larger.*

The next simple example illustrates the use of different methods for selection of controlled variables.

Example 10.7 Selection of controlled variables. *As a simple example, consider a scalar unconstrained problem, with the cost function $J = (u-d)^2$, where nominally $d^* = 0$. For this problem we have three candidate measurements,*

$$y_1 = 0.1(u-d); \quad y_2 = 20u; \quad y_3 = 10u - 5d$$

We assume the disturbance and measurement noises are of unit magnitude, i.e. $|d| \le 1$ and $|n_i^y| \le 1$. For this problem, we always have $J_{\rm opt}(d) = 0$ corresponding to

$$u_{\rm opt}(d) = d, \quad y_{1,opt}(d) = 0, \quad y_{2,opt}(d) = 20d \quad \text{and} \quad y_{3,opt}(d) = 5d$$

For the nominal case with $d^* = 0$, we thus have $u_{\rm opt}(d^*) = 0$ and $y_{\rm opt}(d^*) = 0$ for all candidate controlled variables and at the nominal operating point we have $J_{uu} = 2, J_{ud} = -2$. The linearized models for the three measured variables are

$$\begin{array}{lll} y_1\text{:} & G_1^y = 0.1, & G_{d1}^y = -0.1 \\ y_2\text{:} & G_2^y = 20, & G_{d2}^y = 0 \\ y_3\text{:} & G_3^y = 10, & G_{d3}^y = -5 \end{array}$$

Let us first consider selecting one of the individual measurements as a controlled variable. We have

$$\begin{array}{lll} \text{Case } 1: & z = y_1, & G = G_1^y \\ \text{Case } 2: & z = y_2, & G = G_2^y \\ \text{Case } 3: & z = y_3, & G = G_3^y \end{array}$$

The losses for this example can be evaluated analytically, and we find for the three cases

$$L_1 = (10e_1)^2; \quad L_2 = (0.05e_2 - d)^2; \quad L_3 = (0.1e_3 - 0.5d)^2$$

(For example, with $z = y_3$, we have $u = (y_3 + 5d)/10$ and with $z = n_3^y$, we get $L_3 = (u - d)^2 = (0.1n_3^y + 0.5d - d)^2$.) With $|d| \le 1$ and $|n_i^y| \le 1$, the worst-case losses (with $|d| = 1$ and $|n_i^y| = 1$) are $L_1 = 100$, $L_2 = 1.052 = 1.1025$ and $L_3 = 0.62 = 0.36$, and we find that $z = y_3$ is the best overall choice for self-optimizing control and $z = y_1$ is the worst. We note that $z = y_1$ is perfectly self-optimizing with respect to disturbances, but has the highest loss. This highlights the importance of considering the implementation error when selecting controlled variables. Next, we compare the three different methods discussed earlier in this section.

A. Maximum scaled gain (singular value rule): For the three choices of controlled variables we have without scaling $|G_1| = \underline{\sigma}(G_1) = 0.1$, $\underline{\sigma}(G_2) = 20$ and $\underline{\sigma}(G_3) = 10$. This indicates that z_2 is the best choice, but this is only correct with no disturbances. Let us now follow the singular value procedure.

1. The input is scaled by the factor $1/\sqrt{(\partial^2 J/\partial u^2)_{\text{opt}}} = 1/\sqrt{2}$ such that a unit deviation in each input from its optimal value has the same effect on the cost function J.
2. To find the optimum setpoint error, first note that $u_{\text{opt}}(d) = d$. Substituting $d = 1$ (the maximum disturbance) and $u = u_{\text{opt}} = 1$ (the optimal input) into the defining expressions for the candidate measurements, then gives $e_{\text{opt},1} = 0.1(u - d) = 0$, $e_{\text{opt},2} = 20u = 20$ and $e_{\text{opt},3} = 10u - 5d = 5$. Alternatively, one may use the expression (Halvorsen et al., 2003) $e_{\text{opt},i} = (G_i^y J_{uu}^{-1} J_{ud} - G_{di}^y)\Delta d$. Note that only the magnitude of $e_{\text{opt},i}$ matters.
3. For each candidate controlled variable the implementation error is assumed to be $n^z = 1$.
4. The expected variation ("span") for $z = y_1$ is $|e_{\text{opt},i}| + |n_1^y| = 0 + 1 = 1$. Similarly, for $z = y_2$ and $z = y_3$, the spans are $20 + 1 = 21$ and $5 + 1 = 6$, respectively.
5. The scaled gain matrices and the worst-case losses are

$$\begin{array}{lll} z = y_1: & |G_1'| = \frac{1}{1} \cdot 0.1/\sqrt{2} = 0.071; & L_1 = \frac{1}{2|G'|^2} = 100 \\ z = y_2: & |G_2'| = \frac{1}{21} \cdot 20/\sqrt{2} = 0.67; & L_2 = \frac{1}{2|G'|^2} = 1.1025 \\ z = y_3: & |G_3'| = \frac{1}{6} \cdot 10/\sqrt{2} = 1.18; & L_3 = \frac{1}{2|G'|^2} = 0.360 \end{array}$$

We note from the computed losses that the singular value rule (= maximize scaled gain rule) suggests that we should control $z = y_3$, which is the same as found with the "exact" procedure. The losses are also identical.

B. Exact local method: In this case, we have $W_d = 1$ and $W_{e_i} = 1$ and for y_1

$$M_d = \sqrt{2}\left(2^{-1} \cdot (-2) - 0.1^{-1} \cdot (-0.1)\right) \cdot 1 = 0 \quad \text{and} \quad M_e = \sqrt{2} \cdot 0.1^{-1} \cdot 1 = 10\sqrt{2}$$

$$L_1 = \frac{\bar{\sigma}([\, M_d \quad M_e \,])2}{2} = \frac{1}{2}(\bar{\sigma}(0 \quad 10\sqrt{2})) = 100$$

Similarly, we find with z_2 and z_3

$$L_2 = \frac{1}{2}(\bar{\sigma}(-\sqrt{2} \quad \sqrt{2}/20)) = 1.0025 \quad \text{and} \quad L_3 = \frac{1}{2}(\bar{\sigma}(-\sqrt{2}/2 \quad \sqrt{2}/10)) = 0.26$$

Thus, the exact local method also suggests selecting $z = y_3$ as the controlled variable. The reason for the slight difference from the "exact" nonlinear losses is that we assumed d and n^y individually to be less than 1 in the exact nonlinear method, whereas in the exact linear method we assumed that the combined 2-norm of d and n^y was less than 1.

C. Combinations of measurements: We now want to find the best combination $z = Hy$. In addition to y_1, y_2 and y_3, we also include the input u in the set y, i.e.

$$y = [\,y_1 \quad y_2 \quad y_3 \quad u\,]^T$$

We assume that the implementation error for u is 1, i.e. $n^u = 1$. We then have $W_n^y = I$, where W_n^y is a 4×4 matrix. Furthermore, we have

$$G^y = [\,0.1 \quad 20 \quad 10 \quad 1\,]^T \qquad G_d^y = [\,-0.1 \quad 0 \quad -5 \quad 0\,]^T$$

Optimal combination. We wish to find H such that $\bar{\sigma}([\,M_d \quad M_e\,])$ in (10.12) is minimized, where $G = HG^y$, $G_d = HG_d^y$, $W_e = HW_n^y$, $J_{uu} = 2$, $J_{ud} = -2$ and $W_d = 1$. Numerical optimization yields $H_{\rm opt} = [\,0.0209 \quad -0.2330 \quad 0.9780 \quad -0.0116\,]$; that is, the optimal combination of the three measurements and the manipulated input u is

$$z = 0.0209y_1 - 0.23306y_2 + 0.9780y_3 - 0.0116u$$

We note, as expected, that the most important contribution to z comes from the variable y_3. The loss is $L = 0.0405$, so it is reduced by a factor 6 compared to the previous best case ($L = 0.26$) with $z = y_3$.

Null space method. In the null space method we find the optimal combination without implementation error. This first step is to find the optimal sensitivity with respect to the disturbances. Since $u_{\rm opt} = d$, we have

$$\Delta y_{opt} = F\Delta d = G^y \Delta u_{\rm opt} + G_d^y \Delta d = \underbrace{(G^y + G_d^y)}_{F}\Delta d$$

and thus the optimal sensitivity is

$$F = [\,0 \quad 20 \quad 5 \quad 1\,]^T$$

To have zero loss with respect to disturbances we need to combine at least $n_u + n_d = 1 + 1 = 2$ measurements. Since we have four candidate measurements, there are an infinite number of possible combinations, but for simplicity of the control system, we prefer to combine only two measurements. To reduce the effect of implementation errors, it is best to combine measurements y with a large gain, provided they contain different information about u and d. More precisely, we should maximize $\underline{\sigma}([\,G^y \quad G_d^y\,])$. From this we find that measurements 2 and 3 are the best, with $\underline{\sigma}([\,G^y \quad G_d^y\,]) = \underline{\sigma}\begin{bmatrix} 20 & 0 \\ 10 & -5 \end{bmatrix} = 4.45$. To find the optimal combination we use $HF = 0$ or

$$20h_2 + 5h_3 = 0$$

Setting $h_2 = 1$ gives $h_3 = -4$, and the optimal combination is $z = y_2 - 4y_3$ or (normalizing the 2-norm of H to 1):

$$z = -0.2425y_2 + 0.9701y_3$$

The resulting loss when including the implementation error is $L = 0.0425$. We recommend the use of this solution, because the loss is only marginally higher (0.0425 instead of 0.0405) than that obtained using the optimal combination of all four measurements.

Maximizing scaled gain for combined measurements. For the scalar case, the "maximize scaled gain rule" can also be used to find the best combination. Consider a linear combination of measurements 2 and 3, $z = h_2y_2 + h_3y_3$. The gain from u to z is $G = h_2G_2^y + h_3G_3^y$. The span for z, $\mathrm{span}(z) = |e_{\rm opt,z}| + |e_z|$, is obtained by combining the individual spans

$$e_{\rm opt,z} = h_2e_{\rm opt,2} + h_3e_{\rm opt,3} = h_2f_2 + h_3f_3 = 20h_2 + 5h_3$$

and $|e_z| = h_2|e_2| + h_3|e_3|$. *If we assume that the combined implementation errors are 2-norm bounded,* $\| \begin{bmatrix} e_2 \\ e_3 \end{bmatrix} \|_2 \leq 1$, *then the worst-case implementation error for z is* $|e_z| = \| \begin{bmatrix} h_2 \\ h_3 \end{bmatrix} \|_2$. *The resulting scaled gain that should be maximized in magnitude is*

$$G' = \frac{G}{\text{span}} = \frac{h_2 G_2^y + h_3 G_3^y}{|h_2 e_{\text{opt},2} + h_3 e_{\text{opt},3}| + |e_z|} \tag{10.18}$$

The expression (10.18) gives considerable insight into the selection of a good measurement combination. We should select H (i.e. h_2 *and* h_3*) in order to maximize* $|G'|$. *The null space method corresponds to selecting H such that* $e_{\text{opt}} = h_2 e_{\text{opt},2} + h_3 e_{\text{opt},3} = 0$. *This gives* $h_2 = -0.2425$ *and* $h_3 = 0.9701$, *and* $|e_z| = \| \begin{bmatrix} h_2 \\ h_3 \end{bmatrix} \|_2 = 1$. *The corresponding scaled gain is*

$$G' = \frac{-20 \cdot 0.2425 + 10 \cdot 0.9701}{0 + 1} = -4.851$$

with a loss $L = \alpha/(2|G'|^2) = 0.0425$ *(as found above). (The factor* $\alpha = J_{uu} = 2$ *is included because we did not scale the inputs when obtaining* G'.*)*

Some additional examples can be found in Skogestad (2000), Halvorsen et al. (2003), Skogestad (2004b) and Govatsmark (2003).

Exercise 10.2* *Suppose that we want to minimize the LQG-type objective function,* $J = x^2 + ru^2$, $r > 0$, *where the steady-state model of the system is*

$$x + 2u - 3d = 0$$

$$y_1 = 2x, \quad y_2 = 6x - 5d, \quad y_3 = 3x - 2d$$

Which measurement would you select as a controlled variable for $r = 1$*? How does your conclusion change with variation in r? Assume unit implementation error for all measurements.*

Exercise 10.3 *In Exercise 10.2, how would your conclusions change when u (open-loop implementation policy) is also included as a candidate controlled variable? First, assume the implementation error for u is unity. Repeat the analysis, when the implementation error for u and each of the measurements is* 10.

10.3.8 Selection of controlled variables: summary

When the optimum coincides with constraints, optimal operation is achieved by controlling the active constraints. It is for the remaining unconstrained degrees of freedom that the selection of controlled variables is a dif£cult issue.

The most common "unconstrained case" is when there is only a single unconstrained degree of freedom. The rule is then to select a controlled variable such that the (scaled) gain is maximized.

Scalar rule: "maximize scaled gain $|G'|$"

- G = unscaled gain from u to z
- Scaled gain $G' = G/\text{span}$
- span = optimal range ($|e_{\text{opt}}|$) + implementation error ($|e|$)

In words, this "maximize scaled gain rule" may be expressed as follows:

> *Select controlled variables z with a large controllable range compared to their sum of optimal variation and implementation error.* Here
>
> - controllable range = range which may be reached by varying the inputs (as given by the steady-state gain)
> - optimal variation: due to disturbance (at steady-state)
> - implementation error = sum of control error and measurement error (at steady-state)

For cases with more than one unconstrained degree of freedom, we use the gain in the most dif£cult direction as expressed by the minimum singular value.

> **General "maximum gain" rule:** *"maximize the (scaled) minimum singular value $\underline{\sigma}(G')$ (at steady-state)", where $G' = S_1 G S_2$ and $S_2 = J_{uu}^{-1/2}$ (see page 395 for details).*

We have written "at steady-state" because the cost usually depends on the steady-state, but more generally it could be replaced by "at the bandwidth frequency of the layer above (which adjusts the setpoints for z)".

10.4 Regulatory control layer

In this section, we are concerned with the regulatory control layer. This is at the bottom of the control hierarchy and the objective of this layer is generally to "stabilize" the process and facilitate smooth operation. It is *not* to optimize objectives related to pro£t, which is done at higher layers. Usually, this is a decentralized control system of "low complexity" which keeps a set of measurements at given setpoints. The regulatory control layer is usually itself hierarchical, consisting of cascaded loops. If there are "truly" unstable modes (RHP-poles) then these are usually stabilized £rst. Then, we close loops to "stabilize" the system in the more general sense of keeping the states within acceptable bounds (avoiding drift), for which the key issue is local disturbance rejection.

The most important issues for regulatory control are what to measure and what to manipulate. Some simple rules for these are given on page 405. A fundamental issue is whether the introduction of a separate regulatory control layer imposes an inherent performance loss in terms of control of the primary variables z. Interestingly, the answer is "no" provided the regulatory controller does not contain RHP-zeros, and provided the layer above has full access to changing the reference values in the regulatory control layer (see Theorem 10.2 on page 416).

10.4.1 Objectives of regulatory control

Some more speci£c objectives of the regulatory control layer may be:

O1. Provide suf£cient quality of control to enable a trained operator to keep the plant running safely without use of the higher layers in the control system.

This sharply reduces the need for providing costly backup systems for the higher layers of the control hierarchy in case of failures.

O2. Allow for simple decentralized (local) controllers (in the regulatory layer) that can be tuned on-line.

O3. Take care of “fast” control, such that acceptable control is achievable using “slow” control in the layer above.

O4. Track references (setpoints) set by the higher layers in the control hierarchy.

The setpoints of the lower layers are often the manipulated variables for the higher levels in the control hierarchy, and we want to be able to change these variables as directly and with as little interaction as possible. Otherwise, the higher layer will need a model of the dynamics and interactions of the outputs from the lower layer.

O5. Provide for local disturbance rejection.

This follows from O4, since we want to be able to keep the controlled variables in the regulatory control system at their setpoints.

O6. Stabilize the plant (in the mathematical sense of shifting RHP-poles to the LHP).

O7. Avoid “drift” so that the system stays within its “linear region” which allows the use of linear controllers.

O8. Make it possible to use simple (at least in terms of dynamics) models in the higher layers.

We want to use relatively simple models because of reliability and the costs involved in obtaining and maintaining a detailed dynamic model of the plant, and because complex dynamics will add to the computational burden on the higher-layer control system.

O9. Do not introduce unnecessary performance limitations for the remaining control problem.

The “remaining control problem” is the control problem as seen from the higher layer which has as manipulated inputs the setpoints to the lower-level control system and the possible “unused” manipulated inputs. By “unnecessary” we mean limitations (e.g. RHP-zeros, large RGA elements, strong sensitivity to disturbances) that do not exist in the original problem formulation.

10.4.2 Selection of variables for regulatory control

For the following discussion, it is useful to divide the outputs y into two classes:

- y_1 – (locally) uncontrolled outputs (for which there is an associated control objective)
- y_2 – (locally) measured and controlled outputs (with reference value r_2)

By “locally” we mean here “in the regulatory control layer”. Thus, the variables y_2 are the selected controlled variables in the regulatory control layer. We also subdivide the available manipulated inputs u in a similar manner:

- u_1 – (locally) unused inputs (this set may be empty)

- u_2 – (locally) used inputs for control of y_2 (usually $n_{u_2} = n_{y_2}$)

We will study the regulatory control layer, but a similar subdivision and analysis could be performed for any control layer. The variables y_1 are sometimes called "primary" outputs, and the variables y_2 "secondary" outputs. Note that y_2 is the controlled variable (CV) in the control layer presently considered. Typically, you can think of y_1 as the variables we would really like to control and y_2 as the variables we control locally to make control of y_1 easier.

The regulatory control layer should assist in achieving the overall operational goals, so if the "economic" controlled variables z are known, then we should include them in y_1. In other cases, if the objective is to stop the system from "drifting" away from its steady-state, then the variables y_1 could be a weighted subset of the system states; see the discussion on page 418.

The most important issues for regulatory control are:

1. What should we control (what is the variable set y_2)?
2. What should we select as manipulated variables (what is the variable set u_2) and how should it be paired with y_2?

The pairing issue arises because we aim at using decentralized SISO control, if at all possible. In many cases, it is "clear" from physical considerations and experience what the variables y_2 are (see the distillation example below for a typical case). However, we have put the word "clear" in quotes, because it may sometimes be useful to question the conventional control wisdom.

We will below, see (10.28), derive transfer functions for "partial control", which are useful for a more exact analysis of the effects of various choices for y_2 and u_2. However, we will £rst present some simple rules that may be useful for reducing the number of alternatives that could be studied. This is important in order to avoid a combinatorial growth in possibilities. For a plant where we want to select m from M candidate inputs u, and l from L candidate measurements y, the number of possibilities is

$$\binom{L}{l}\binom{M}{m} = \frac{L!}{l!(L-l)!}\,\frac{M!}{m!(M-m)!} \tag{10.19}$$

A few examples: for $m = l = 1$ and $M = L = 2$ the number of possibilities is 4; for $m = l = 2$ and $M = L = 4$ it is 36; and for $m = M$, $l = 5$ and $L = 100$ (selecting 5 measurements out of 100 possible) there are 75287520 possible combinations.

It is useful to distinguish between two main cases:

1. **Cascade and indirect control.** The variables y_2 are controlled solely to assist in achieving good control of the "primary" outputs y_1. In this case r_2 (sometimes denoted $r_{2,u}$) is usually "free" for use as manipulated inputs (MVs) in the layer above for the control of y_1.
2. **Decentralized control (using sequential design).** The variables y_2 are important in themselves. In this case, their reference values r_2 (sometimes denoted $r_{2,d}$) are usually not available for the control of y_1, but rather act as disturbances to the control of y_1.

Rules for selecting y_2. Especially for the £rst case (cascade and indirect control), the following rules may be useful for identifying candidate controlled variables y_2 in the regulatory control layer:

1. y_2 should be easy to measure.

2. Control of y_2 should "stabilize" the plant.
3. y_2 should have good controllability; that is, it has favourable dynamics for control.
4. y_2 should be located "close" to the manipulated variable u_2 (as a consequence of rule 3, because for good controllability we want a small effective delay; see page 57).
5. The (scaled) gain from u_2 to y_2 should be large.

In words, the last rule says that the controllable range for y_2 (which may be reached by varying the inputs u_2) should be large compared to its expected variation (span). It is a restatement of the maximum gain rule presented on page 395 for selecting primary ("economic") controlled variables z. The rule follows because we would like to control variables y_2 that contribute to achieving optimal operation. For the scalar case, we should maximize the gain $|G'_{22}| = |G_{22}|/\text{span}(y_2)$, where G_{22} is the unscaled transfer function from u_2 to y_2, and span(y_2) is the sum of the optimal variation and the implementation error for y_2. For cases with more than one output, the "gain" is given by the minimum singular value, $\underline{\sigma}(G'_{22})$. The scaled gain (including the optimal variation and implementation error) should be evaluated for constant u_1 and approximately at the bandwidth frequency of the control layer immediately above (which adjust the references r_2 for y_2).

Rules for selecting u_2. To control y_2, we select a subset u_2 of the available manipulated inputs u. Similar considerations as for y_2 apply to the choice of candidate manipulated variables u_2:

1. Select u_2 so that controllability for y_2 is good; that is, u_2 has a "large" and "direct" effect on y_2. Here "large" means that the gain is large, and "direct" means good dynamics with no inverse response and a small effective delay.
2. Select u_2 to maximize the magnitude of the (scaled) gain from u_2 to y_2.
3. Avoid using variables u_2 that may saturate.

The last item is the only "new" requirement compared to what we stated for selecting y_2. By "saturate" we mean that the desired value of the input u_2 exceeds a physical constraint; for example, on its magnitude or rate. The last rule applies because, when an input saturates, we have effectively lost control, and reconfiguration may be required. Preferably, we would like to minimize the need for reconfiguration and its associated logic in the regulatory control layer, and rather leave such tasks for the upper layers in the control hierarchy.

Example 10.8 **Regulatory control for distillation column: basic layer.** *The overall control problem for the distillation column in Figure 10.6 has five manipulated inputs*

$$u = [\,L \quad V \quad D \quad B \quad V_T\,]^T$$

These are all flows [mol/s]: reflux L, boilup V, distillate D, bottom flow B, and overhead vapour (cooling) V_T. What to control (y) is yet to be decided.

Overall objective. From a steady-state (and economic) point of view, the column has only three degrees of freedom[2] *With pressure also controlled, there are two remaining steady-state degrees of freedom, and we want to identify the economic controlled variables $y_1 = z$ associated with these. To do this, we define the cost function J and minimize it for various disturbances, subject to the constraints, which include specifications on top composition (x_D) and bottom composition (x_B), together with upper and lower bounds on the flows. In most cases, the optimal solution lies at the constraints. A very*

[2] A distillation column has two fewer steady-state than dynamic degrees of freedom, because the integrating condenser and reboiler levels, which need to be controlled to stabilize the process, have no steady-state effect.

common situation is that both top and bottom composition optimally lie at their specifications ($y_{D,\min}$ and $x_{B,\max}$). We generally choose to control active constraints and then have

$$y_1 = z = [\,x_D \quad x_B\,]^T$$

Regulatory control: selection of y_2. We need to stabilize the two integrating modes associated with the liquid holdups (levels) in the condenser and reboiler of the column (M_D and M_B [mol]). In addition, we normally have tight control of pressure (p), because otherwise the (later) control of temperature and composition becomes more difficult. In summary, we decide to control the following three variables in the regulatory control layer:

$$y_2 = [\,M_D \quad M_B \quad p\,]^T$$

Note that these three variables are important to control in themselves.

Overall control problem. In summary, we have now identified five variables that we want to control

$$y = [\underbrace{x_D \quad x_B}_{y_1} \quad \underbrace{M_D \quad M_B \quad p}_{y_2}]^T$$

The resulting overall 5×5 control problem from u to y can be approximated as (Skogestad and Morari, 1987a):

$$\begin{bmatrix} x_D \\ x_B \\ M_D \\ M_B \\ M_V(p) \end{bmatrix} = \begin{bmatrix} g_{11}(s) & g_{12}(s) & 0 & 0 & 0 \\ g_{21}(s) & g_{22}(s) & 0 & 0 & 0 \\ -1/s & 0 & -1/s & 0 & 0 \\ g_L(s)/s & -1/s & 0 & -1/s & 0 \\ 0 & 1/(s+k_p) & 0 & 0 & -1/(s+k_p) \end{bmatrix} \begin{bmatrix} L \\ V \\ D \\ B \\ V_T \end{bmatrix} \tag{10.20}$$

In addition, there are high-frequency dynamics (delays) associated with the inputs (valves) and outputs (measurements). For control purposes it is very important to include the transfer function $g_L(s)$, which represents the liquid flow dynamics from the top to the bottom of the column, $\Delta L_B = g_L(s)\Delta L$. For control purposes, it may be approximated by a delay, $g_L(s) = e^{-\theta_L s}$. $g_L(s)$ also enters into the transfer function $g_{21}(s)$ from L to x_B, and by this decouples the distillation column dynamics at high frequencies. The overall plant model in (10.20) usually has no inherent control limitations caused by RHP-zeros, but the plant has two poles at the origin (from the integrating liquid levels, M_D and M_B), and also one pole close to the origin ("almost integrating") in $G_{\rm LV} = \begin{bmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{bmatrix}$ originating from the internal recycle in the column. These three modes need to be "stabilized". In addition, for high-purity separations, there is a potential control problem in that the $G_{\rm LV}$-subsystem is strongly coupled at steady-state, e.g. resulting in large elements in the RGA matrices for $G_{\rm LV}$ and also for the overall 5×5 plant, but fortunately the system is decoupled at high frequency because of the liquid flow dynamics represented by $g_L(s)$. Another complication is that composition measurements (y_1) are often expensive and unreliable.

Regulatory control: selection of u_2. As already mentioned, the distillation column is first stabilized by closing three decentralized SISO loops for level and pressure, $y_2 = [\,M_D \quad M_B \quad p\,]^T$. These loops usually interact weakly with each other and may be tuned independently. However, there exist many possible choices for u_2 (and thus for u_1). For example, the condenser holdup tank (M_D) has one inlet flow (V_T) and two outlet flows (L and D), and any one of these flows, or a combination, may be used effectively to control M_D. By convention, each choice ("configuration") of u_2 used for controlling level and pressure is named by the inputs u_1 left for composition control. For example, the "LV-configuration" used in many examples in this book refers to a partially controlled system where $u_2 = [\,D \quad B \quad V_T\,]^T$ is used to control levels and pressure (y_2) in the regulatory layer, and we are left with

$$u_1 = [\,L \quad V\,]^T$$

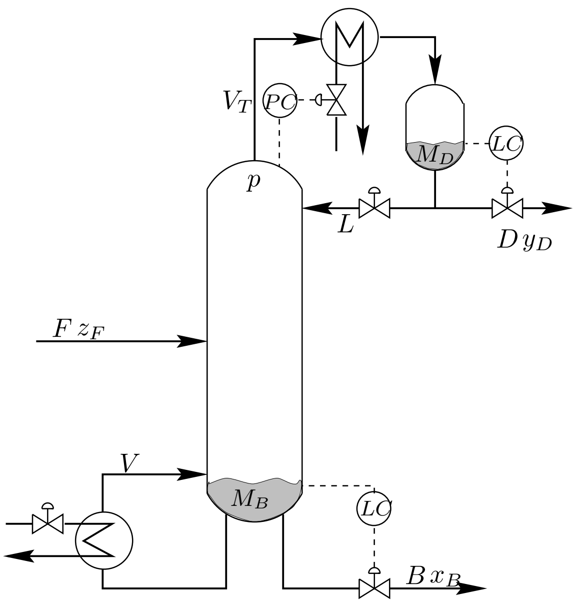


Figure 10.6: Distillation column controlled with the LV-configuration

to control composition (y_1). The LV-configuration is known to be strongly interactive at steady-state, as can been seen from the large steady-state RGA elements; see (3.94) on page 100. On the other hand, the LV-configuration is good from the point of view that it is the only configuration where control of y_1 (using u_1) is nearly independent of the tuning of the level controllers (K_2). This is quite important, because we normally want "slow" (smooth control) rather than tight control of the levels (M_D and M_B). This may give undesirable interactions from the regulatory control layer (y_2) into the primary control layer (y_1). However, this is avoided with the LV-configuration.

Another configuration is the DV-configuration where $u_2 = [\,L \quad B \quad V_T\,]^T$ is used to control levels and pressure, and we are left with

$$u_1 = [\,D \quad V\,]^T$$

to control compositions. If we were only concerned with controlling the condenser level (M_D) then this choice would be better for cases with difficult separations where $L/D \gg 1$. This is because to avoid saturation in u_2 we would like to use the largest flow (in this case $u_2 = L$) to control condenser level (M_D). In addition for this case, the steady-state interactions from u_1 to y_1, as expressed by the RGA, are generally much less; see (6.74) on page 245. However, a disadvantage with the DV-configuration is that the effect of u_1 on y_1 depends strongly on the tuning of K_2. This is not surprising, since using D to control x_D corresponds to pairing on $g_{31} = 0$ in (10.20), and D (u_1) therefore only has an effect on x_D (y_1) when the level loop (from $u_2 = L$ to $y_2 = M_D$) has been closed.

There are also many other possible configurations (choices for the two inputs in u_1); with five inputs there are ten alternative configurations. Furthermore, one often allows for the possibility of using ratios between flows, e.g. L/D, as possible degrees of freedom in u_1, and this sharply increases the number of alternatives. However, for all these configurations, the effect of u_1 on y_1 depends on the tuning of K_2, which is undesirable. This is one reason why the LV-configuration is used most in practice. In the next section, we discuss how closing a "fast" temperature loop may improve the controllability of the LV-configuration.

In the above example, the variables y_2 were important variables in themselves. In the following example, the variable y_2 is controlled to assist in the control of the primary variables y_1.

Example 10.9 Regulatory control for distillation column: temperature control. *We will assume that we have closed the three basic control loops for liquid holdup (M_D, M_B) and pressure (p) using the LV-con£guration, see Example 10.8, and we are left with a 2×2 control problem with*

$$u = [\,L \quad V\,]^T$$

(re¤ux and boilup) and

$$y_1 = [\,x_D \quad x_B\,]^T$$

(product compositions). A controllability analysis of the model $G_{\rm LV}(s)$ from u to y_1 shows that there is (1) an almost integrating mode, and (2) strong interactions. The integrating mode results in high sensitivity to disturbances at lower frequencies. The control implication is that we need to close a "stabilizing" loop. A closer analysis of the interactions (e.g. a plot of the RGA elements as a function of frequency) shows that they are much smaller at high frequencies. The physical reason for this is that L and x_D are at the top of the column, and V and x_B at the bottom, and since it takes some time (θ_L) for a change in L to reach the bottom, the high-frequency response is decoupled. The control implication is that the interactions may be avoided by closing a loop with a closed-loop response time less than about θ_L.

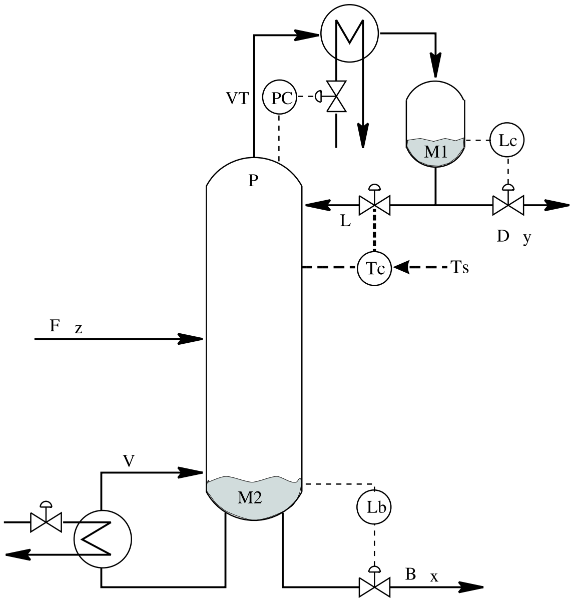


Figure 10.7: Distillation column with LV-con£guration and regulatory temperature loop

It turns out that closing one fast loop may take care of both stabilization and reducing interactions. The issue is then which loop to close. The most obvious choice is to close one of the composition loops (y_1). However, there is usually a time delay involved in measuring composition (x_D and x_B), and the measurement may be unreliable. On the other hand, the temperature T is a good indicator of composition and is easy to measure. The preferred solution is therefore to close a fast temperature loop somewhere along the column. This loop will be implemented as part of the regulatory control system.

We have two available manipulated variables u, so temperature may be controlled using reflux L or boilup V. We choose reflux L here (see Figure 10.7) because it is more likely that boilup V will reach its maximum value, and input saturation is not desired in the regulatory control layer. In terms of the notation presented above, we then have a SISO regulatory loop with

$$y_2 = T; \quad u_2 = L$$

and $u_1 = V$. The "primary" composition control layer adjusts the temperature setpoint $r_2 = T_s$ for the regulatory layer. Thus, for the primary layer we have

$$y_1 = [\,x_D \quad x_B\,]^T; \quad u = [\,u_1 \quad r_2\,]^T = [\,V \quad T_s\,]^T$$

The issue is to find which temperature T in the column to control, and for this we may use the "maximum gain rule". The objective is to maximize the scaled gain $|G'_{22}(j\omega)|$ from $u_2 = L$ to $y_2 = T$. Here, $|G'_{22}| = |G_{22}|/\mathrm{span}$ where G_{22} is the unscaled gain and span = optimal range ($|e_{opt}|$) + implementation error ($|e|$) for the selected temperature. The gain should be evaluated at approximately the bandwidth frequency of the composition layer that adjusts the setpoint $r_2 = T_s$. For this application, we assume that the primary layer is relatively slow, such that we can evaluate the gain at steady-state, i.e. $\omega = 0$.

In Table 10.2, we show the normalized temperatures $y_2 = x$, unscaled gain, optimal variation for the two disturbances, implementation error, and the resulting span and scaled gain for measurements located at stages 1 (reboiler), 5, 10, 15, 21 (feed stage), 26, 31, 36 and 41 (condenser). The gains are also plotted as a function of stage number in Figure 10.8. The largest scaled gain of about 88 is achieved when the temperature measurement is located at stage 15 from the bottom. However, this is below the feed stage and it takes some time for the change in reflux ($u_2 = L$), which enters at the top, to reach this stage. Thus, for dynamic reasons it is better to place the measurement in the top part of the column; for example, at stage 27 where the gain has a "local" peak of about 74.

Table 10.2: Evaluation of scaled gain $|G'_{22}|$ for alternative temperature locations (y_2) for distillation example. Span = $|\Delta y_{2,\mathrm{opt}}(d_1)| + |\Delta y_{2,\mathrm{opt}}(d_2)| + e_{y_2}$. Scaled gain $|G'_{22}| = |G_{22}|/\mathrm{span}$.

	Nominal	Unscaled					Scaled
Stage	value y_2	G_{22}	$\Delta y_{2,\mathrm{opt}}(d_1)$	$\Delta y_{2,\mathrm{opt}}(d_2)$	e_{y_2}	$\mathrm{span}(y_2)$	$\lvert G'_{22}\rvert$
1	0.0100	1.0846	0.0077	0.0011	0.05	0.0588	18.448
5	0.0355	3.7148	0.0247	0.0056	0.05	0.0803	46.247
10	0.1229	10.9600	0.0615	0.0294	0.05	0.1408	77.807
15	0.2986	17.0030	0.0675	0.0769	0.05	0.1944	87.480
21	0.4987	9.6947	-0.0076	0.0955	0.05	0.1532	63.300
26	0.6675	14.4540	-0.0853	0.0597	0.05	0.1950	74.112
31	0.8469	10.5250	-0.0893	0.0130	0.05	0.1524	69.074
36	0.9501	4.1345	-0.0420	-0.0027	0.05	0.0947	43.646
41	0.9900	0.8754	-0.0096	-0.0013	0.05	0.0609	14.376

Remarks to example.

1. We use data for "column A" (see Section 13.4) which has 40 stages. This column separates a binary mixture, and for simplicity we assume that the temperature T on stage i is directly given by the mole fraction of the light component, $T_i = x_i$. This can be regarded as a "normalized" temperature which ranges from 0 in the bottom to 1 in the top of the column. The implementation error is assumed to be the same on all stages, namely $e_{y_2} = 0.05$ (and with a temperature difference between the two components of 13.5 K, this corresponds to an implementation error of ± 0.68 K). The disturbances are a 20% increase in feed rate F ($d_1 = 0.2$) and a change from 0.5 to 0.6 in feed mole fraction z_F ($d_2 = 0.1$).

2. *The optimal variation ($\Delta y_{2,\mathrm{opt}}(d)$) is often obtained from a detailed steady-state model, but it was generated here from the linear model. For any disturbance d we have in terms of deviation variables (we omit the Δ's)*

$$y_1 = G_{12}u_2 + G_{d1}d$$
$$y_2 = G_{22}u_2 + G_{d2}d$$

The optimal strategy is to have the product compositions constant; that is, $y_1 = [\, x_D \quad x_B \,]^T = 0$. However, since $u_2 = L$ is a scalar, this is not possible. The best solution in a least squares sense (minimize $\|y_1\|_2$) is found by using the pseudo-inverse, $u_2^{\mathrm{opt}} = -G_{12}^{\dagger}G_{d1}d$. The resulting optimal change in the temperature $y_2 = T$ is then

$$y_2^{\mathrm{opt}} = (-G_{22}G_{12}^{\dagger}G_{d1} + G_{d2})d \tag{10.21}$$

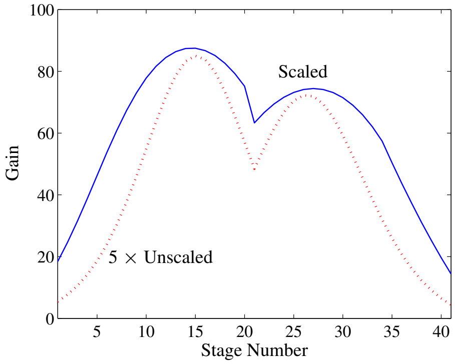


Figure 10.8: Scaled ($|G'_{22}|$) and unscaled ($|G_{22}|$) gains for alternative temperature locations for the distillation example

3. *As seen from the solid and dashed lines in Figure 10.8, the local peaks of the unscaled and scaled gains occur at stages 26 and 27, respectively. Thus, scaling does not affect the £nal conclusion much in this case. However, if we were to set the implementation error e to zero, then the maximum scaled gain would be at the bottom of the column (stage 1).*
4. *We made the choice $u_2 = L$ to avoid saturation in the boilup V in the regulatory control layer. However, if saturation is not a problem, then the other alternative $u_2 = V$ may be better. A similar analysis with $u_2 = V$ gives a maximum scaled gain of about 100 is obtained with the temperature measured at stage 14.*

In summary, the overall 5×5 distillation control problem may be solved by £rst designing a 4×4 "stabilizing" (regulatory) controller K_2 for levels, pressure and temperature

$$y_2 = [\, M_D \quad M_B \quad p \quad T \,]^T, \quad u_2 = [\, D \quad B \quad V_T \quad L \,]^T$$

and then designing a 2×2 "primary" controller K_1 for composition control

$$y_1 = [\, x_D \quad x_B \,], \quad u_1 = [\, V \quad T_s \,]$$

Alternatively, we may interchange L and V in u_1 and u_2. The temperature sensor (T) should be located at a point with a large scaled gain.

We have discussed some simple rules and tools ("maximum gain rule") for selecting the variables in the regulatory control layer. The regulatory control layer is usually itself hierarchical, consisting of a layer for stabilization of unstable modes (RHP-poles) and a layer for "stabilization" in terms of disturbance rejection. Next, we introduce pole vectors and partial control, which are more speci£c tools for addressing the issues of stabilization and disturbance rejection.

10.4.3 Stabilization: pole vectors

Pole vectors are useful for selecting inputs and outputs for stabilization of unstable modes (RHP-poles) when input usage is an issue. An important advantage is that the selection of inputs is treated separately from the selection of outputs and hence we avoid the combinatorial issue. The main disadvantage is that the theoretical results only hold for cases with a *single* RHP-pole, but applications show that the tool is more generally useful.

The issue is: which outputs (measurements) and inputs (manipulations) should be used for stabilization? We should clearly avoid saturation of the inputs, because this makes the system effectively open-loop and stabilization is then impossible. A reasonable objective is therefore to minimize the input usage required for stabilization. In addition, this choice also minimizes the "disturbing" effect that the stabilization layer has on the remaining control problem.

Recall that $u = -KS(r + n - d)$, so input usage is minimized when the norm of KS is minimal. We will consider both the $\mathcal{H}_2$ and $\mathcal{H}_\infty$ norms.

Theorem 10.1 (Input usage for stabilization) *For a rational plant with a single unstable mode p, the minimal $\mathcal{H}_2$ and $\mathcal{H}_\infty$ norms of the transfer function KS are given as (Havre and Skogestad, 2003; Kariwala, 2004)*

$$\min_K \|KS\|_2 = \frac{(2p)^{3/2} \cdot |q^T t|}{\|u_p\|_2 \cdot \|y_p\|_2} \tag{10.22}$$

$$\min_K \|KS\|_\infty = \frac{2p \cdot |q^T t|}{\|u_p\|_2 \cdot \|y_p\|_2} \tag{10.23}$$

Here u_p and y_p denote the input and output pole vectors (see page 127), respectively, and t and q are the right and left eigenvectors of the state matrix A, satisfying $At = pt$ and $q^T A = q^T p$.

Theorem 10.1 applies to plants with any number of RHP-zeros and to both multivariable (MIMO) and single-loop (SISO) control. In the SISO case, u_p and y_p are the elements in the pole vectors, $u_{p,j}$ and $y_{p,i}$, corresponding to the selected input (u_j) and output (y_i). Notice that the term $(q^T t)$ is independent of the selected inputs and outputs, u_j and y_i. Thus, for a single unstable mode and SISO control:

> *The input usage required for stabilization is minimized by selecting the output y_i (measurement) and input u_j (manipulation) corresponding to the largest elements in the output and input pole vectors (y_p and u_p), respectively* (see also Remark 2 on page 137).

This choice maximizes the (state) controllability and observability of the unstable mode. Note that the selections of measurement y_i and input u_j are performed *independently*. The above result is for unstable poles. However, Havre (1998) shows that the input requirement for pole placement is minimized by selecting the output and input corresponding to the largest elements in the y_p and u_p, respectively. This property also holds for LHP-poles, and shows that pole vectors may also be useful when we want to move stable poles.

Exercise 10.4 * *Show that for a system with a single unstable pole, (10.23) represents the least achievable value of* $\|KS\|_\infty$. *(Hint: Rearrange (5.31) on page 178 using the definition of pole vectors.)*

When the plant has *multiple* unstable poles, the pole vectors associated with a specific RHP-pole give a measure of input usage required to move this RHP-pole assuming that the other RHP-poles are unchanged. This is of course unrealistic; nevertheless, the pole vector approach can be used by stabilizing one source of instability at a time. That is, first an input and an output are selected considering one real RHP-pole or a pair of complex RHP-poles and a stabilizing controller is designed. Then, the pole vectors are recomputed for the partially controlled system and another set of variables is selected. This process is repeated until all the modes are stabilized. This process results in a sequentially designed decentralized controller and has been useful in several practical applications, as demonstrated by the next example.

Example 10.10 **Stabilization of Tennessee Eastman process.** *The Tennessee Eastman chemical process (Downs and Vogel, 1993) was introduced as a challenge problem to test methods for control structure design.*[3] *The process has 12 manipulated inputs and 41 candidate measurements, of which we consider 11 here; see Havre (1998) for details on the selection of these variables and scaling. The model has six unstable poles at the operating point considered,* $p = [\,0 \quad 0.001 \quad 0.023 \pm j0.156 \quad 3.066 \pm j5.079\,]$. *The absolute values of the output and input pole vectors are*

$$|Y_p| = \begin{bmatrix} 0.000 & 0.001 & 0.041 & 0.112 \\ 0.000 & 0.004 & 0.169 & 0.065 \\ 0.000 & 0.000 & 0.013 & 0.366 \\ 0.000 & 0.001 & 0.051 & 0.410 \\ 0.009 & 0.581 & 0.488 & 0.316 \\ 0.000 & 0.001 & 0.041 & 0.115 \\ 1.605 & 1.192 & 0.754 & 0.131 \\ 0.000 & 0.001 & 0.039 & 0.108 \\ 0.000 & 0.001 & 0.038 & 0.217 \\ 0.000 & 0.001 & 0.055 & \mathbf{1.485} \\ 0.000 & 0.002 & 0.132 & 0.272 \end{bmatrix} \quad |U_p|^T = \begin{bmatrix} 6.815 & 6.909 & 2.573 & 0.964 \\ 6.906 & 7.197 & 2.636 & 0.246 \\ 0.148 & 1.485 & 0.768 & 0.044 \\ 3.973 & 11.550 & 5.096 & 0.470 \\ 0.012 & 0.369 & 0.519 & 0.356 \\ 0.597 & 0.077 & 0.066 & 0.033 \\ 0.135 & 1.850 & 1.682 & 0.110 \\ 22.006 & 0.049 & 0.000 & 0.000 \\ 0.007 & 0.054 & 0.010 & 0.013 \\ 0.247 & 0.708 & 1.501 & \mathbf{2.021} \\ 0.109 & 0.976 & 1.447 & 0.753 \\ 0.033 & 0.095 & 0.201 & 0.302 \end{bmatrix}$$

where we have combined pole vectors corresponding to a complex eigenvalue into a single column. The individual columns of $|Y_p|$ *and individual rows of* $|U_p|$ *correspond to the poles at* 0, 0.001, $0.023 \pm j0.156$ *and* $3.066 \pm j5.079$, *respectively.*

When designing a stabilizing control system, we normally start by stabilizing the "most unstable" (fastest) pole, i.e. complex poles at $3.066 \pm j5.079$ *in this case. From the pole vectors, this mode is most easily stabilized by use of* u_{10} *and* y_{10}. *A PI controller, with proportional gain of* -0.05 *and integral time of* 300 *minutes, is designed for this loop. This simple controller stabilizes the complex unstable poles at* $3.066 \pm j5.079$ *and also at* $0.023 \pm j0.156$. *This is reasonable since the pole vectors show that the modes at* $0.023 \pm j0.156$ *are observable and controllable through* y_{10} *and* u_{10}, *respectively. For stabilizing the integrating modes, the pole vectors can be recomputed to select two additional inputs and outputs; see Havre (1998) for details.*

Note that the different choices of inputs and outputs for stabilization have different effects on the controllability of the stabilized system. Thus, in some cases, variable selection using pole vectors may need to be repeated a few times before a satisfactory solution is obtained. An alternative approach is to use the method by Kariwala (2004), which also handles the case of multiple unstable modes directly, but is more involved than the simple pole-vector-based method.

[3] Simulink and Matlab models for the Tennessee Eastman process are available from Professor Larry Ricker at the University of Washington (easily found using a search engine).

Exercise 10.5 * *For systems with multiple unstable poles, the variables can be selected sequentially using the pole vector approach by stabilizing one real pole or a pair of complex poles at a time. Usually, the selected variable does not depend on the controllers designed in the previous steps. Verify this for each of the following two systems:*

$$G_1(s) = Q(s) \cdot \begin{bmatrix} 10 & 2 & 1 \\ 12 & 1.5 & 5.01 \end{bmatrix} \quad G_2(s) = Q(s) \cdot \begin{bmatrix} 10 & 2 & 1 \\ 12 & 1 & 1.61 \end{bmatrix}$$

$$Q(s) = \begin{bmatrix} 1/(s-1) & 0 \\ 0 & 1/(s-0.5) \end{bmatrix}$$

(Hint: Use simple proportional controllers for stabilization of $p = 1$ and evaluate the effect of change of controller gain on pole vectors in the second iteration.)

10.4.4 Local disturbance rejection: partial control

Let y_1 denote the primary variables, and y_2 the locally controlled variables. We start by deriving the transfer functions for y_1 for the *partially controlled system* when y_2 is controlled. We also partition the inputs u into the sets u_1 and u_2, where the set u_2 is used to control y_2. The model $y = Gu$ may then be written[4]

$$y_1 = G_{11}u_1 + G_{12}u_2 + G_{d1}d \tag{10.24}$$

$$y_2 = G_{21}u_1 + G_{22}u_2 + G_{d2}d \tag{10.25}$$

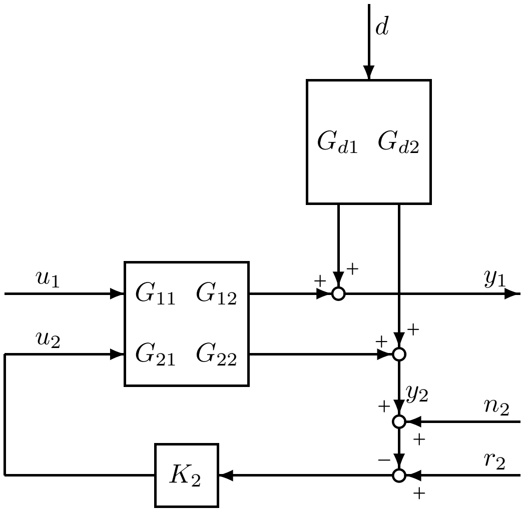


Figure 10.9: Partial control

Now assume that feedback control

$$u_2 = K_2(r_2 - y_{2,m})$$

[4] We may assume that any stabilizing loops have already been closed, so for the model $y = Gu$, G includes the stabilizing controller and u includes any "free" setpoints to the stabilizing layer below.

is used for the secondary subsystem involving u_2 and y_2, see Figure 10.9, where $y_{2,m} = y_2 + n_2$ is the measured value of y_2. By eliminating u_2 and y_2, we then get the following model for the resulting partially controlled system from u_1, r_2, d and n_2 to y_1:

$$\begin{aligned} y_1 \;=\; & \underbrace{\left(G_{11} - G_{12}K_2(I + G_{22}K_2)^{-1}G_{21}\right)}_{P_u} u_1 \\ & + \underbrace{\left(G_{d1} - G_{12}K_2(I + G_{22}K_2)^{-1}G_{d2}\right)}_{P_d} d \\ & + \underbrace{G_{12}K_2(I + G_{22}K_2)^{-1}}_{P_r}(r_2 - n_2) \end{aligned} \tag{10.26}$$

Note that P_d, the *partial disturbance gain*, is the disturbance gain for a system under partial control. P_u is the effect of u_1 on y_1 with y_2 controlled. In many cases, the set u_1 is empty because there are no extra inputs. In such cases, r_2 is probably available for control of y_1, and P_r gives the effect of r_2 on y_1. In other cases, r_2 may be viewed as a disturbance for the control of y_1.

In the following discussion, we assume that the control of y_2 is fast compared to the control of y_1. This results in a time scale separation between these layers, which simplifies controller design. To obtain the resulting model we may let $K_2 \to \infty$ in (10.26). Alternatively, we may solve for u_2 in (10.25) to get

$$u_2 = -G_{22}^{-1}G_{d2}d - G_{22}^{-1}G_{21}u_1 + G_{22}^{-1}y_2 \tag{10.27}$$

We have assumed that G_{22} is square and invertible, otherwise we can use a least squares solution by replacing G_{22}^{-1} by the pseudo-inverse, $G_{22}^{\dagger}$. On substituting (10.27) into (10.24) and assuming $y_2 \approx r_2 - n_2$ (“perfect” control), we get

$$y_1 \approx \underbrace{\left(G_{11} - G_{12}G_{22}^{-1}G_{21}\right)}_{P_u} u_1 + \underbrace{\left(G_{d1} - G_{12}G_{22}^{-1}G_{d2}\right)}_{P_d} d + \underbrace{G_{12}G_{22}^{-1}}_{P_r}\underbrace{(r_2 - n_2)}_{y_2} \tag{10.28}$$

The advantage of the approximation (10.28) over (10.26) is that it is independent of K_2, but we stress that it is useful only at frequencies where y_2 is tightly controlled.

Remark 1 Relationships similar to those given in (10.28) have been derived by many authors, e.g. see the work of Manousiouthakis et al. (1986) on block relative gains and the work of Haggblom and Waller (1988) on distillation control configurations.

Remark 2 Equation (10.26) may be rewritten in terms of linear fractional transformations (page 543). For example, the transfer function from u_1 to y_1 is

$$F_l(G, -K_2) = G_{11} - G_{12}K_2(I + G_{22}K_2)^{-1}G_{21} \tag{10.29}$$

Exercise 10.6 *The block diagram in Figure 10.11 below shows a cascade control system where the primary output y_1 depends directly on the extra measurement y_2, so $G_{12} = G_1G_2$, $G_{22} = G_2$, $G_{d1} = [\,I \quad G_1\,]$ and $G_{d2} = [\,0 \quad I\,]$. Assume tight control of y_2. Show that $P_d = [\,I \quad 0\,]$ and $P_r = G_1$ and discuss the result. Note that P_r is the “new” plant as it appears with the inner loop closed.*

The selection of secondary variables y_2 depends on whether u_1 or r_2 (or any) are available for control of y_1. Next, we consider in turn each of the three cases that may arise.

1. Cascade control system

Cascade control is a special case of partial control, where we use u_2 to control (tightly) the secondary outputs y_2, and r_2 replaces u_2 as a degree of freedom for controlling y_1. We would like to avoid the introduction of additional (new) RHP-zeros, when closing the secondary loops. The next theorem shows that this is not a problem.

Theorem 10.2 (RHP-zeros due to closing of secondary loop) *Assume that $n_{y_1} = n_{u_1} + n_{u_2}$ and $n_{y_2} = n_{r_2} = n_{u_2}$ (see Figure 10.9). Let the plant $G = \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix}$ and the secondary loop ($S_2 = (I + G_{22}K_2)^{-1}$) be stable. Then the partially controlled plant*

$$P_{\text{CL}} = [\, G_{11} - G_{12}K_2S_2G_{21} \quad G_{12}K_2S_2 \,] \tag{10.30}$$

from $[u_1 \; r_2]$ to y_1 in (10.26) has <u>no</u> additional RHP-zeros (that are not present in the open-loop plant $[\, G_{11} \quad G_{12} \,]$ from $[u_1 \; u_2]$ to y_1) if

1. *r_2 is available for control of y_1, and*
2. *K_2 is minimum-phase.*

Proof: Under the dimensional and stability assumptions, P_{CL} is a stable and square transfer function matrix. Thus, the RHP-zeros of P_{CL} are the points in RHP where $\det(P_{\text{CL}}(s)) = 0$ (also see Remark 4 on page 141). Using Schur's formula in (A.14),

$$\det(P_{\text{CL}}) = \det(M) \cdot \det(S_2)$$

where

$$M = \left[\begin{array}{cc|c} G_{11} & 0 & G_{12}K_2 \\ \hline G_{21} & -I & I + G_{22}K_2 \end{array}\right]$$

with the partitioning as shown above. By exchanging the columns of M, we have

$$\begin{aligned} \det(M) &= (-1)^n \det\left(\left[\begin{array}{cc|c} G_{11} & G_{12}K_2 & 0 \\ \hline G_{21} & I + G_{22}K_2 & -I \end{array}\right]\right) \\ &= \det\left(\left[\begin{array}{cc} G_{11} & G_{12}K_2 \end{array}\right]\right) \\ &= \det\left(\left[\begin{array}{cc} G_{11} & G_{12} \end{array}\right]\right) \det\left(\left[\begin{array}{cc} I & 0 \\ 0 & K_2 \end{array}\right]\right) \\ &= \det\left(\left[\begin{array}{cc} G_{11} & G_{12} \end{array}\right]\right) \cdot \det(K_2) \end{aligned}$$

The second equality follows since the rearranged matrix is block triangular and $\det(-I) = (-1)^n$. Then, putting everything together, we have that

$$\det(P_{\text{CL}}) = \det\left(\left[\begin{array}{cc} G_{11} & G_{12} \end{array}\right]\right) \cdot \det(K_2) \cdot \det(S_2)$$

Although the RHP-poles of K_2 appear as RHP-zeros of S_2 due to the interpolation constraints, these zeros are cancelled by K_2 and thus $\det(K_2) \cdot \det(S_2)$ evaluated at RHP-poles of K_2 is non-zero. Therefore, when r_2 is available for control of y_1 and K_2 is minimum-phase, the RHP-zeros of P_{CL} are the same as the RHP-zeros of $[\, G_{11} \quad G_{12} \,]$ and the result follows. When u_1 is empty, the transfer matrix from r_2 to y_1 is given as $G_{12}K_2(I+G_{22}K_2)^{-1}$ and thus K_2 being minimum-phase implies that the secondary loop does not introduce any additional RHP-zeros. A somewhat more restrictive version of this theorem was proven by Larsson (2000). The proof here is due to V. Kariwala. Note that the assumptions on the dimensions of y_1 and u_2 are made for simplicity of the proof and the conclusions of Theorem 10.2 still hold when these assumptions are relaxed. □

For a stable plant G, the controller K_2 can usually be chosen to be minimum-phase. Then, Theorem 10.2 implies that whenever r_2 is available for control of y_1, closing the secondary loops does not introduce additional RHP-zeros. However, note that closing secondary loops *may* make the system more sensitive to disturbances if the action of the secondary (inner) loop "overcompensates" and thereby makes the system more sensitive to the disturbance. As an example consider a plant with $G_{d1} = 1, G_{12} = 1, G_{22} = -0.1$ and $G_{d2} = 1$. Then with tight control of y_2, the disturbance gain for y_1 increases by a factor 9, from $G_{d1} = 1$ to $P_d = G_{d1} - G_{12}G_{22}^{-1}G_{d2} = 9$. In summary, it follows that we should select secondary variables for cascade control such that the input–output controllability of the "new" partially controlled plant $P_{\rm CL} = [\, G_{11} - G_{12}K_2S_2G_{21} \quad G_{12}K_2S_2 \,] = [\, P_u \quad P_r \,]$ with disturbance model P_d is better than that of the "original" plant $[\, G_{11} \quad G_{12} \,]$ with disturbance model G_{d1}. In particular, this requires that

1. $\underline{\sigma}([\, P_u \quad P_r \,])$ (or $\underline{\sigma}(P_r)$, if u_1 is empty) is large at low frequencies.
2. $\bar{\sigma}([\, P_d \quad -P_r \,])$ is small and at least smaller than $\bar{\sigma}(G_{d1})$. In particular, this argument applies at higher frequencies. Note that P_r measures the effect of measurement noise n_2 on y_1.
3. To ensure that u_2 has enough power to reject the local disturbances d and track r_2, based on (10.27), we require that $\bar{\sigma}(G_{22}^{-1}G_{d2}) < 1$ and $\bar{\sigma}(G_{22}^{-1}) < 1$. Here, we have assumed that the inputs have been scaled as outlined in Section 1.4.

Remark 1 The above recommendations for selection of secondary variables are stated in terms of singular values, but the choice of norm is usually of secondary importance. The minimization of $\bar{\sigma}([\, P_d \quad -P_r \,])$ arises if $\| \begin{bmatrix} d \\ n_2 \end{bmatrix} \|_2 \leq 1$ and we want to minimize $\|y_1\|_2$.

Remark 2 By considering the cost function $J = \min_{d,n_2} y_1^T y_1$, the selection of secondary variables for disturbance rejection using the objectives outlined above is closely related to the concept of self-optimizing control discussed in Section 10.3.

2. Sequentially designed decentralized control system

When r_2 is *not* available for control of y_1, we have a sequentially designed decentralized controller. Here the variables y_2 are important in themselves and we first design a controller K_2 to control the subset y_2. With this controller K_2 in place (a partially controlled system), we may then design a controller K_1 for the remaining outputs.

In this case, secondary loops can introduce "new" RHP-zeros in the partially controlled system P_u. For example, this is likely to happen if we pair on negative RGA elements (Shinskey, 1967; 1996); see Example 10.22 (page 447). Such zeros, however, can be moved to high frequencies (beyond the bandwidth), if it is possible to tune the inner (secondary) loop sufficiently fast (Cui and Jacobsen, 2002).

In addition, based on the general objectives for variable selection, we require that $\underline{\sigma}(P_u)$ instead of $\underline{\sigma}([\, P_u \quad P_r \,])$ be large. The other objectives for secondary variable selection are the same as for cascade control and are therefore not repeated here.

3. Indirect control

Indirect control is when neither r_2 nor u_1 are available for control of y_1. The objective is to minimize $J = \|y_1 - r_1\|$, but we assume that we cannot measure y_1. Instead we hope that y_1

is indirectly controlled by controlling y_2. With perfect control of y_2, as before

$$y_1 = P_d d + P_r(r_2 - n_2)$$

With $n_2 = 0$ and $d = 0$ this gives $y_1 = G_{12}G_{22}^{-1}r_2$, so r_2 must be chosen such that

$$r_2 = G_{22}G_{12}^{-1}r_1 \tag{10.31}$$

The control error in the primary output is then

$$y_1 - r_1 = P_d d - P_r n_2 \tag{10.32}$$

To minimize $J = \|y_1 - r_1\|$ we should therefore (as for the two other cases) select the controlled outputs y_2 such that $\|P_d d\|$ and $\|P_r n_2\|$ are small or, in terms of singular values, $\bar{\sigma}([\,P_d \quad -P_r\,])$ is small. The problem of indirect control is closely related to that of cascade control. The main difference is that in *cascade control* we also measure and control y_1 in an outer loop; so in cascade control we need $\|\,[\,P_d \quad P_r\,]\,\|$ small only at frequencies outside the bandwidth of the outer control loop (involving y_1).

Remark 1 In some cases, this measurement selection problem involves a trade-off between wanting $\|P_d\|$ small (wanting a strong correlation between measured outputs y_2 and "primary" outputs y_1) and wanting $\|P_r\|$ small (wanting the effect of control errors (measurement noise) to be small). For example, this is the case in a distillation column when we use temperatures inside the column (y_2) for indirect control of the product compositions (y_1). For a high-purity separation, we cannot place the measurement close to the column end due to sensitivity to measurement error ($\|P_r\|$ becomes large), and we cannot place it far from the column end due to sensitivity to disturbances ($\|P_d\|$ becomes large); see also Example 10.9 (page 409).

Remark 2 Indirect control is related to the idea of *inferential control* which is commonly used in the process industry. However, with inferential control the idea is usually to use the measurement of y_2 to estimate (infer) y_1 and then to control this estimate rather than controlling y_2 directly, e.g. see Stephanopoulos (1984). However, there is no universal agreement on these terms, and Marlin (1995) uses the term inferential control to mean indirect control as discussed above.

Optimal "stabilizing" control in terms of minimizing drift

A primary objective of the regulatory control system is to "stabilize" the plant in terms of minimizing its steady-state drift from a nominal operating point. To quantify this, let w represent the variables in which we would like to avoid drift; for example, w could be the weighted states of the plant. For now let y denote the available measurements and u the manipulated variables to be used for stabilizing control. The problem is: to minimize the drift, which variables c should be controlled (at constant setpoints) by u? We assume linear measurement combinations,

$$c = Hy \tag{10.33}$$

and that we control as many variables as the number of degrees of freedom, $n_c = n_u$. The linear model is

$$w = G^w u + G_d^w d = \widetilde{G}^w \begin{bmatrix} u \\ d \end{bmatrix}$$

$$y = G^y u + G_d^y d = \widetilde{G}^y \begin{bmatrix} u \\ d \end{bmatrix}$$

With perfect regulatory control ($c = 0$), the closed-loop response from d to w is

$$w = P_d^w d; \quad P_d^w = G_d^w - G^w (HG^y)^{-1} HG_d^y$$

Since generally $n_w > n_u$, we do not have enough degrees of freedom to make $w = 0$ ("zero drift"). Instead, we seek the least squares solution that minimizes $\|w\|_2$. In the absence of implementation error, an explicit solution, which also minimizes $\|P_d^w\|_2$, is

$$H = (G^w)^T \widetilde{G}^w (\widetilde{G}^y)^\dagger \tag{10.34}$$

where we have assumed that we have enough measurements, $n_y \geq n_u + n_d$.

Proof of (10.34): We want to minimize

$$J = \|w\|_2^2 = u^T (G^w)^T G^w u + d^T (G_d^w)^T G_d^w d + 2u^T (G^w)^T G_d^w d$$

Then,

$$dJ/du = 2(G^w)^T G^w u + 2(G^w)^T G_d^w d = 2(G^w)^T \widetilde{G}^w \begin{bmatrix} u \\ d \end{bmatrix}$$

An ideal "self-optimizing" variable is $c = dJ/du$, as then $c = 0$ is always optimal with zero loss (in the absence of implementation error). Now, $c = Hy = H\widetilde{G}^y \begin{bmatrix} u \\ d \end{bmatrix}$, so to get $c = dJ/du$, we would like

$$H\widetilde{G}^y = (G^w)^T \widetilde{G}^w \tag{10.35}$$

(the factor 2 does not matter). Since $n_y \geq n_u + n_d$, (10.35) has an infinite number of solutions, and the one using the right inverse of $\widetilde{G}^y$ is given by (10.34). It can be shown that the use of the right inverse is optimal in terms of minimizing the effect of the (until now neglected) implementation error on w, provided the measurements (y) have been normalized (scaled) with respect to their expected measurement error (n^y) (Alstad, 2005, p. 52). The result (10.34) was originally proved by Hori et al. (2005), but this proof is due to V. Kariwala.

□

H computed from (10.34) will be dynamic (frequency-dependent), but for practical purposes, we recommend that it is evaluated at the closed-loop bandwidth frequency of the outer loop that adjusts the setpoints for r. In most cases. it is acceptable to use the steady-state matrices.

Example 10.11 Combination of measurements for minimizing drift of distillation column. *We consider the distillation column (column "A") with the LV-configuration and use the same data as in Example 10.9 (page 409). The objective is to minimize the steady-state drift of the* 41 *composition variables (w = states) due to variations in the feed rate and feed composition by controlling a combination of the available temperature measurements. We have $u = L$, $n_u = 1$ and $n_d = 2$ and we need at least $n_u + n_d = 1+2 = 3$ measurements to achieve zero loss (see null space method, page 397). We select three temperature measurements (y) at stages 15, 20 and 26. One reason for not selecting the measurements located at the column ends is their sensitivity to implementation error, see Example 10.9. By ignoring the implementation error, the optimal combination of variables that minimizes $\|P_d^w(0)\|_2$ is, from (10.34),*

$$c = 0.719T_{15} - 0.018T_{20} + 0.694T_{26}$$

When c is controlled perfectly at $c_s = 0$, this gives $\bar{\sigma}(P_d^w(0)) = 0.363$. This is significantly smaller than $\bar{\sigma}(G_d^w(0)) = 9.95$, which is the "open-loop" deviation of the state variables due to the disturbances. We have not considered the effect of implementation error so far. Similar to (10.28), it can be shown that the effect of implementation error on w is given by $\bar{\sigma}(G_w(G_y)^\dagger)$. With an implementation error of 0.05 *in the individual temperature measurements, we get $\bar{\sigma}(G_w(G_y)^\dagger) = 0.135$, which is small.*

10.5 Control configuration elements

In this section, we discuss in more detail some of the control configuration elements mentioned above. We assume that the measurements y, manipulations u and controlled outputs z are fixed. The available synthesis theories presented in this book result in a multivariable controller K which connects all available measurements/commands (y) with all available manipulations (u),

$$u = Ky \tag{10.36}$$

However, such a “big” (full) controller may not be desirable. By control configuration selection we mean the partitioning of measurements/commands and manipulations within the control layer. More specifically, we define

> **Control configuration.** *The restrictions imposed on the overall controller K by decomposing it into a set of local controllers (subcontrollers, units, elements, blocks) with predetermined links and with a possibly predetermined design sequence where subcontrollers are designed locally.*

In a conventional feedback system, a typical restriction on K is to use a one degree-of-freedom controller (so that we have the same controller for r and $-y$). Obviously, this limits the achievable performance compared to that of a two degrees-of-freedom controller. In other cases, we may use a two degrees-of-freedom controller, but we may impose the restriction that the feedback part of the controller (K_y) is first designed locally for disturbance rejection, and then the prefilter (K_r) is designed for command tracking. In general, this will limit the achievable performance compared to a simultaneous design (see also the remark on page 111). Similar arguments apply to other cascade schemes.

Some elements used to build up a specific control configuration are:

- Cascade controllers
- Decentralized controllers
- Feedforward elements
- Decoupling elements
- Selectors

These are discussed in more detail below, and in the context of the process industry in Shinskey (1967, 1996) and Balchen and Mumme (1988). First, some definitions:

> **Decentralized control** *is when the control system consists of independent feedback controllers which interconnect a subset of the output measurements/commands with a subset of the manipulated inputs. These subsets should not be used by any other controller.*

This definition of decentralized control is consistent with its use by the control community. In decentralized control, we may rearrange the ordering of measurements/commands and manipulated inputs such that the feedback part of the overall controller K in (10.36) has a fixed block-diagonal structure.

> **Cascade control** *arises when the output from one controller is the input to another.* This is broader than the conventional definition of cascade control which is that the output from one controller is the reference command (setpoint) to another. In addition, in cascade control, it is usually assumed that the inner loop (K_2) is much faster than the outer loop (K_1).

Feedforward elements *link measured disturbances to manipulated inputs.*

Decoupling elements *link one set of manipulated inputs ("measurements") with another set of manipulated inputs. They are used to improve the performance of decentralized control systems, and are often viewed as feedforward elements (although this is not correct when we view the control system as a whole) where the "measured disturbance" is the manipulated input computed by another decentralized controller.*

Selectors *are used to select for control, depending on the conditions of the system, a subset of the manipulated inputs or a subset of the outputs.*

In addition to restrictions on the structure of K, we may impose restrictions on the way, or rather in which *sequence*, the subcontrollers are designed. For most decomposed control systems we design the controllers sequentially, starting with the "fast" or "inner" or "lower-layer" control loops in the control hierarchy. Since cascade and decentralized control systems depend more strongly on feedback rather than models as their source of information, it is usually more important (relative to centralized multivariable control) that the fast control loops are tuned to respond quickly.

In this section, we discuss cascade controllers and selectors, and in the following section, we consider decentralized diagonal control. Let us first give some justification for using such "suboptimal" configurations rather than directly designing the overall controller K.

10.5.1 Why use simplified control configurations?

Decomposed control configurations can be quite complex, see for example Figure 10.13 (page 427), and it may therefore be both simpler and better in terms of control performance to set up the controller design problem as an optimization problem and let the computer do the job, resulting in a centralized multivariable controller as used in other chapters of this book.

If this is the case, why are simplified parameterizations (e.g. PID) and control configurations (e.g. cascade and decentralized control) used in practice? There are a number of reasons, but the most important one is probably the cost associated with obtaining good plant models, which are a prerequisite for applying multivariable control. On the other hand, with cascade and decentralized control the controllers are usually tuned one at a time with a minimum of modelling effort, sometimes even *on-line* by selecting only a few parameters (e.g., the gain and integral time constant of a PI controller). Thus:

- *A fundamental reason for applying cascade and decentralized control is to save on modelling effort.*

Other *benefits* of cascade and decentralized control may include the following:

- easy for operators to understand
- ease of tuning because the tuning parameters have a direct and "localized" effect
- insensitive to uncertainty, e.g. in the input channels
- failure tolerance and the possibility of taking individual control elements into or out of service
- few control links and the possibility for simplified (decentralized) implementation
- reduced computation load

The latter two benefits are becoming less relevant as the cost of computing power is reduced. Based on the above discussion, the main challenge is to find a *control configuration* which allows the (sub)controllers to be tuned independently based on a minimum of model information (the pairing problem). For industrial problems, the number of possible pairings is usually very high, but in most cases physical insight and simple tools, such as the RGA, can be helpful in reducing the number of options to a manageable number. To be able to tune the controllers independently, we must require that the loops interact only to a limited extent. For example, one desirable property is that the steady-state gain from u_i to y_i in an "inner" loop (which has already been tuned) does not change too much as outer loops are closed. For decentralized diagonal control the RGA is a useful tool for addressing this pairing problem (see page 450).

Remark. We just argued that the main advantage of applying cascade and decentralized control is that the controllers can be tuned on-line and this saves on the modelling effort. However, in our theoretical treatment we need a model, for example, to decide on a control configuration. This seems to be a contradiction, but note that the model required for selecting a configuration may be more "generic" and does not need to be modified for each particular application. Thus, if we have found a good control configuration for one particular applications, then it is likely that it will work well also for similar applications.

10.5.2 Cascade control systems

We want to illustrate how a control system which is decomposed into subcontrollers can be used to solve multivariable control problems. For simplicity, we use SISO controllers here of the form

$$u_i = K_i(s)(r_i - y_i) \tag{10.37}$$

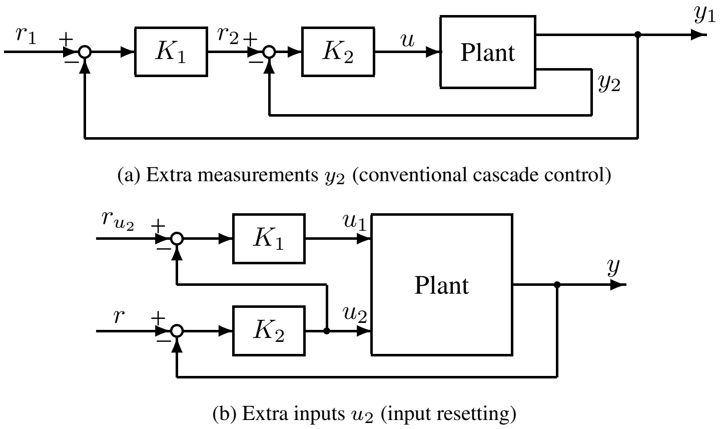
where $K_i(s)$ is a scalar. Note that whenever we close a SISO control loop we lose the corresponding input, u_i, as a degree of freedom, but at the same time the reference, r_i, becomes a new degree of freedom.

It may look like it is not possible to handle non-square systems with SISO controllers. However, since the input to the controller in (10.37) is a reference minus a measurement, we can cascade controllers to make use of extra measurements or extra inputs. A *cascade control structure* results when either of the following two situations arise:

- The reference r_i is an output from another controller (typically used for the case of an extra measurement y_i), see Figure 10.10(a). This is *conventional cascade control*.
- The "measurement" y_i is an output from another controller (typically used for the case of an extra manipulated input u_j, e.g. in Figure 10.10(b) where u_2 is the "measurement" for controller K_1). This cascade scheme where the "extra" input u_2 is used to improve the dynamic response, but is reset to a desired "mid-range" target value on a longer time scale, is referred to as *input resetting* (also known as *mid-ranging* or valve position control).

10.5.3 Extra measurements: cascade control

In many cases, we make use of extra measurements y_2 (*secondary outputs*) to provide local disturbance rejection and linearization, or to reduce the effects of measurement noise. For example, velocity feedback is frequently used in mechanical systems, and local flow cascades



(a) Extra measurements y_2 (conventional cascade control)

(b) Extra inputs u_2 (input resetting)

Figure 10.10: Cascade implementations

are used in process systems. For distillation columns, it is usually recommended to close an inner temperature loop $(y_2 = T)$, see Example 10.9.

A typical implementation with two cascaded SISO controllers is shown in Figure 10.10(a) where

$$r_2 = K_1(s)(r_1 - y_1) \tag{10.38}$$

$$u = K_2(s)(r_2 - y_2) \tag{10.39}$$

u is the manipulated input, y_1 the controlled output (with an associated control objective r_1) and y_2 the extra measurement. Note that the output r_2 from the slower *primary* controller K_1 is not a manipulated *plant* input, but rather the reference input to the faster *secondary* (or slave) controller K_2. For example, cascades based on measuring the actual manipulated variable (in which case $y_2 = u_m$) are commonly used to reduce uncertainty and nonlinearity at the plant input.

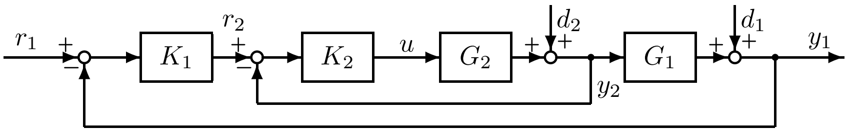


Figure 10.11: Common case of cascade control where the primary output y_1 depends directly on the extra measurement y_2

In the general case, y_1 and y_2 in Figure 10.10(a) are not directly related to each other, and this is sometimes referred to as *parallel cascade control*. However, it is common to encounter the situation in Figure 10.11 where y_1 depends directly on y_2. This is a special case of Figure 10.10(a) with "Plant" $= \begin{bmatrix} G_1G_2 \\ G_2 \end{bmatrix}$, and it is considered further in Example 10.12 and Exercise 10.7.

Remark. Centralized (parallel) implementation. Alternatively, we may use a centralized implementation $u = K(r - y)$ where K is a 2-input 1-output controller. This gives

$$u = K_{11}(s)(r_1 - y_1) + K_{12}(s)(r_2 - y_2) \tag{10.40}$$

where in most cases $r_2 = 0$ (since we do not have a degree of freedom to control y_2). With $r_2 = 0$ in (10.40) the relationship between the centralized and cascade implementations is $K_{11} = K_2 K_1$ and $K_{12} = K_2$.

An advantage with the cascade implementation is that it more clearly decouples the design of the two controllers. It also shows that r_2 is not a degree of freedom at higher layers in the control system. Finally, it allows for integral action in both loops (whereas usually only K_{11} would have integral action in (10.40)). On the other hand, a centralized implementation is better suited for direct multivariable synthesis; see the velocity feedback for the helicopter case study in Section 13.2.

When should we use cascade control? With reference to the special (but common) case of conventional cascade control shown in Figure 10.11, Shinskey (1967, 1996) states that the principal advantages of cascade control are:

1. Disturbances arising within the secondary loop (before y_2 in Figure 10.11) are corrected by the secondary controller before they can influence the primary variable y_1.
2. Phase lag existing in the secondary part of the process (G_2 in Figure 10.11) is reduced measurably by the secondary loop. This improves the speed of response of the primary loop.
3. Gain variations in the secondary part of the process are overcome within its own loop.

Morari and Zafiriou (1989) conclude, again with reference to Figure 10.11, that the use of an extra measurement y_2 is useful under the following circumstances:

(a) The disturbance d_2 (entering before the measurement y_2) is significant and G_1 is non-minimum-phase – e.g. G_1 contains an effective time delay [see Example 10.12].
(b) The plant G_2 has considerable uncertainty associated with it – e.g. G_2 has a poorly known nonlinear behaviour – and the inner loop serves to remove the uncertainty.

In terms of design, they recommended that K_2 is first designed to minimize the effect of d_2 on y_1 (with $K_1 = 0$) and then K_1 is designed to minimize the effect of d_1 on y_1.

An example where local feedback control is required to counteract the effect of high-order lags is given for a neutralization process in Figure 5.25 on page 216. The benefits of local feedback are also discussed by Horowitz (1991).

Exercise 10.7 *We want to derive the above conclusions (a) and (b) from an input–output controllability analysis, and also explain (c) why we may choose to use cascade control if we want to use simple controllers (even with $d_2 = 0$).*

Outline of solution: *(a) Note that if G_1 is minimum-phase, then the input–output controllability of G_2 and $G_1 G_2$ are in theory the same, and for rejecting d_2 there is no fundamental advantage in measuring y_1 rather than y_2. (b) The inner loop $L_2 = G_2 K_2$ removes the uncertainty if it is sufficiently fast (high-gain feedback). It yields a transfer function $(I + L_2)^{-1} L_2$ which is close to I at frequencies where K_1 is active. (c) In most cases, such as when PID controllers are used, the practical closed-loop bandwidth is limited approximately by the frequency w_u, where the phase of the plant is -180° (see Section 5.8 on page 191), so an inner cascade loop may yield faster control (for rejecting d_1 and tracking r_1) if the phase of G_2 is less than that of $G_1 G_2$.*

Tuning of cascaded PID controllers using the SIMC rules. Recall the SIMC PID procedure presented on page 57, where the idea is to tune the controllers such that the resulting transfer function from r to y is $T \approx \frac{e^{-\theta s}}{\tau_c s+1}$. Here, θ is the effective delay in G (from u to y) and τ_c is a tuning parameter with $\tau_c = \theta$ being selected for fast (and still robust) control. Let us apply this approach to the cascaded system in Figure 10.11. The inner loop (K_2) is tuned based on G_2. We then get $y_2 = T_2 r_2$, where $T_2 \approx \frac{e^{-\theta_2 s}}{\tau_{c2}s+1}$ and θ_2 is the effective delay in G_2. Since the inner loop is fast (θ_2 and τ_{c2} are small), its response may be approximated as a pure time delay for the tuning of the slower outer loop (K_1),

$$T_2 \approx 1 \cdot e^{-(\theta_2+\tau_{c2})s} \tag{10.41}$$

The resulting model for tuning of the outer loop (K_1) is then

$$\widetilde{G}_1 = G_1 T_2 \approx G_1 e^{-(\theta_2+\tau_{c2})s} \tag{10.42}$$

and the PID tuning parameters for K_1 are easily obtained using the SIMC rules. For a "fast response" from r_2 to y_2 in the inner loop, the SIMC-rule is to select $\tau_{c2} = \theta_2$. However, this may be unnecessarily fast and to improve robustness we may want to select a larger τ_{c2}. Its value will not affect the outer loop, provided $\tau_{c2} < \tau_{c1}/5$ approximately, where τ_{c1} is the response time in the outer loop.

Example 10.12 *Consider the closed-loop system in Figure 10.11, where*

$$G_1 = \frac{(-0.6s+1)}{(6s+1)}e^{-s} \quad \text{and} \quad G_2 = \frac{1}{(6s+1)(0.4s+1)}$$

We £rst consider the case where we only use the primary measurement (y_1), i.e. design the controller based on $G = G_1G_2$. Using the half rule on page 57, we £nd that the effective delay is $\theta_1 = 6/2+0.4+0.6+1 = 5$, and using the SIMC tuning rules on page 57, a PI controller is designed with $K_c = 0.9$ and $\tau_I = 9$. The closed-loop response of the system to step changes of magnitude 1 in the setpoint (at $t = 0$) and of magnitude 6 in disturbance d_2 (at $t = 50$) is shown in Figure 10.12. From the dashed line, we see that the closed-loop disturbance rejection is poor.

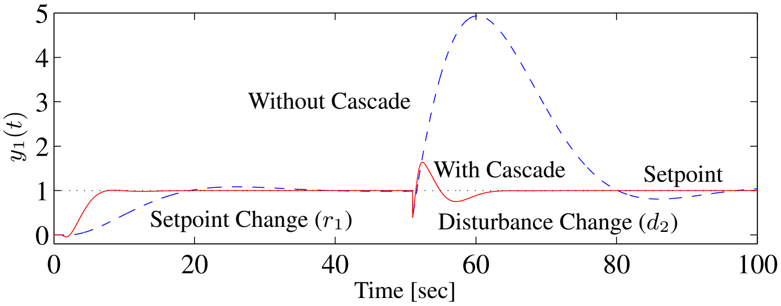


Figure 10.12: Improved control performance with cascade control (solid) as compared to single-loop control (dashed)

Next, to improve disturbance rejection, we make use of the measurement y_2 in a cascade implementation as shown in Figure 10.11. First, the PI controller for the inner loop is designed based on G_2. The effective delay is $\theta_2 = 0.2$. For "fast control" the SIMC rule (page 57) is to use $\tau_{c2} = \theta_2$. However, since this is an inner loop, where tight control is not critical, we choose $\tau_{c2} = 2\theta_2 = 0.4$, which gives somewhat less aggressive settings with $K_{c2} = 10.33$ and $\tau_{I2} = 2.4$. The PI controller for the outer loop is next designed with the inner loop closed. From (10.41), the transfer function for the inner loop is approximated as a delay of $\tau_{c2}+\theta_2 = 0.6$ giving $\widetilde{G}_1 = G_1 e^{-0.6s} = \frac{(-0.6s+1)}{(6s+1)}e^{-1.6s}$. Thus, for the outer loop, the effective delay is $\theta_1 = 0.6+1.6 = 2.2$ and with $\tau_{c1} = \theta_1 = 2.2$ ("fast *control"), the resulting SIMC PI tunings are $K_{c1} = 1.36$ and $\tau_{I1} = 6$. From Figure 10.12, we note that the cascade controller greatly improves the rejection of d_2. The speed of the setpoint tracking is also improved, because the local control (K_2) reduces the effective delay for control of y_1.*

Exercise 10.8 *To illustrate the bene£t of using inner cascades for high-order plants, consider Figure 10.11 and a plant $G = G_1G_2G_3G_4G_5$ with*

$$G_1 = G_2 = G_3 = G_4 = G_5 = \frac{1}{s+1}$$

Consider the following two cases:

(a) Measurement of y_1 only, i.e. $G = \frac{1}{(s+1)^5}$.

(b) Four additional measurements available (y_2, y_3, y_4, y_5) on outputs of G_1, G_2, G_3 and G_4.

For case (a) design a PID controller and for case (b) use £ve simple proportional controllers with gains with gains 10 (innermost loop), 5, 2, 1 and 0.5 (outer loop) (note that the gain has to be smaller in the outer loop to avoid instability caused by the effective delay in the inner loop). For case (b) also try using a PI controller in the outer loop to avoid the steady-state offset. Compare the responses to disturbances entering before G_1 (at $t = 0$), G_2 ($t = 20$), G_3 ($t = 40$), G_4 ($t = 60$), G_5 ($t = 80$), and for a setpoint change ($t = 100$)".

10.5.4 Extra inputs

In some cases, we have more manipulated inputs than controlled outputs. These may be used to improve control performance. Consider a plant with a single controlled output y and two manipulated inputs u_1 and u_2. Sometimes u_2 is an extra input which can be used to improve the fast (transient) control of y, but if it does not have suf£cient power or is too costly to use for long-term control, then after a while it is reset to some desired value ("ideal resting value").

Cascade implementation (input resetting). An implementation with two cascaded SISO controllers is shown in Figure 10.10(b). We let input u_2 take care of the fast control and u_1 the long-term control. The fast control loop is then

$$u_2 = K_2(s)(r - y) \tag{10.43}$$

The objective of the other slower controller is then to use input u_1 to reset input u_2 to its desired value r_{u_2}:

$$u_1 = K_1(s)(r_{u_2} - y_1), \quad y_1 = u_2 \tag{10.44}$$

and we see that the output u_2 from the fast controller K_2 is the "measurement" y_1 for the slow controller K_1.

In process control, the cascade implementation with input resetting often involves *valve position control*, because the extra input u_2, usually a valve, is reset to a desired position by the outer cascade.

Centralized (parallel) implementation. Alternatively, we may use a centralized implementation $u = K(r - y)$ where K is a 1-input 2-output controller. This gives

$$u_1 = K_{11}(s)(r - y), \quad u_2 = K_{21}(s)(r - y) \tag{10.45}$$

Here two inputs are used to control one output, so to get a unique steady-state for the inputs u_1 and u_2 we usually let K_{11} have integral control, whereas K_{21} does not. Then $u_2(t)$ will only be used for transient (fast) control and will return to zero (or more precisely to its desired value r_{u_2}) as $t \to \infty$. With $r_{u_2} = 0$ the relationship between the centralized and cascade implementation is $K_{11} = -K_1 K_2$ and $K_{21} = K_2$.

Comparison of cascade and centralized implementations. The cascade implementation in Figure 10.10(b) has the advantage, compared to the centralized (parallel) implementation, of decoupling the design of the two controllers. It also shows more clearly that r_{u_2}, the reference for u_2, may be used as a degree of freedom at higher layers in the control system. Finally, we can have integral action in both K_1 and K_2, but note that the gain of K_1 should be negative (if effects of u_1 and u_2 on y are both positive).

Exercise 10.9 * *Draw the block diagrams for the two centralized (parallel) implementations corresponding to Figure 10.10.*

Exercise 10.10 *Derive the closed-loop transfer functions for the effect of r on y, u_1 and u_2 in the cascade input resetting scheme of Figure 10.10(b). As an example use $G = [\,G_{11} \quad G_{12}\,] = [\,1 \quad 1\,]$ and use integral action in both controllers, $K_1 = -1/s$ and $K_2 = 10/s$. Show that input u_2 is reset at steady-state.*

10.5.5 Extra inputs and outputs

In some cases performance may be improved with local control loops involving both extra manipulated inputs and extra measurements. However, as always, the improvement must be traded off against the cost of the extra actuators, measurements and control system.

Example 10.13 **Two layers of cascade control.** *Consider the system in Figure 10.13 with two manipulated plant inputs (u_2 and u_3), one controlled output (y_1, which should be close to r_1) and two measured variables (y_1 and y_2). Input u_2 has a more direct effect on y_1 than does input u_3 (since there is a large delay in $G_3(s)$). Input u_2 should only be used for transient control as it is desirable that it remains close to $r_3 = r_{u_2}$. The extra measurement y_2 is closer than y_1 to the input u_2 and may be useful for detecting disturbances (not shown) affecting G_1.*

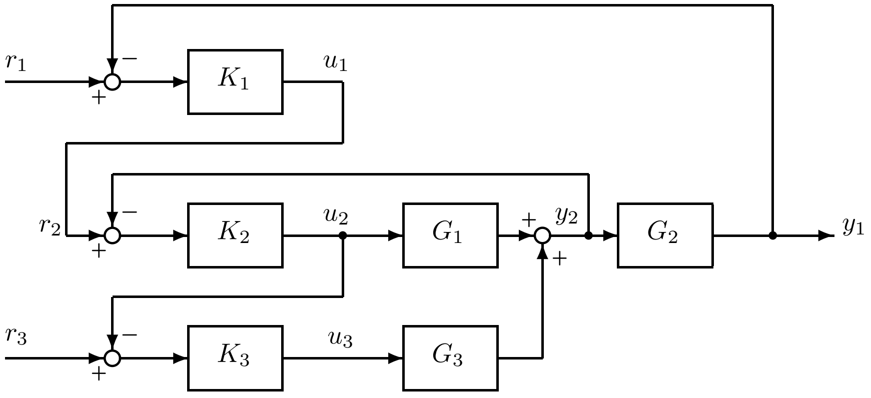


Figure 10.13: Control con£guration with two layers of cascade control

In Figure 10.13, controllers K_1 and K_2 are cascaded in a conventional manner, whereas controllers K_2 and K_3 are cascaded to achieve input resetting. The "input" u_1 is not a (physical) plant input, but it *does play the role of an input (manipulated variable) as seen from the controller K_1. The corresponding equations are*

$$\begin{aligned} u_1 &= K_1(s)(r_1 - y_1) && (10.46)\\ u_2 &= K_2(s)(r_2 - y_2), \quad r_2 = u_1 && (10.47)\\ u_3 &= K_3(s)(r_3 - y_3), \quad y_3 = u_2 && (10.48) \end{aligned}$$

Controller K_1 controls the primary output y_1 at its reference r_1 by adjusting the "input" u_1, which is the reference value for y_2. Controller K_2 controls the secondary output y_2 using input u_2. Finally, controller K_3 manipulates u_3 slowly in order to reset input u_2 to its desired value r_3.

Typically, the controllers in a cascade system are tuned one at a time starting with the fastest loop. For example, for the control system in Figure 10.13 we would probably tune the three controllers in the order K_2 (inner cascade using fast input), K_3 (input resetting using slower input), and K_1 (final adjustment of y_1).

Exercise 10.11 * **Process control application.** *A practical case of a control system like the one in Figure 10.13 is in the use of a pre-heater to keep a reactor temperature y_1 at a given value r_1. In this case, y_2 may be the outlet temperature from the pre-heater, u_2 the bypass flow (which should be reset to r_3, say 10% of the total flow), and u_3 the flow of heating medium (steam). Process engineering students: Make a process flowsheet with instrumentation lines (not a block diagram) for this heater/reactor process.*

10.5.6 Selectors

Split-range control for extra inputs. We considered above the case where the primary input is "slow", and an extra input is added to improve the dynamic performance. For economic reasons or to avoid saturation the extra input is reset to a desired "mid-range" target value on a longer time scale (input resetting or mid-ranging). Another situation is when the primary input may saturate, and an extra input is added to maintain control of the output. In this case, the control range is often split such that, for example, u_1 is used for control when $y \in [y_{\min}, y_1]$, and u_2 is used when $y \in [y_1, y_{\max}]$.

Selectors for too few inputs. A completely different situation occurs if there are too few inputs. Consider the case with one input (u) and several outputs ($y_1, y_2, \ldots$). In this case, we cannot control all the outputs independently, so we either need to control all the outputs in some average manner, or we need to make a choice about which outputs are the most important to control. Selectors or logic switches are often used for the latter. *Auctioneering selectors* are used to decide to control one of several similar outputs. For example, such a selector may be used to adjust the heat input (u) to keep the maximum temperature ($\max_i y_i$) in a fired heater below some value. *Override selectors* are used when several controllers compute the input value, and we select the smallest (or largest) as the input. For example, this is used in a heater where the heat input (u) normally controls temperature (y_1), except when the pressure (y_2) is too large and pressure control takes over.

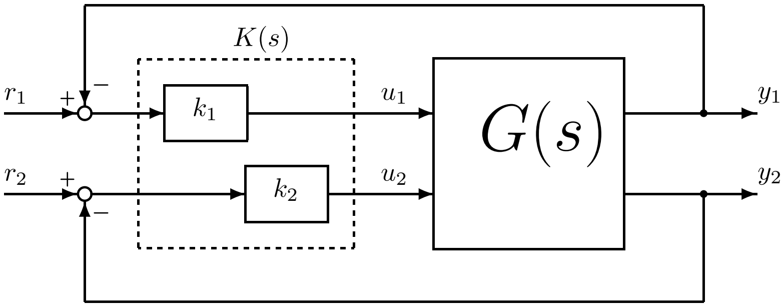


Figure 10.14: Decentralized diagonal control of a 2 × 2 plant

10.6 Decentralized feedback control

10.6.1 Introduction

We have already discussed, in the previous sections on control con£gurations, the use of decentralized control, but here we consider it in more detail. To this end, we assume in this section that $G(s)$ is a square plant which is to be controlled using a diagonal controller (see Figure 10.14)

$$K(s) = \mathrm{diag}\{k_i(s)\} = \begin{bmatrix} k_1(s) & & & \\ & k_2(s) & & \\ & & \ddots & \\ & & & k_m(s) \end{bmatrix} \tag{10.49}$$

This is the problem of decentralized (or diagonal) feedback control.

It may seem like the use of decentralized control seriously limits the achievable control performance. However, often the performance loss is small, partly because of the bene£ts of high-gain feedback. For example, it can be proved theoretically (Zames and Bensoussan, 1983) that with decentralized control one may achieve perfect control of all outputs, provided the plant has no RHP-zeros that limit the use of high feedback gains. Furthermore, for a stable plant $G(s)$ (also with RHP-zeros), it is possible to use integral control in all channels (to achieve perfect steady-state control) if and only if $G(0)$ is non-singular (Campo and Morari, 1994). Both these conditions are also required with full multivariable control. Nevertheless, for “interactive” plants and £nite bandwidth controllers, there is a performance loss with decentralized control because of the interactions caused by non-zero off-diagonal elements in G. The interactions may also cause stability problems. A key element in decentralized control is therefore to select good “pairings” of inputs and outputs, such that the effect of the interactions is minimized.

The design of decentralized control systems typically involves two steps:

1. The choice of pairings (control con£guration selection).
2. The design (tuning) of each controller, $k_i(s)$.

The optimal solution to this problem is very dif£cult mathematically. First, the number of pairing options in step 1 is $m!$ for an $m \times m$ plant and thus increases *exponentially* with the size of the plant. Second, the optimal controller in step 2 is in general of in£nite order and may be non-unique. In step 2, there are three main approaches:

Fully coordinated design. All the diagonal controller elements $k_i(s)$ are designed simultaneously based on the complete model $G(s)$. This is the theoretically optimal approach for decentralized control, but it is not commonly used in practice. First, as just mentioned, the design problem is very dif£cult. Second, it offers few of the "normal" bene£ts of decentralized control (see page 421), such as ease of tuning, reduced modelling effort, and good failure tolerance. In fact, since a detailed dynamic model is required for the design, an optimal coordinated decentralized design offers few bene£ts compared to using a "full" multivariable controller which is easier to design and has better performance. The exception is situations where multivariable control cannot be used, for example, when centralized cooordination is dif£cult for geographical reasons. We do not address the optimal coordinated design of decentralized controllers in this book, and the reader is referred to the literature (e.g. Sourlas and Manousiouthakis, 1995) for more details.

Independent design. Each controller element $k_i(s)$ is designed based on the corresponding diagonal element of $G(s)$, such that each individual loop is stable. Possibly, there is some consideration of the off-diagonal interactions when tuning each loop. This approach is the main focus in the remaining part of this chapter. It is used when it is desirable that we have *integrity* where the individual parts of the system (including each loop) can operate independently. The pairing rules on page 450 can be used to obtain pairings for independent design. In short the rules are to (1) pair on RGA elements close to 1 at crossover frequencies, (2) pair on positive steady-state RGA elements, and (3) pair on elements that impose minimal bandwidth limitations (e.g., small delay). The £rst and second rules are to avoid that the interactions cause instability. The third rule follows because we for good performance want to use high-gain feedback, but we require stable individual loops. For many interactive plants, it is not possible to £nd a set of pairing satisfying all the three rules.

Sequential design. The controllers are designed sequentially, one at a time, with the previously designed ("inner") controllers implemented. This has the important advantage of reducing each design to a scalar (SISO) problem, and is well suited for on-line tuning. The sequential design approach can be used for interactive problems where the independent design approach does not work, provided it is acceptable to have "slow" control of some output so that we get a difference in the closed-loop response times of the outputs. One then starts by closing the fast "inner" loops (involving the outputs with the fastest desired response times), and continues by closing the slower "outer" loops. The main disadvantage with this approach is that failure tolerance is not guaranteed when the inner loops fail (integrity). In particular, the individual loops are not guaranteed to be stable. Furthermore, one has to decide on the order in which to close the loops.

The effective use of a decentralized controller requires some element of decoupling. Loosely speaking, *independent design* is used when the system is decoupled in space ($G(s)$ is close to diagonal), whereas *sequential design* is used when the system outputs can be decoupled in time.

The analysis of *sequentially designed* decentralized control systems may be performed using the results on partial control presented earlier in this chapter. For example, after closing the inner loops (from u_2 to y_2), the transfer function for the remaining outer system (from u_1 to y_1) is $P_u = \big(G_{11} - G_{12}K_2(I + G_{22}K_2)^{-1}G_{21}\big)$; see (10.26). Notice that in the general case we need to take into account the details of the controller K_2. However, when there is a time scale separation between the layers with the fast loops (K_2) being closed £rst, then we may for the design of K_1 assume $K_2 \to \infty$ ("perfect control of y_2"), and the transfer function for the remaining "slow" outer system becomes $P_u = G_{11} - G_{12}G_{22}^{-1}G_{21}$; see (10.28). The advantages of the time scale separation for sequential design of decentralized controllers (with fast "inner" and slow "outer" loops), are the same as those for hierarchical cascade control (with fast "lower" and slow "upper" layers) as listed on page 387. Examples of sequential design are given in Example 10.15 (page 433) and in Section 10.6.6 (page 446).

The relative gain array (RGA) is a very useful tool for decentralized control. It is de£ned as $\Lambda = G \times (G^{-1})^T$, where × denotes element-by-element multiplication. It is recommended to read the discussion about the "original interpretation" of the RGA on page 83, before continuing. Note in particular from (3.56) that each RGA element represents the ratio between the open-loop (g_{ij}) and "closed-loop" ($\widehat{g}_{ij}$) gains for the corresponding input-output pair, $\lambda_{ij} = g_{ij}/\widehat{g}_{ij}$. By "closed-loop" here we mean "partial control with the other outputs perfectly controlled". Intuitively, we would like to pair on elements with $\lambda_{ij}(s)$ close to 1, because then the transfer function from u_j to y_i is unaffected by closing the other loops.

Remark. We assume in this section that the decentralized controllers $k_i(s)$ are scalar. The treatment may be generalized to block-diagonal controllers by, for example, introducing tools such as the block relative gain; e.g., see Manousiouthakis et al. (1986) and Kariwala et al. (2003).

10.6.2 Introductory examples

To provide some insight into decentralized control and to motivate the material that follows we start with some simple 2 × 2 examples. We assume that the outputs y_1 and y_2 have been scaled so that the allowable control errors $(e_i = y_i - r_i), i = 1, 2$ are approximately between 1 and -1. We design the decentralized controller to give £rst-order responses with time constant τ_i in each of the individual loops, that is, $y_i = \frac{1}{\tau_i s+1} r_i$. For simplicity, the plants have no dynamics, and the individual controllers are then simple integral controllers $k_i(s) = \frac{1}{g_{ii}}\frac{1}{\tau_i s}$; see the IMC design procedure on page 54. To make sure that we do not use aggressive control, we use (in all simulations) a "real" plant, where we add a delay of 0.5 time units in each output, i.e. $G_{\rm sim} = Ge^{-0.5s}$. This delay is not included in the analytic expressions, e.g. (10.52), in order to simplify our discussion, but it is included for simulation and tuning. With a delay of 0.5 we should, for stability and acceptable robustness, select $\tau_i \geq 1$; see the SIMC rule for "fast but robust" control on page 57. In all simulations we drive the system with reference changes of $r_1 = 1$ at $t = 0$ and $r_2 = 1$ at $t = 20$.

Example 10.14 **Diagonal plant.** *Consider the simplest case of a diagonal plant*

$$G = \begin{bmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{10.50}$$

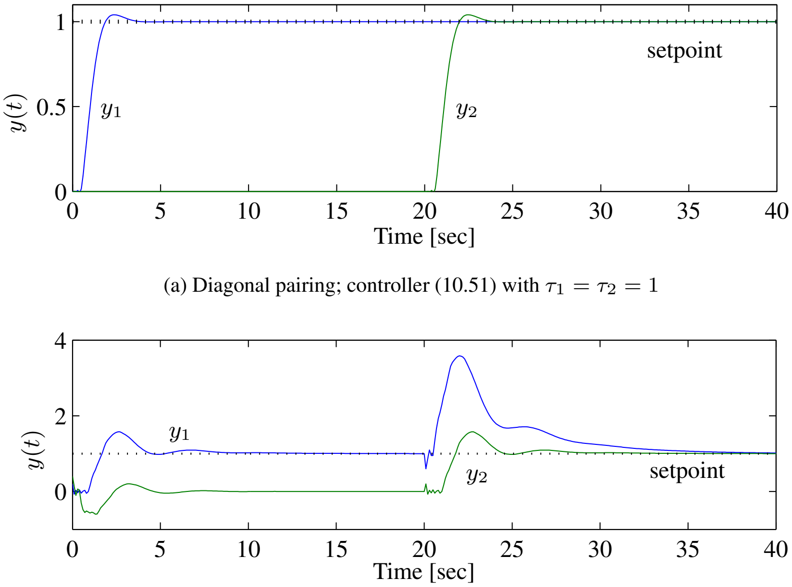
with RGA $= I$. The off-diagonal elements are zero, so there are no interactions and decentralized control with diagonal pairings is obviously optimal.

Diagonal pairings. The controller

$$K = \begin{bmatrix} \frac{1}{\tau_1 s} & 0 \\ 0 & \frac{1}{\tau_2 s} \end{bmatrix} \tag{10.51}$$

gives nice decoupled £rst-order responses

$$y_1 = \frac{1}{\tau_1 s + 1} r_1 \quad \text{and} \quad y_2 = \frac{1}{\tau_2 s + 1} r_2 \tag{10.52}$$



(a) Diagonal pairing; controller (10.51) with $\tau_1 = \tau_2 = 1$

(b) Off-diagonal pairing; plant (10.53) and controller (10.54)

Figure 10.15: Decentralized control of diagonal plant (10.50)

as illustrated in Figure 10.15(a) for the case with $\tau_1 = \tau_2 = 1$.

Off-diagonal pairings. *When considering pairings other than diagonal, we recommend to £rst permute the inputs such that the paired elements are along the diagonal. For the off-diagonal pairing, we use the permuted inputs*

$$u_1^* = u_2 \ , \ u_2^* = u_1$$

corresponding to the permuted plant (denoted with **)*

$$G^* = G \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}^T = \begin{bmatrix} g_{12} & g_{11} \\ g_{22} & g_{21} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \tag{10.53}$$

This corresponds to pairing on two zero elements, $g_{11}^* = 0$ *and* $g_{22}^* = 0$*, and we cannot use independent or sequential controller design. A coordinated (simultaneous) controller design is required and after some trial and error we arrived at the following design*

$$K^*(s) = \begin{bmatrix} \frac{-(0.5s+0.1)}{s} & 0 \\ 0 & \frac{(0.5s+2)}{s} \end{bmatrix} \tag{10.54}$$

Performance is of course quite poor as is illustrated in Figure 10.15(b), but it is nevertheless workable (surprisingly!).

Remark. The last example, where a diagonal plant is controlled using the off-diagonal pairings, is quite striking. A simple physical example is the control of temperatures in two unrelated rooms, say one located in the UK (Ian's of£ce) and one in Norway (Sigurd's of£ce). The setup is then that Ian gets a measurement of Sigurd's room temperature, and based on this adjusts the heating in his room (in the UK). Similarly, Sigurd gets a measurement of

Ian's room temperature, and based on this adjusts the heating in his room (in Norway). As shown in Figure 10.15(b), such a ridiculous setup (with $g_{11} = 0$ and $g_2 = 0$) is actually workable because of the "hidden" feedback loop going through the off-diagonal elements and the controllers ($k_1 k_2 g_{12} g_{21}$ is nonzero), provided one is able to tune the controllers k_1 and k_2 (which is not trivial – as seen it requires a negative sign in one of the controllers). Two lessons from this example are that (1) decentralized control can work for almost any plant, and (2) the fact that we have what seems to be acceptable closed-loop performance does not mean that we are using the best pairing.

Exercise 10.12 *Consider in more detail the off-diagonal pairings for the diagonal plant in the example above. (i) Explain why it is necessary to use a negative sign in (10.54). (ii) Show that the plant (10.53) cannot be stabilized by a pure integral action controller of the form $K^*(s) = \mathrm{diag}(\frac{k_i}{s})$.*

Example 10.15 One-way interactive (triangular) plant. *Consider*

$$G = \begin{bmatrix} 1 & 0 \\ 5 & 1 \end{bmatrix} \tag{10.55}$$

for which

$$G^{-1} = \begin{bmatrix} 1 & 0 \\ -5 & 1 \end{bmatrix} \quad \text{and} \quad \mathrm{RGA} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

The RGA matrix is identity, which suggests that the diagonal pairings are best for this plant. However, we see that there is a large interaction $(g_{21} = 5)$ from u_1 to y_2, which, as one might expect, implies poor performance with decentralized control. Note that this is not a fundamental control limitation as the decoupling controller $K(s) = \frac{1}{s}\begin{bmatrix} 1 & 0 \\ -5 & 1 \end{bmatrix}$ gives nice decoupled responses, identical to those shown in Figure10.15 (but the decoupler may be sensitive to uncertainty; see Exercise 10.13).

Diagonal pairings using independent design. If we use independent design based on the paired (diagonal) elements only (without considering the interactions caused by $g_{21} = 5 \neq 0$), then the controller becomes

$$K = \begin{bmatrix} \frac{1}{\tau_1 s} & 0 \\ 0 & \frac{1}{\tau_2 s} \end{bmatrix} \tag{10.56}$$

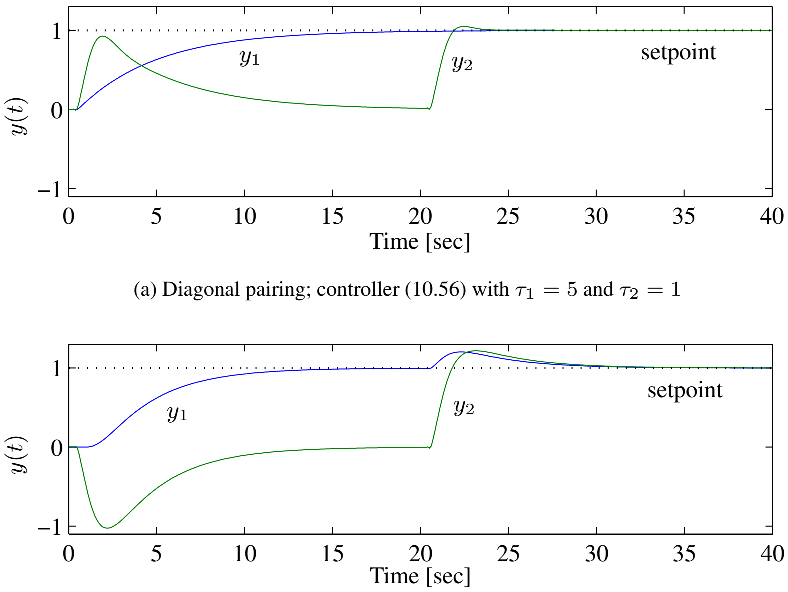
with $\tau_1 = \tau_2 = 1$ (assuming a 0.5 time delay). However, a closer analysis shows that the closed-loop response with the controller (10.56) becomes

$$y_1 = \frac{1}{\tau_1 s + 1} r_1 \tag{10.57}$$

$$y_2 = \frac{5\tau_2 s}{(\tau_1 s + 1)(\tau_2 s + 1)} r_1 + \frac{1}{\tau_2 s + 1} r_2 \tag{10.58}$$

If we plot the interaction term from r_1 to y_2 as a function of frequency, then we £nd that for $\tau_1 = \tau_2$ it has a peak value of about 2.5. Therefore, with this controller the response for y_2 is not acceptable when we make a change in r_1. To keep this peak below 1, we need to select $\tau_1 \geq 5\tau_2$, approximately. This is illustrated in Figure10.16(a) where we have selected $\tau_1 = 5$ and $\tau_2 = 1$. Thus, to keep $|e_2| \leq 1$, we must accept slow control of y_1.

Remark. *The performance problem was not detected from the RGA matrix, because it only measures two-way interactions. However, it may be detected from the "Performance RGA" matrix (PRGA), which for our plant with unity diagonal elements is equal to G^{-1}. As discussed on page 438, a large element in a row of PRGA indicates that fast control is needed to get acceptable reference tracking. Thus, the 2, 1 element in G^{-1} of magnitude 5, con£rms that control of y_2 must be about 5 times faster than that of y_1.*



(a) Diagonal pairing; controller (10.56) with $\tau_1 = 5$ and $\tau_2 = 1$

(b) Off-diagonal pairing; plant (10.59) and controller (10.60) with $\tau_1 = 5$ and $\tau_2 = 1$

Figure 10.16: Decentralized control of triangular plant (10.55)

Off-diagonal pairings using sequential design. The permuted plant is

$$G^* = G \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}^T = \begin{bmatrix} 0 & 1 \\ 1 & 5 \end{bmatrix} \tag{10.59}$$

This corresponds to pairing on a zero element $g^*_{11} = 0$. This pairing is not *acceptable if we use the independent design approach, because u^*_1 has no effect on y_1 so "loop 1" does not work by itself. However, with the sequential design approach, we may £rst close the loop around y_2 (on the element $g^*_{22} = 5$). With the IMC design approach, the controller becomes $k^*_2(s) = 1/(g^*_{22}\tau_2 s) = 1/(5\tau_2 s)$ and with this loop closed, u^*_1 does have an effect on y_1. Assuming tight control of y_2 gives (using the expression for "perfect" partial control in (10.28))*

$$y_1 = \left(g^*_{11} - \frac{g^*_{12} g^*_{21}}{g^*_{22}} \right) u^*_1 = -\frac{1}{5} u^*_1$$

The controller for the pairing u^*_1-y_1 becomes $k^*_1(s) = 1/(g^*_{11}\tau_1 s) = -5/(\tau_1 s)$ and thus

$$K^* = \begin{bmatrix} \frac{-5}{\tau_1 s} & 0 \\ 0 & \frac{1}{5\tau_2 s} \end{bmatrix} \tag{10.60}$$

The response with $\tau_1 = 5$ and $\tau_2 = 1$ is shown in Figure 10.16(b). We see that performance is only slightly worse than with the diagonal pairings. However, more seriously, we have the problem that if control of y_2 fails, e.g. because $u^*_2 = u_1$ saturates, then we also lose control of y_1 (in addition, we get instability with y_2 drifting away, because of the integral action for y_1). The situation is particularly bad in this case because of the pairing on a zero element, but the dependence on faster (inner) loops being in service is a general problem with sequential design.

Exercise 10.13 . *Redo the simulations in Example 10.15 with 20% diagonal input uncertainty. Speci£cally, add a block* $\begin{bmatrix} 1.2 & 0 \\ 0 & 0.8 \end{bmatrix}$ *between the plant and the controller. Also simulate with the decoupler* $K(s) = \frac{1}{s}\begin{bmatrix} 1 & 0 \\ -5 & 1 \end{bmatrix}$ *which is expected to be particularly sensitive to uncertainty (why? – see conclusions on page 251 and note that* $\gamma_I^*(G) = 10$ *for this plant).*

Example 10.16 **Two-way interactive plant.** *Consider the plant*

$$G = \begin{bmatrix} 1 & g_{12} \\ 5 & 1 \end{bmatrix} \tag{10.61}$$

for which

$$G^{-1} = \frac{1}{1-5g_{12}} \begin{bmatrix} 1 & -g_{12} \\ -5 & 1 \end{bmatrix} \quad \text{and} \quad \text{RGA} = \frac{1}{1-5g_{12}} \begin{bmatrix} 1 & -5g_{12} \\ -5g_{12} & 1 \end{bmatrix}$$

The control properties of this plant depend on the parameter g_{12}*. The plant is singular* $(\det(G) = 1 - 5g_{12} = 0)$ *for* $g_{12} = 0.2$*, and in this case independent control of both outputs is impossible, whatever the controller. We will examine the diagonal pairings using the independent design controller*

$$K = \begin{bmatrix} \frac{1}{\tau_1 s} & 0 \\ 0 & \frac{1}{\tau_2 s} \end{bmatrix} \tag{10.62}$$

The individual loops are stable with responses $y_1 = \frac{1}{(\tau_1 s+1)} r_1$ *and* $y_2 = \frac{1}{(\tau_2 s+1)} r_2$*, respectively. With both loops closed, the response is* $y = GK(I + GK)^{-1} r = Tr$*, where*

$$T = \frac{1}{(\tau_1 s+1)(\tau_2 s+1) - 5g_{12}} \begin{bmatrix} \tau_2 s + 1 - 5g_{12} & g_{12}\tau_1 s \\ 5\tau_2 s & \tau_1 s + 1 - 5g_{12} \end{bmatrix}$$

We see that $T(0) = I$*, so we have perfect steady-state control, as is expected with integral action. However, the interactions as expressed by the term* $5g_{12}$ *may yield instability, and we £nd that the system is closed-loop unstable for* $g_{12} > 0.2$*. This is also expected because the diagonal RGA elements are negative for* $g_{12} > 0.2$*, indicating a gain change between the open-loop* (g_{ii}) *and closed-loop* ($\widehat{g}_{ii}$) *transfer functions, which is incompatible with integral action. Thus, for* $g_{12} > 0.2$*, the off-diagonal pairings must be used if we want to use an independent design (with stable individual loops).*

We will now consider three cases, (a) $g_{12} = 0.17$*,* (b) $g_{12} = -0.2$ *and* (c) $g_{12} = -1$*, each with the same controller (10.62) with* $\tau_1 = 5$ *and* $\tau_2 = 1$*. Because of the large interactions given by* $g_{21} = 5$*, we need to control* y_2 *faster than* y_1*.*

(a) $\underline{g_{12} = 0.17}$*. In this case,*

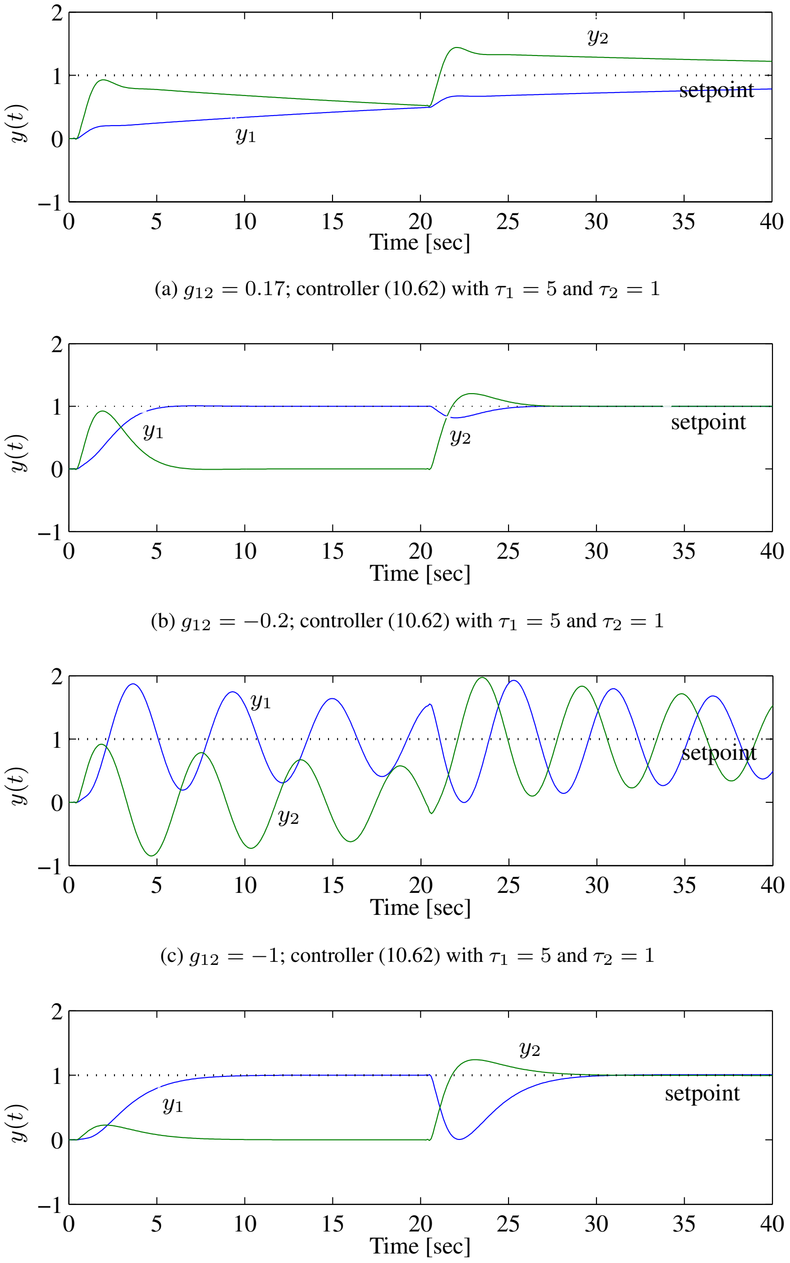
$$G^{-1} = \begin{bmatrix} 6.7 & -1.1 \\ -33.3 & 6.7 \end{bmatrix} \quad \text{and} \quad \text{RGA} = \begin{bmatrix} 6.7 & -5.7 \\ -5.7 & 6.7 \end{bmatrix}$$

The large RGA elements indicate strong interactions. Furthermore, recall from (3.56) that the RGA gives the ratio of the open-loop and (partially) closed-loop gains, $g_{ij}/\widehat{g}_{ij}$*. Thus, in terms of decentralized control, the large positive RGA elements indicate that* $\widehat{g}_{ij}$ *is small and the loops will tend to counteract each other by reducing the effective loop gain. This is con£rmed by simulations in Figure 10.17(a).*

(b) $\underline{g_{12} = -0.2}$*. In this case,*

$$G^{-1} = \begin{bmatrix} 0.5 & 0.1 \\ -2.5 & 0.5 \end{bmatrix} \quad \text{and} \quad \text{RGA} = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}$$

The RGA elements of 0.5 *indicate quite strong interactions and show that the interaction increases the effective gain. This is con£rmed by the closed-loop responses in Figure 10.17(b).*



(a) $g_{12} = 0.17$; controller (10.62) with $\tau_1 = 5$ and $\tau_2 = 1$

(b) $g_{12} = -0.2$; controller (10.62) with $\tau_1 = 5$ and $\tau_2 = 1$

(c) $g_{12} = -1$; controller (10.62) with $\tau_1 = 5$ and $\tau_2 = 1$

(d) $g_{12} = -1$; controller (10.62) with $\tau_1 = 21.95$ and $\tau_2 = 1$

Figure 10.17: Decentralized control of plant (10.61) with diagonal pairings

(c) $\underline{g_{12} = -1}$. *In this case,*

$$G^{-1} = \begin{bmatrix} 0.17 & 0.17 \\ -0.83 & 0.17 \end{bmatrix} \quad \text{and} \quad \text{RGA} = \begin{bmatrix} 0.17 & 0.83 \\ 0.83 & 0.17 \end{bmatrix}$$

The RGA indicates clearly that the off-diagonal pairings are preferable. Nevertheless, we will consider the diagonal pairings with $\tau_1 = 5$ *and* $\tau_2 = 1$ *(as before). The response is poor as seen in Figure 10.17(c). The closed-loop system is stable, but very oscillatory. This is not surprising as the diagonal RGA elements of* 0.17 *indicate that the interactions increase the effective loop gains by a factor* 6 $(= 1/0.17)$. *To study this in more detail, we write the closed-loop polynomial in standard form*

$$(\tau_1 s + 1)(\tau_2 s + 1) - 5g_{12} = \tau^2 s^2 + 2\tau\zeta s + 1$$

with

$$\tau = \sqrt{\frac{\tau_1\tau_2}{1 - 5g_{12}}} \quad \text{and} \quad \zeta = \frac{1}{2}\frac{\tau_1 + \tau_2}{\sqrt{\tau_1\tau_2}}\frac{1}{\sqrt{1 - 5g_{12}}}$$

We note that we get oscillations $(0 < \zeta < 1)$, *when* g_{12} *is negative and large. For example,* $g_{12} = -1$, $\tau_1 = 5$ *and* $\tau_2 = 1$ *gives* $\zeta = 0.55$. *Interestingly, we see from the expression for* ζ *that the oscillations may be reduced by selecting* τ_1 *and* τ_2 *to be more different. This follows because* $\frac{1}{2}\frac{\tau_1+\tau_2}{\sqrt{\tau_1\tau_2}}$ *is the ratio between the arithmetic and geometric means, which is larger the more different* τ_1 *and* τ_2 *are. Indeed, with* $g_{12} = -1$ *we find that oscillations can be eliminated* $(\zeta = 1)$ *by selecting* $\tau_1 = 21.95\tau_2$. *This is confirmed by the simulations in Figure10.17(d). The response is surprisingly good taking into account that we are using the wrong pairings.*

Exercise 10.14 *Design decentralized controllers for the* 3×3 *plant* $G(s) = G(0)e^{-0.5s}$ *where* $G(0)$ *is given by (10.80). Try both the diagonal pairings and the pairings corresponding to positive steady-state RGA elements, i.e.* $G^* = G\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}^T$.

The above examples show that in many cases we can achieve quite good performance with decentralized control, even for interactive plants. However, decentralized controller design is more difficult for such plants, and this, in addition to the possibility for improved performance, favours the use of multivariable control for interactive plants.

With the exception of Section 10.6.6, the focus in the rest of this chapter is on *independently designed* decentralized control systems, which cannot be analyzed using the expressions for partial control presented earlier in (10.28). We present tools for pairing selections (step 1) and for analyzing the stability and performance of decentralized control systems based on independent design. Readers who are primarily interested in applications of decentralized control may want to go directly to the summary in Section 10.6.8 (page 449).

10.6.3 Notation and factorization of sensitivity function

$G(s)$ denotes a square $m \times m$ plant with elements g_{ij}. With a particular choice of pairings we can rearrange the columns or rows of $G(s)$ such that the paired elements are along the diagonal of $G(s)$. We then have that the controller $K(s)$ is diagonal $(\text{diag}\{k_i\})$. We introduce

$$\widetilde{G} \triangleq \text{diag}\{g_{ii}\} = \begin{bmatrix} g_{11} & & & \\ & g_{22} & & \\ & & \ddots & \\ & & & g_{mm} \end{bmatrix} \tag{10.63}$$

as the matrix consisting of the diagonal elements of G. The loop transfer function in loop i is denoted $L_i = g_{ii}k_i$, which is also equal to the i'th diagonal element of $L = GK$.

$$\widetilde{S} \triangleq (I + \widetilde{G}K)^{-1} = \mathrm{diag}\left\{\frac{1}{1 + g_{ii}k_i}\right\} \quad \text{and} \quad \widetilde{T} = I - \widetilde{S} \tag{10.64}$$

contain the sensitivity and complementary sensitivity functions for the individual loops. Note that $\widetilde{S}$ is *not* equal to the matrix of diagonal elements of $S = (I + GK)^{-1}$.

With decentralized control, the interactions are given by the off-diagonal elements $G - \widetilde{G}$. The interactions can be normalized with respect to the diagonal elements and we define

$$E \triangleq (G - \widetilde{G})\widetilde{G}^{-1} \tag{10.65}$$

The "magnitude" of the matrix E is commonly used as an "interaction measure". We will show that $\mu(E)$ (where μ is the structured singular value) is the best (least conservative) measure, and will define "generalized diagonal dominance" to mean $\mu(E) < 1$. To derive these results we make use of the following important factorization of the "overall" sensitivity function $S = (I + GK)^{-1}$ with all loops closed,

$$\boxed{\underbrace{S}_{\text{overall}} = \underbrace{\widetilde{S}}_{\text{individual loops}} \underbrace{(I + E\widetilde{T})^{-1}}_{\text{interactions}}} \tag{10.66}$$

Equation (10.66) follows from (A.147) with $G = \widetilde{G}$ and $G' = G$. The reader is encouraged to confirm that (10.66) is correct, because most of the important results for stability and performance using independent design may be derived from this expression.

A related factorization which follows from (A.148) is

$$S = \widetilde{S}(I - E_S\widetilde{S})^{-1}(I - E_S) \tag{10.67}$$

where

$$E_S = (G - \widetilde{G})G^{-1} \tag{10.68}$$

(10.67) may be rewritten as

$$S = (I + \widetilde{S}(\Gamma - I))^{-1}\widetilde{S}\Gamma \tag{10.69}$$

where Γ is the performance relative gain array (PRGA),

$$\Gamma(s) \triangleq \widetilde{G}(s)G^{-1}(s) \tag{10.70}$$

Γ is a normalized inverse of the plant. Note that $E_S = I - \Gamma$ and $E = \Gamma^{-1} - I$. In Section 10.6.7 we discuss in more detail the use of the PRGA.

These factorizations are particularly useful for analyzing decentralized control systems based on *independent design*, because the basis is then the individual loops with transfer function $\widetilde{S}$.

10.6.4 Stability of decentralized control systems

We consider the independent design procedure and assume that (a) the plant G is stable and (b) each individual loop is stable by itself ($\widetilde{S}$ and $\widetilde{T}$ are stable). Assumption (b) is the basis for independent design. Assumption (a) is also required for independent design because we want to be able to take any loop(s) out of service and remain stable, and this is not possible if the plant is unstable.

To achieve stability of the overall system with all loops closed, we must require that the interactions do not cause instability. We use the expressions for S in (10.66) and (10.69) to derive conditions for this.

Theorem 10.3 *With assumptions (a) and (b), the overall system is stable (S is stable):*
(i) if and only if $(I + E\widetilde{T})^{-1}$ is stable, where $E = (G - \widetilde{G})\widetilde{G}^{-1}$,
(ii) if and only if $\det(I + E\widetilde{T}(s))$ does not encircle the origin as s traverses the Nyquist D-contour,
(iii) if

$$\rho(E\widetilde{T}(j\omega)) < 1, \forall\omega \tag{10.71}$$

(iv) (and (10.71) is satisfied) if

$$\bar{\sigma}(\widetilde{T}) = \max_i |\widetilde{t}_i| < 1/\mu(E) \quad \forall\omega \tag{10.72}$$

The structured singular value $\mu(E)$ is computed with respect to a diagonal structure (of $\widetilde{T}$).

Proof: (Grosdidier and Morari, 1986) (ii) follows from the factorization $S = \widetilde{S}(I + E\widetilde{T})^{-1}$ in (10.66) and the generalized Nyquist theorem in Lemma A.5 (page 543). (iii) Condition (10.71) follows from the spectral radius stability condition in (4.110). (iv) The least conservative way to split up $\rho(E\widetilde{T})$ is to use the structured singular value. From (8.92) we have $\rho(E\widetilde{T}) \leq \mu(E)\bar{\sigma}(T)$ and (10.72) follows. □

Theorem 10.4 *With assumptions (a) and (b) and also assuming that that G and $\widetilde{G}$ have no RHP-zeros, the overall system is stable (S is stable):*
(i) if and only if $(I - E_S\widetilde{S}(s))^{-1}$ is stable, where $E_S = (G - \widetilde{G})G^{-1}$,
(ii) if and only if $\det(I - E_S\widetilde{S})$ does not encircle the origin as s traverses the Nyquist D-contour,
(iii) if

$$\rho(E_S\widetilde{S}(j\omega)) < 1, \forall\omega \tag{10.73}$$

(iv) (and (10.73) is satisfied) if

$$\bar{\sigma}(\widetilde{S}) = \max_i |\widetilde{s}_i| < 1/\mu(E_S) \quad \forall\omega \tag{10.74}$$

The structured singular value $\mu(E_S)$ is computed with respect to a diagonal structure (of $\widetilde{S}$).

Proof: The proof is similar to that of Theorem 10.3. We need to assume no RHP-zeros in order to get (i). □

Remark. The μ-conditions (10.72) and (10.74) for (nominal) stability of the decentralized control system can be generalized to include robust stability and robust performance; see equations (31a-b) in Skogestad and Morari (1989).

In both the above Theorems, (i) and (ii) are necessary and sufficient conditions for stability, whereas the spectral radius condition (iii) is weaker (only sufficient) and the μ-condition condition (iv) is even weaker. Nevertheless, the use of μ is the least conservative way of "splitting up" the spectral radius ρ in condition (iii).

Equation (10.72) is easy to satisfy at high frequencies, where generally $\bar{\sigma}(\widetilde{T}) \rightarrow 0$. Similarly, (10.74) is usually easy to satisfy at low frequencies since $\bar{\sigma}(\widetilde{S}(0)) = 0$ for systems with integral control (no steady-state offset). Unfortunately, the two conditions cannot be combined over different frequency ranges (Skogestad and Morari, 1989). Thus, to guarantee stability we need to satisfy one of the conditions over the whole frequency range.

Since (10.72) is generally most dif£cult to satisfy at low frequencies, where usually $\bar{\sigma}(\widetilde{T}) \approx 1$, this gives rise to the following pairing rule:

- *Prefer pairings with* $\mu(E) < 1$ *("diagonal dominance") at frequencies within the closed-loop bandwidth.*

Let Λ denote the RGA of G. For an $n \times n$ plant $\lambda_{ii}(0) > 0.5 \, \forall i$ is a necessary condition for $\mu(E(0)) < 1$ (diagonal dominance at steady state) (Kariwala et al., 2003). This gives the following pairing rule: *Prefer pairing on steady-state RGA elements larger than* 0.5 *(because otherwise we can never have* $\mu(E(0)) < 1$*).*

Since (10.74) is generally most dif£cult to satisfy at high frequencies where $\bar{\sigma}(\widetilde{S}) \approx 1$, and since encirclement of the origin of $\det(I - E_S\widetilde{S}(s))$ is most likely to occur at frequencies up to crossover, this gives rise to the following pairing rule:

- *Prefer pairings with* $\mu(E_S) < 1$ *("diagonal dominance") at crossover frequencies.*

Gershgorin bounds. An alternative to splitting up $\rho(E\widetilde{T})$ using μ, is to use Gershgorin's theorem, see page 519. From (10.71) we may then derive (Rosenbrock, 1974) suf£cient conditions for overall stability, either in terms of the rows of G,

$$|\widetilde{t}_i| < |g_{ii}| / \sum_{j \neq i} |g_{ij}| \quad \forall i, \forall \omega \tag{10.75}$$

or, alternatively, in terms of the columns,

$$|\widetilde{t}_i| < |g_{ii}| / \sum_{j \neq i} |g_{ji}| \quad \forall i, \forall \omega \tag{10.76}$$

This gives the important insight that it is preferable to pair on large elements in G, because then the sum of the off-diagonal elements, $\sum_{j \neq i} |g_{ij}|$ and $\sum_{j \neq i} |g_{ji}|$, is small. The "Gershgorin bounds", which should be small, are the inverse of the right hand sides in (10.75) and (10.76),

The Gershgorin conditions (10.75) and (10.76), are complementary to the μ-condition in (10.72). Thus, the use of (10.72) is not always better (less conservative) than (10.75) and (10.76). It is true that the *smallest* of the $i = 1, \dots m$ upper bounds in (10.75) or (10.76) is always smaller (more restrictive) than $1/\mu(E)$ in (10.72). However, (10.72) imposes the *same* bound on $|\widetilde{t}_i|$ for each loop, whereas (10.75) and (10.76) give *individual* bounds, some of which may be less restrictive than $1/\mu(E)$.

Diagonal dominance. Although "diagonal dominance" is a matrix property, its de£nition has been motivated by control, where, loosely speaking, diagonal dominance means that the interactions will not introduce instability. Originally, for example in the Inverse Nyquist Array method of Rosenbrock (1974), diagonal dominance was de£ned in terms of the Gershgorin bounds, resulting in the conditions $\|E\|_{i1} < 1$ ("column dominance") and $\|E\|_{i\infty} < 1$ ("row dominance"), where $E = (G - \widetilde{G})\widetilde{G}^{-1}$. However, stability is scaling independent, and by "optimally" scaling the plant using DGD^{-1}, where the scaling matrix D is diagonal, one obtains from these conditions that the matrix G is (generalized) diagonally dominant if $\rho(|E|) < 1$; see (A.128). Here $\rho(|E|)$ is the Perron root of E. An even less restrictive definition of diagonal dominance is obtained by starting from the stability condition in terms of $\mu(E)$ in (10.72). This leads us to propose the improved definition below.

Definition 10.1 A matrix G is generalized diagonally dominant if and only if $\mu(E) < 1$.

Here the term "generalized diagonally dominant" means "can be scaled to be diagonally dominant". Note that we always have $\mu(E) \leq \rho(|E|)$, so the use of μ is less restrictive than the Perron root. Also note that $\mu(E) = 0$ for a triangular plant.[5] It is also possible to use $\mu(E_s)$ as measure of diagonal dominance, and we then have that a matrix is generalized diagonally dominant if $\mu(E) < 1$ or if $\mu(E_S) < 1$.

Example 10.17 *Consider the following plant where we pair on its diagonal elements:*

$$G = \begin{bmatrix} -5 & 1 & 2 \\ 4 & 2 & -1 \\ -3 & -2 & 6 \end{bmatrix}; \quad \widetilde{G} = \begin{bmatrix} -5 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 6 \end{bmatrix}; \quad E = (G - \widetilde{G})\widetilde{G}^{-1} = \begin{bmatrix} 0 & 0.5 & 0.33 \\ -0.8 & 0 & -0.167 \\ 0.6 & -1 & 0 \end{bmatrix}$$

The μ-interaction measure is $\mu(E) = 0.9189$, so the plant is diagonally dominant. From (10.72), stability of the individual loops $\widetilde{t}_i$ guarantees stability of the overall closed-loop system, provided we keep the individual peaks of $|\widetilde{t}_i|$ less than $1/\mu(E) = 1.08$. This allows for integral control with $\widetilde{t}(0) = 1$. Note that it is not possible in this case to conclude from the Gershgorin bounds in (10.75) and (10.76) that the plant is diagonally dominant, because the $2,2$ element of G $(= 2)$ is smaller than both the sum of the off-diagonal elements in row 2 $(= 5)$ and in column 2 $(= 3)$.

Iterative RGA. An iterative computation of the RGA, $\Lambda^k(G)$, gives a permuted identity matrix that corresponds to the (permuted) generalized diagonal dominant pairing, if it exists (Johnson and Shapiro, 1986, Theorem 2) (see also page 88). Note that the iterative RGA avoids the combinatorial problem of testing all pairings, as is required when computing $\mu(E)$ or the RGA number. Thus, we may use the iterative RGA to find a promising pairing, and check for diagonal dominance using $\mu(E)$.

Exercise 10.15 *For the plant in Example 10.17 check that the iterative RGA converges to the diagonally dominant pairings.*

Example 10.18 **RGA number.** *The RGA number, $\|\Lambda - I\|_{\text{sum}}$, is commonly used as a measure of diagonal dominance, but unfortunately for 4×4 plants or larger, a small RGA number does not guarantee diagonal dominance. To illustrate this, consider the matrix* `G = [1 1 0 0; 0 0.1 1 1; 1 1 0.1 0; 0 0 1 1]`. *It has has RGA$= I$, but $\mu(E) = \mu(E_S) = 10.9$ so it is far from diagonally dominant.*

Triangular plants. Overall stability is trivially satisfied for a triangular plant as described in the theorem below.

Theorem 10.5 *Suppose the plant $G(s)$ is stable and upper or lower triangular (at all frequencies), and is controlled by a diagonal controller. Then the overall system is stable if and only if the individual loops are stable.*

[5] A triangular plant may have large off-diagonal elements, but it can be scaled to be diagonal. For example $\begin{bmatrix} d_1 & 0 \\ 0 & d_2 \end{bmatrix}\begin{bmatrix} g_{11} & 0 \\ g_{21} & g_{22} \end{bmatrix}\begin{bmatrix} 1/d_1 & 0 \\ 0 & 1/d_2 \end{bmatrix} = \begin{bmatrix} g_{11} & 0 \\ \frac{d_2}{d_1}g_{12} & g_{22} \end{bmatrix}$ which approaches $\begin{bmatrix} g_{11} & 0 \\ 0 & g_{22} \end{bmatrix}$ for $|d_1| \gg |d_2|$.

Proof: For a triangular plant G, $E = (G - \widetilde{G})\widetilde{G}^{-1}$ is triangular with all diagonal elements zero, so it follows that all eigenvalues of $E\widetilde{T}$ are zero. Thus $\det(I + E\widetilde{T}(s)) = 1$ and from (ii) in Theorem 10.3 the interactions can not cause instability. □

Because of interactions, there may not exists pairings such that the plant is triangular at low frequencies. Fortunately, in practice it is sufficient for stability that the plant is triangular at crossover frequencies, and we have:

> **Triangular pairing rule.** *To achieve stability with decentralized control, prefer pairings such that at frequencies ω around crossover, the rearranged plant matrix $G(j\omega)$ (with the paired elements along the diagonal) is close to triangular.*

Derivation of triangular pairing rule. The derivation is based on Theorem 10.4. From the spectral radius stability condition in (10.74) the overall system is stable if $\rho(\widetilde{S}E_S(j\omega)) < 1, \quad \forall\omega$. At low frequencies, this condition is usually satisfied because $\widetilde{S}$ is small. At higher frequencies, where $\widetilde{S} = \mathrm{diag}\{\widetilde{s}_i\} \approx I$, (10.74) may be satisfied if $G(j\omega)$ is close to triangular. This is because E_S and thus $\widetilde{S}E_S$ are then close to triangular, with diagonal elements close to zero, so the eigenvalues of $\widetilde{S}E_S(j\omega)$ are close to zero. Thus (10.74) is satisfied and we have stability of S. The use of Theorem 10.4 assumes that G and $\widetilde{G}$ have no RHP-zeros, but in practice the result also holds for plants with RHP-zeros provided they are located beyond the crossover frequency range. □

Remark. Triangular plant, RGA$=I$ and stability. An important RGA-property is that the RGA of a triangular plant is always the identity matrix ($\Lambda = I$) or equivalently the RGA number is zero; see property 4 on page 527. In the first edition of this book (Skogestad and Postlethwaite, 1996), we incorrectly claimed that the reverse is also true; that is, an identity RGA matrix ($\Lambda(G) = I$) implies that G is triangular. Then, in the first printing of the second edition we incorrectly claimed that it holds for 3×3 systems or smaller, but actually it holds only for 2×2 systems or smaller as illustrated by the following 3×3 counterexample (due to Vinay Kariwala):

$$G = \begin{bmatrix} g_{11} & 0 & 0 \\ g_{21} & g_{22} & g_{23} \\ g_{31} & 0 & g_{33} \end{bmatrix} \tag{10.77}$$

has RGA$=I$ in all cases (for any nonzero value of the indicated entries g_{ij}), but G is not triangular. On the other hand, note that this G is diagonally dominant since $\mu(E) = 0$ in all cases. However, more generally RGA$=I$ does not imply diagonal dominance as illustrated by the following 4×4 matrix [6]

$$G = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & \alpha & 1 & 1 \\ 1 & 1 & \beta & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \tag{10.78}$$

which has RGA$=I$ for any nonzero value of α and β, but G is not triangular and not always diagonal dominant. For example, $\mu(E) = 3.26$ (not diagonally dominant) for $\alpha = \beta = 0.4$. Also, for this plant stability of the individual loops does not necessarily give overall stability. For example, $\widetilde{T} = \frac{1}{\tau s+1} I$ (stable individual loops) gives instability (T unstable) with $\alpha = \beta$ when $|\alpha| = |\beta| < 0.4$. Therefore, RGA$=I$ and stable individual loops do *not* generally guarantee overall stability (it is *not* a sufficient stability condition). Nevertheless, it is clear that we would *prefer* to have RGA$=I$, because otherwise the plant cannot be triangular. Thus, from the triangular pairing rule we have that it is desirable to select pairings such that the RGA is close to the identity matrix in the crossover region.

[6] (10.78) is a generalization of a counterexample given by Johnson and Shapiro (1986). On our book's home page a physical mixing process is given with a transfer function of this form.

10.6.5 Integrity and negative RGA elements

A desirable property of a decentralized control system is that it has *integrity*, that is, the closed-loop system should remain stable as subsystem controllers are brought in and out of service or when inputs saturate. Mathematically, the system possesses integrity if it remains stable when the controller K is replaced by $\mathbb{E}K$ where $\mathbb{E} = \text{diag}\{\epsilon_i\}$ and ϵ_i may take on the values of $\epsilon_i = 0$ or $\epsilon_i = 1$.

An even stronger requirement ("complete detunability") is when it is required that the system remains stable as the gain in various loops is reduced (detuned) by an arbitrary factor, i.e. ϵ_i may take any value between 0 and 1, $0 \leq \epsilon_i \leq 1$. *Decentralized integral controllability* (DIC) is concerned with whether complete detunability is *possible* with *integral control*.

De£nition 10.2 Decentralized integral controllability (DIC). *The plant $G(s)$ (corresponding to a given pairing with the paired elements along its diagonal) is DIC if there exists a stabilizing decentralized controller with integral action in each loop such that each individual loop may be detuned independently by a factor ϵ_i ($0 \leq \epsilon_i \leq 1$) without introducing instability.*

Note that DIC considers the *existence* of a controller, so it depends only on the plant G and the chosen pairings. The steady-state RGA provides a very useful tool to test for DIC, as is clear from the following result which was £rst proved by Grosdidier et al. (1985).

Theorem 10.6 Steady-state RGA and DIC. *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. 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:*
(a) The overall closed-loop system is unstable.
(b) The loop with the negative relative gain is unstable by itself.
(c) The closed-loop system is unstable if the loop with the negative relative gain is opened (broken).

This can be summarized as follows:

$$\text{A stable (reordered) plant } G(s) \text{ is DIC only if } \lambda_{ii}(0) \geq 0 \text{ for all } i. \tag{10.79}$$

Proof: Use Theorem 6.7 on page 252 and select $G' = \text{diag}\{g_{ii}, G^{ii}\}$. Since $\det G' = g_{ii} \det G^{ii}$ and from (A.78) $\lambda_{ii} = \frac{g_{ii} \det G^{ii}}{\det G}$ we have $\det G' / \det G = \lambda_{ii}$ and Theorem 10.6 follows. □

Each of the three possible instabilities in Theorem 10.6 resulting from pairing on a negative value of $\lambda_{ij}(0)$ is undesirable. The worst case is (a) when the overall system is unstable, but situation (c) is also highly undesirable as it will imply instability if the loop with the negative relative gain somehow becomes inactive, e.g. due to input saturation. Situation (b) is unacceptable if the loop in question is intended to be operated by itself, or if all the other loops may become inactive, e.g. due to input saturation.

The RGA is a very ef£cient tool because it does not have to be recomputed for each possible choice of pairing. This follows since any permutation of the rows and columns of G results in the same permutation in the RGA of G. To achieve DIC one has to pair on a positive RGA(0) element in each row and column, and therefore one can often eliminate many candidate pairings by a simple glance at the RGA matrix. This is illustrated by the following examples:

Example 10.19 *Consider a* 3×3 *plant with*

$$G(0) = \begin{bmatrix} 10.2 & 5.6 & 1.4 \\ 15.5 & -8.4 & -0.7 \\ 18.1 & 0.4 & 1.8 \end{bmatrix} \quad \text{and} \quad \Lambda(0) = \begin{bmatrix} 0.96 & \mathbf{1.45} & -1.41 \\ \mathbf{0.94} & -0.37 & 0.43 \\ -0.90 & -0.07 & \mathbf{1.98} \end{bmatrix} \tag{10.80}$$

For a 3×3 *plant there are six possible pairings, but from the steady-state RGA we see that there is only one positive element in column* 2 *(*$\lambda_{12} = 1.45$*), and only one positive element in row* 3 *(*$\lambda_{33} = 1.98$*), and therefore there is only one possible pairing with all RGA elements positive (*$u_1 \leftrightarrow y_2$*,* $u_2 \leftrightarrow y_1$*,* $u_3 \leftrightarrow y_3$*). Thus, if we require to pair on the positive RGA elements, we can from a quick glance at the steady-state RGA eliminate five of the six pairings.*

Example 10.20 *Consider the following plant and RGA:*

$$G(0) = \begin{bmatrix} 0.5 & 0.5 & -0.004 \\ 1 & 2 & -0.01 \\ -30 & -250 & 1 \end{bmatrix} \quad \text{and} \quad \Lambda(0) = \begin{bmatrix} -1.56 & -2.19 & 4.75 \\ 3.12 & 4.75 & -6.88 \\ -0.56 & -1.56 & 3.12 \end{bmatrix} \tag{10.81}$$

From the RGA, we see that it is impossible to rearrange the plant such that all diagonal RGA elements are positive. Consequently, this plant is not DIC for any choice of pairings.

Example 10.21 *Consider the following plant and RGA:*

$$G(s) = \frac{(-s+1)}{(5s+1)^2} \begin{bmatrix} 1 & -4.19 & -25.96 \\ 6.19 & 1 & -25.96 \\ 1 & 1 & 1 \end{bmatrix} \quad \text{and} \quad \Lambda(G) = \begin{bmatrix} 1 & 5 & -5 \\ -5 & 1 & 5 \\ 5 & -5 & 1 \end{bmatrix}$$

Note that the RGA is constant, independent of frequency. Only two of the six possible pairings give positive steady-state RGA elements (see pairing rule 2 on page 450): (a) the (diagonal) pairing on all $\lambda_{ii} = 1$ *and (b) the pairing on all* $\lambda_{ii} = 5$*. Intuitively, one may expect pairing (a) to be the best since it corresponds to pairing on RGA elements equal to* 1*. However, the RGA matrix is far from identity, and the RGA number,* $\|\Lambda - I\|_{\text{sum}}$*, is* 30 *for both pairings. Also, none of the pairings are diagonally dominant as* $\mu(E) = 8.84$ *for pairing (a) and* $\mu(E) = 1.25$ *for the pairing (b). These are larger than* 1*, so none of the two alternatives satisfy pairing rule 1 discussed on page 450, and we are led to conclude that decentralized control should not be used for this plant.*

Hovd and Skogestad (1992) confirm this conclusion by designing PI controllers for the two cases. They found pairing (a) corresponding to $\lambda_{ii} = 1$ *to be significantly worse than (b) with* $\lambda_{ii} = 5$*, in agreement with the values for* $\mu(E)$*. They also found the achievable closed-loop time constants to be* 1160 *and* 220*, respectively, which in both cases is very slow compared to the RHP-zero which has a time constant of 1.*

Exercise 10.16 *Use the method of "iterative RGA" (page 88) on the model in Example 10.21, and confirm that it results in "recommending" the pairing on* $\lambda_{ii} = 5$*, which indeed was found to be the best choice based on* $\mu(E)$ *and the simulations. (This is partly good luck, because the proven theoretical result for iterative RGA only holds for a generalized diagonally dominant matrix.)*

Exercise 10.17* *(a) Assume that the* 4×4 *matrix in (A.83) represents the steady-state model of a plant. Show that* 20 *of the* 24 *possible pairings can be eliminated by requiring DIC. (b) Consider the* 3×3 *FCC process in Exercise 6.17 on page 257. Show that five of the six possible pairings can be eliminated by requiring DIC.*

Remarks on DIC and RGA.

1. DIC was introduced by Skogestad and Morari (1988b) who also give necessary and sufficient conditions for testing DIC. A detailed survey of conditions for DIC and other related properties is given by Campo and Morari (1994).

2. DIC is also closely related to D-stability, see papers by Yu and Fan (1990) and Campo and Morari (1994). The theory of D-stability provides necessary and sufficient conditions (except in a few special cases, such as when the determinant of one or more of the submatrices is zero).
3. Unstable plants are not DIC. The reason for this is that with all $\epsilon_i = 0$ we are left with the uncontrolled plant G, and the system will be (internally) unstable if $G(s)$ is unstable.
4. For $\epsilon_i = 0$ we assume that the integrator of the corresponding SISO controller has been removed, otherwise the integrator would yield internal instability.
5. For 2×2 and 3×3 plants we have even tighter RGA conditions for DIC than (10.79). For 2×2 plants (Skogestad and Morari, 1988b)

$$\text{DIC} \quad \Leftrightarrow \quad \lambda_{11}(0) > 0 \tag{10.82}$$

For 3×3 plants with positive diagonal RGA elements of $G(0)$ and of $G^{ii}(0), i = 1, 2, 3$ (its three principal submatrices), we have (Yu and Fan, 1990)

$$\text{DIC} \quad \Leftrightarrow \quad \sqrt{\lambda_{11}(0)} + \sqrt{\lambda_{22}(0)} + \sqrt{\lambda_{33}(0)} \geq 1 \tag{10.83}$$

(Strictly speaking, as pointed out by Campo and Morari (1994), we do not have equivalence for the case when $\sqrt{\lambda_{11}(0)} + \sqrt{\lambda_{22}(0)} + \sqrt{\lambda_{33}(0)}$ is identically equal to 1, but this has little practical significance.)
6. One cannot in general expect tight conditions for DIC in terms of the RGA (i.e. for 4×4 systems or higher). The reason for this is that the RGA essentially only considers "corner values", $\epsilon_i = 0$ or $\epsilon_i = 1$, for the detuning factor, that is, it tests for integrity. This is clear from the fact that $\lambda_{ii} = \frac{g_{ii} \det G^{ii}}{\det G}$, where G corresponds to $\epsilon_i = 1$ for all i, g_{ii} corresponds to $\epsilon_i = 1$ with the other $\epsilon_k = 0$, and G^{ii} corresponds to $\epsilon_i = 0$ with the other $\epsilon_k = 1$. A more complete integrity ("corner-value") result is given next.
7. **Determinant condition for integrity (DIC).** The following condition is concerned with whether it is possible to design a decentralized controller for the plant such that the system possesses *integrity*, which is a prerequisite for having DIC. *Assume without loss of generality that the signs of the rows or columns of G have been adjusted such that all diagonal elements of $G(0)$ are positive, i.e. $g_{ii}(0) \geq 0$. Then one may compute the determinant of $G(0)$ and all its principal submatrices (obtained by deleting rows and corresponding columns in $G(0)$), which should all have the same sign for integrity.* This determinant condition follows by applying Theorem 6.7 to all possible combinations of $\epsilon_i = 0$ or 1 as illustrated in the proof of Theorem 10.6.
8. The Niederlinski index of a matrix G is defined as

$$N_I(G) = \det G / \Pi_i g_{ii} \tag{10.84}$$

A simple way to test the determinant condition for integrity, which is a necessary condition for DIC, is to require that the Niederlinski index of $G(0)$ and the Niederlinski indices of all the principal submatrices $G^{ii}(0)$ of $G(0)$ are positive.

The original result of Niederlinski, which involved only testing N_I of $G(0)$, obviously yields less information than the determinant condition as does the use of the sign of the RGA elements. This is because the RGA element is $\lambda_{ii} = \frac{g_{ii} \det G^{ii}}{\det G}$, so we may have cases where two negative determinants result in a positive RGA element. Nevertheless, the RGA is usually the preferred tool because it does not have to be recomputed for each pairing. Let us first consider an example where the Niederlinski index is inconclusive:

$$G_1(0) = \begin{bmatrix} 10 & 0 & 20 \\ 0.2 & 1 & -1 \\ 11 & 12 & 10 \end{bmatrix} \quad \text{and} \quad \Lambda(G_1(0)) = \begin{bmatrix} 4.58 & 0 & -3.58 \\ 1 & -2.5 & 2.5 \\ -4.58 & 3.5 & 2.08 \end{bmatrix}$$

Since one of the diagonal RGA elements is negative, we conclude that this pairing is *not* DIC. On the other hand, $N_I(G_1(0)) = 0.48$ (which is positive), so Niederlinski's original condition is inconclusive. However, the N_I of the three principal submatrices $\begin{bmatrix} 10 & 0 \\ 0.2 & 1 \end{bmatrix}$, $\begin{bmatrix} 10 & 20 \\ 11 & 10 \end{bmatrix}$ and $\begin{bmatrix} 1 & -1 \\ 12 & 10 \end{bmatrix}$ are 1, -1.2 and 2.2, and since one of these is negative, the determinant condition correctly tells us that we do not have DIC.

For this 4×4 example the RGA is inconclusive:

$$G_2(0) = \begin{bmatrix} 8.72 & 2.81 & 2.98 & -15.80 \\ 6.54 & -2.92 & 2.50 & -20.79 \\ -5.82 & 0.99 & -1.48 & -7.51 \\ -7.23 & 2.92 & 3.11 & 7.86 \end{bmatrix} \quad \text{and} \quad \Lambda(G_2(0)) = \begin{bmatrix} 0.41 & 0.47 & -0.06 & 0.17 \\ -0.20 & 0.45 & 0.32 & 0.44 \\ 0.40 & 0.08 & 0.17 & 0.35 \\ 0.39 & 0.001 & 0.57 & 0.04 \end{bmatrix}$$

All the diagonal RGA values are positive, so it is inconclusive when it comes to DIC. However, the Niederlinski index of the gain matrix is negative, $N_I(G_2(0)) = -18.65$, and we conclude that this pairing is not DIC (further evaluation of the 3×3 and 2×2 submatrices is not necessary in this case).

9. The above results, including the requirement that we should pair on positive RGA elements, give *necessary* conditions for DIC. If we assume that the controllers have integral action, then $T(0) = I$, and we can derive from (10.72) that a *sufficient condition for DIC* is that G is generalized diagonally dominant at steady-state, i.e.

$$\mu(E(0)) < 1$$

This is proved by Braatz (1993, p. 154). Since the requirement is only sufficient for DIC, it cannot be used to eliminate designs.

10. If the plant has $j\omega$-axis poles, e.g. integrators, it is recommended that, prior to the RGA analysis, these are moved slightly into the LHP (e.g. by using very low-gain feedback). This will have no practical significance for the subsequent analysis.
11. Since Theorem 6.7 applies to unstable plants, we may also easily extend Theorem 10.6 to unstable plants (and in this case one may actually desire to pair on a negative RGA element). This is shown in Hovd and Skogestad (1994). Alternatively, one may first implement a stabilizing controller and then analyze the partially controlled system as if it were the plant $G(s)$.

10.6.6 RHP-zeros and RGA: reasons for avoiding negative RGA elements with sequential design

So far we have considered decentralized control based on independent design, where we require that the individual loops are stable and that we do not get instability as loops are closed or taken out of service. This led to the integrity (DIC) result of avoiding pairing on negative RGA elements at steady state. However, if we use sequential design, then the "inner" loops should *not* be taken out of service, and one may even end up with loops that are unstable by themselves (if the inner loops were to be removed). Nevertheless, for sequential design we find that it is also generally undesirable to pair on negative RGA elements, and the purpose of this section is primarily to illustrate this, by using some results that link the RGA and RHP-zeros.

Bristol (1966) claimed that negative values of $\lambda_{ii}(0)$ imply the presence of RHP-zeros, but did not provide any proof. However, it is indeed true as illustrated by the following two theorems.

Theorem 10.7 *(Hovd and Skogestad, 1992) Consider a transfer function matrix $G(s)$ with no zeros or poles at $s = 0$. Assume that $\lim_{s\to\infty} \lambda_{ij}(s)$ is finite and different from zero. If $\lambda_{ij}(j\infty)$ and $\lambda_{ij}(0)$ have different signs then at least one of the following must be true:*
(a) The element $g_{ij}(s)$ has a RHP-zero.

(b) The overall plant $G(s)$ has a RHP-zero.
(c) The subsystem with input j and output i removed, $G^{ij}(s)$, has a RHP-zero.

Theorem 10.8 *(Grosdidier et al., 1985) Consider a stable transfer function matrix $G(s)$ with elements $g_{ij}(s)$. Let $\widehat{g}_{ij}(s)$ denote the closed-loop transfer function between input u_j and output y_i with all the other outputs under integral control. Assume that: (i) $g_{ij}(s)$ has no RHP-zeros, (ii) the loop transfer function GK is strictly proper, (iii) all other elements of $G(s)$ have equal or higher pole excess than $g_{ij}(s)$. We then have:*

If $\lambda_{ij}(0) < 0$, then for $\widehat{g}_{ij}(s)$ the number of RHP-poles plus RHP-zeros is odd.

Note that $\widehat{g}_{ij}(s)$ in Theorem 10.8 is the same as the transfer function P_u from u_1 to y_1 for the partially controlled system in (10.26).

Sequential design and RHP-zeros. We design and implement the diagonal controller by tuning and closing one loop at a time in a sequential manner. Assume that we end by pairing on a *negative* steady-state RGA element, $\lambda_{ij}(0) < 0$, and that the corresponding element $g_{ij}(s)$ has no RHP-zero. Then we have the following implications:

(a) If we have integral action (as we normally have), then we will get a RHP-zero in $\widehat{g}_{ij}(s)$ which will limit the performance in the "£nal" output y_i (follows from Theorem 10.8). However, the performance limitation is less if the inner loop is tuned suf£ciently fast (Cui and Jacobsen, 2002), see also Example 10.22.

(b) If $\lambda_{ij}(\infty)$ is positive (it is usually close to 1, see pairing rule 1), then irrespective of integral action, we have a RHP-zero in $G^{ij}(s)$, which will also limit the performance in the *other* outputs (follows from Theorem 10.7).

In conclusion, for performance we should avoid ending up by pairing on a negative RGA element.

Example 10.22 **Negative RGA element and RHP-zeros.** *Consider a plant with*

$$G(s) = \frac{1}{s+10}\begin{bmatrix} 4 & 4 \\ 2 & 1 \end{bmatrix} \quad \Lambda(s) = \begin{bmatrix} -1 & 2 \\ 2 & -1 \end{bmatrix}$$

Note that the RGA is independent of frequency for this plant, so $\lambda_{11}(0) = \lambda_\infty = 1$. We want to illustrate that pairing on negative RGA elements gives performance problems. We start by closing the loop from u_1 to y_1 with a controller $u_1 = k_{11}(s)(r_1 - y_1)$. For the partially controlled system, the resulting transfer function from u_2 to y_2 ("outer loop") is

$$\widehat{g}_{22}(s) = g_{22}(s) - \frac{k_{11}(s)g_{21}(s)g_{12}(s)}{1 + g_{11}(s)k_{11}(s)}$$

With an integral controller $k_{11}(s) = K_I/s$, we £nd, as expected from Theorem 10.8, that

$$\widehat{g}_{22}(s) = \frac{s^2 + 10s - 4K_I}{(s+10)(s^2 + 10s + 4K_I)}$$

always has a RHP-zero. For large values of K_I, the RHP-zero moves further away, and is less limiting in terms of performance for the outer loop. With a proportional controller, $k_{11}(s) = K_c$, we £nd that

$$\widehat{g}_{22}(s) = \frac{s + 10 - 4K_c}{(s+10)(s + 10 + 4K_c)}$$

has a zero at $4K_c - 10$. For $K_c < 2.5$, the zero is in the LHP, but it crosses into the RHP, when K_c exceeds 2.5. For large values of K_c, the RHP-zero moves further away, and does not limit the performance in the outer loop in practice. The worst value is $K_c = 2.5$, where we have a zero at the origin and the steady-state gain $\widehat{g}_{22}(0)$ changes sign.

10.6.7 Performance of decentralized control systems

Consider again the factorization

$$S = (I + \widetilde{S}(\Gamma - I))^{-1}\widetilde{S}\Gamma$$

in (10.69) where $\Gamma = \widetilde{G}G^{-1}$ is the performance relative gain array (PRGA), The diagonal elements of the PRGA matrix are equal to the diagonal elements of the RGA, $\gamma_{ii} = \lambda_{ii}$, and this is the reason for its name. Note that the off-diagonal elements of the PRGA depend on the relative scaling on the outputs, whereas the RGA is scaling independent. On the other hand, the PRGA also measures one-way interaction, whereas the RGA only measures two-way interaction. At frequencies where feedback is effective ($\widetilde{S} \approx 0$), (10.69) yields $S \approx \widetilde{S}\Gamma$ Thus, large elements in the PRGA (Γ) (compared to 1 in magnitude) mean that the interactions "slow down" the overall response and cause performance to be worse than for the individual loops. On the other hand, small PRGA elements (compared to 1 in magnitude) mean that the interactions actually improve performance at this frequency.

We will also make use of the related closed-loop disturbance gain (CLDG) matrix, de£ned as

$$\widetilde{G}_d(s) \triangleq \Gamma(s)G_d(s) = \widetilde{G}(s)G^{-1}(s)G_d(s) \tag{10.85}$$

The CLDG depends on both output and disturbance scaling.

In the following, we consider performance in terms of the control error

$$e = y - r = Gu + G_d d - r \tag{10.86}$$

Suppose the system has been scaled as outlined in Section 1.4, such that at each frequency:

1. Each disturbance is less than 1 in magnitude, $|d_k| < 1$.
2. Each reference change is less than the corresponding diagonal element in R, $|r_j| < R_j$.
3. For each output the acceptable control error is less than 1, $|e_i| < 1$.

Single disturbance. Consider a single disturbance, in which case G_d is a vector, and let g_{di} denote the i'th element of G_d. Let $L_i = g_{ii}k_i$ denote the loop transfer function in loop i. Consider frequencies where feedback is effective so $\widetilde{S}\Gamma$ is small (and (10.89) is valid). Then for acceptable disturbance rejection ($|e_i| < 1$) with decentralized control, we must require for each loop i,

$$|1 + L_i| > |\widetilde{g}_{di}| \tag{10.87}$$

which is the same as the SISO condition (5.77) except that G_d is replaced by the CLDG, $\widetilde{g}_{di}$. In words, $\widetilde{g}_{di}$ gives the "apparent" disturbance gain as seen from loop i when the system is controlled using decentralized control.

Single reference change. We can similarly address a change in reference for output j of magnitude R_j and consider frequencies where feedback is effective (and (10.89) is valid). Then for acceptable reference tracking ($|e_i| < 1$) we must require for each loop i

$$|1 + L_i| > |\gamma_{ij}| \cdot |R_j| \tag{10.88}$$

which is the same as the SISO condition (5.80) except for the PRGA factor, $|\gamma_{ij}|$. In other words, when the other loops are closed the response in loop i gets slower by a factor $|\gamma_{ii}|$. Consequently, for *performance* it is desirable to have *small* elements in Γ, at least at frequencies where feedback is effective. However, at frequencies close to crossover, stability is the main issue, and since the diagonal elements of the PRGA and RGA are equal, we usually prefer to have $\gamma_{ii} = \lambda_{ii}$ close to 1 (see pairing rule 1 on page 450).

Proofs of (10.87) and (10.88): At frequencies where feedback is effective, $\widetilde{S}$ is small, so

$$I + \widetilde{S}(\Gamma - I) \approx I \tag{10.89}$$

and from (10.69) we have

$$S \approx \widetilde{S}\Gamma \tag{10.90}$$

The closed-loop response then becomes

$$e = SG_d d - Sr \approx \widetilde{S}\widetilde{G}_d d - \widetilde{S}\Gamma r \tag{10.91}$$

and the response in output i to a single disturbance d_k and a single reference change r_j is

$$e_i \approx \widetilde{s}_i \widetilde{g}_{dik} d_k - \widetilde{s}_i \gamma_{ik} r_k \tag{10.92}$$

where $\widetilde{s}_i = 1/(1 + g_{ii}k_i)$ is the sensitivity function for loop i by itself. Thus, to achieve $|e_i| < 1$ for $|d_k| = 1$ we must require $|\widetilde{s}_i \widetilde{g}_{dik}| < 1$ and (10.87) follows. Similarly, to achieve $|e_i| < 1$ for $|r_j| = |R_j|$ we must require $|s_i \gamma_{ik} R_j| < 1$ and (10.88) follows. Also note that $|s_i \gamma_{ik}| < 1$ will imply that assumption (10.89) is valid. Since R usually has all of its elements larger than 1, in most cases (10.89) will be automatically satisfied if (10.88) is satisfied, so we normally need not check assumption (10.89). □

Remark 1 Relation (10.90) may also be derived from (10.66) by assuming $\widetilde{T} \approx I$ which yields $(I + E\widetilde{T})^{-1} \approx (I + E)^{-1} = \Gamma$.

Remark 2 Consider a particular disturbance with model g_d. Its effect on output i with no control is g_{di}, and the ratio between $\widetilde{g}_{di}$ (the CLDG) and g_{di} is the *relative disturbance gain* (RDG) (β_i) of Stanley et al. (1985) (see also Skogestad and Morari (1987b)):

$$\beta_i \triangleq \widetilde{g}_{di}/g_{di} = [\widetilde{G}G^{-1}g_d]_i/[g_d]_i \tag{10.93}$$

Thus β_i, which is scaling independent, gives the *change* in the effect of the disturbance caused by decentralized control. It is desirable to have β_i small, as this means that the interactions are such that they reduce the apparent effect of the disturbance, such that one does not need high gains $|L_i|$ in the individual loops.

10.6.8 Summary: pairing selection and controllability analysis for decentralized control

When considering decentralized diagonal control of a plant, one should first check that the plant is controllable with any controller, see Section 6.11.

If the plant is unstable, then it recommended that a lower-layer stabilizing controller is first implemented, at least for the "fast" unstable modes. The pole vectors (page 412) are useful in selecting which inputs and outputs to use for stabilizing control. Note that some unstable plants are not stabilizable with a *diagonal* controller. This happens if the unstable modes belong to the "decentralized fixed modes", which are the modes unaffected by diagonal feedback control (e.g. Lunze (1992)). A simple example is a triangular plant where the unstable mode appears only in the off-diagonal elements, but here the plant can be stabilized by changing the pairings.

10.6.9 Independent design

We £rst consider the case of independent design, where the controller elements are designed based on the diagonal (paired) elements of the plant such that individual loops are stable.

The £rst step is to determine if one can £nd a good set of input–output pairs bearing in mind the following three pairing rules:

> **Pairing rule 1. RGA at crossover frequencies.** *Prefer pairings such that the rearranged system, with the selected pairings along the diagonal, has an RGA matrix close to identity at frequencies around the closed-loop bandwidth.*

To help in identifying the pairing with RGA closest to identity, one may, at the bandwidth frequency, compute the iterative RGA, $\Lambda^k(G)$; see Exercise 10.6.4 on page 441.

Pairing rule 1 is to ensure that we have diagonal dominance where interactions from other loops do not cause instability. Actually, pairing rule 1 does not ensure this, see the Remark on page 442, and to ensure stability we may instead require that the rearranged plant is triangular at crossover frequencies. However, the RGA is simple and only requires one computation, and since (a) all triangular plants have RGA $=I$ and (b) there is at most one choice of pairings with RGA $=I$ at crossover frequencies, we do nothing wrong in terms of missing good pairing alternatives by following pairing rule 1. To check for diagonal dominance of a promising pairing (with RGA $=I$) one may subsequently compute $\mu(E_S) = \mu(\text{PRGA} - I))$ to check if it is smaller than 1 at crossover frequencies.

> **Pairing rule 2.** *For a stable plant avoid pairings that correspond to negative steady-state RGA elements, $\lambda_{ij}(0) < 0$.*

This rule follows because we require integrity (DIC) with independent design (page 443), and also because we would like to avoid the introduction of RHP-zeros with sequential design (page 446).

Remark. Even if we have $\lambda_{ii}(0) = 1$ and $\lambda_{ii}(\infty) = 1$ for all i, this does not necessarily mean that the diagonal pairing is the best, even for a 2×2 plant. The reason for this is that the behaviour at "intermediate" bandwidth frequencies is more important. This was illustrated in Example 3.11, where we found from the frequency-dependent RGA in Figure 3.8 (page 86) that the off-diagonal pairing is preferable, because it has RGA close to identity at the bandwidth frequencies.

> **Pairing rule 3.** *Prefer a pairing ij where g_{ij} puts minimal restrictions on the achievable bandwidth. Speci£cally, the effective delay θ_{ij} in $g_{ij}(s)$ should be small.*

This rule favours pairing on variables physically "close to each other", which makes it easier to use high-gain feedback and satisfy (10.87) and (10.88), while at the same time achieving stability in each loop. It is also consistent with the desire that $\Lambda(j\omega)$ is close to I at crossover frequencies. Pairing rule 3 implies that we should avoid pairing on elements with high order, a time delay or a RHP-zero, because these result in an increased effective delay; see page 58. Goodwin et al. (2005) discuss performance limitations of independent design, in particular when pairing rule 3 is violated.

When a reasonable choice of pairings has been found (if possible), one should rearrange G to have the paired elements along the diagonal and perform a controllability analysis as follows.

1. Compute the PRGA ($\Gamma = \widetilde{G}G^{-1}$) and CLDG ($\widetilde{G}_d = \Gamma G_d$), and plot these as functions of frequency. For systems with many loops, it is best to perform the analysis one loop at a time. That is, for each loop i, plot $|\widetilde{g}_{dik}|$ for each disturbance k and plot $|\gamma_{ij}|$ for each reference j (assuming here for simplicity that each reference is of unit magnitude). For performance, see (10.88) and(10.87), we need $|1 + L_i|$ to be larger than each of these

$$\text{Performance :} \quad |1 + L_i| > \max_{k,j}\{|\widetilde{g}_{dik}|, |\gamma_{ij}|\} \tag{10.94}$$

   To achieve stability of the individual loops one must analyze $g_{ii}(s)$ to ensure that the bandwidth required by (10.94) is achievable. Note that RHP-zeros in the diagonal elements may limit achievable decentralized control, whereas they may not pose any problems for a multivariable controller. Since with decentralized control we usually want to use simple controllers, the achievable bandwidth in each loop will be limited by the effective delay θ_{ij} in $g_{ij}(s)$.
2. In general, see rule 5.13 on page 207, one may check for constraints by considering the elements of $G^{-1}G_d$ and making sure that they do not exceed 1 in magnitude within the frequency range where control is needed. Equivalently, one may plot $|g_{ii}|$ for each loop i, and the requirement is then

$$\text{To avoid input constraints :} \quad |g_{ii}| > |\widetilde{g}_{dik}|, \quad \forall k \tag{10.95}$$

   at frequencies where $|\widetilde{g}_{dik}|$ is larger than 1 (this follows since $\widetilde{G}_d = \widetilde{G}G^{-1}G_d$). This provides a direct generalization of the requirement $|G| > |G_d|$ for SISO systems. The advantage of (10.95) compared to using $G^{-1}G_d$ is that we can limit ourselves to frequencies where control is needed to reject the disturbance (where $|\widetilde{g}_{dik}| > 1$).

If the plant is not controllable with any choice of pairings, then one may consider another pairing choice and go back to step 1. Most likely this will not help, and one would need to consider decentralized sequential design, or multivariable control.

If the chosen pairing *is* controllable then the analysis based on (10.94) tells us directly how large the loop gain $|L_i| = |g_{ii}k_i|$ must be, and this can be used as a basis for designing the controller $k_i(s)$ for loop i.

10.6.10 Sequential design

Sequential design may be applied when it is not possible to £nd a suitable set of pairings for independent design using the above three pairing rules. For example, with sequential design one may choose to pair on an element with $g_{ii} = 0$ (and $\lambda_{ii} = 0$), which violates both pairing rules 1 and 3. One then relies on the interactions to achieve the desired performance, as loop i by itself has no effect. This was illustrated for the case with off-diagonal pairings in Example 10.15 on page 434. Another case with pairing on a zero element is in distillation control when the LV-con£guration is *not* used, see Example 10.8. One may also in some cases pair on negative steady-state RGA elements, although we have established that to avoid introducing RHP-zeros one should avoid closing a loop on a negative steady-state RGA (see page 447).

The procedure and rules for independent design can be used as a starting point for £nding good pairings for sequential design. With sequential design, one also has to decide the order in which the loops are closed, and one generally starts by closing the fast loops. This favours starting with a pairing where g_{ij} has good controllability, including a large gain and a small effective delay. One may also consider the disturbance gain to find which outputs need to be tightly controlled. After closing one loop, one needs to obtain the transfer function for the resulting partially controlled system, see (10.28), and then redo the analysis in order to select the next pairing, and so on.

Example 10.23 **Application to distillation process.** *In order to demonstrate the use of the frequency-dependent RGA and CLDG for evaluation of expected diagonal control performance, we again consider the distillation process used in Example 10.8. The* LV*-configuration is used; that is, the manipulated inputs are reflux* L (u_1) *and boilup* V (u_2)*. The outputs are the product compositions* y_D (y_1) *and* x_B (y_2)*. Disturbances in feed flow rate* F (d_1) *and feed composition* z_F (d_2) *are included in the model. The disturbances and outputs have been scaled such that a magnitude of* 1 *corresponds to a change in* F *of* 20%*, a change in* z_F *of* 20%*, and a change in* x_B *and* y_D *of* 0.01 *mole fraction units. The five state dynamic model is given in Section 13.4.*

Initial controllability analysis. $G(s)$ *is stable and has no RHP-zeros. The plant and RGA matrix at steady-state are*

$$G(0) = \begin{bmatrix} 87.8 & -86.4 \\ 108.2 & -109.6 \end{bmatrix} \quad \Lambda(0) = \begin{bmatrix} 35.1 & -34.1 \\ -34.1 & 35.1 \end{bmatrix} \tag{10.96}$$

The RGA elements are much larger than 1 *and indicate a plant that is fundamentally difficult to control (recall property C1, page 89). Fortunately, the flow dynamics partially decouple the response at higher frequencies, and we find that* $\Lambda(j\omega) \approx I$ *at frequencies above about* 0.5 *rad/min. Therefore if we can achieve sufficiently fast control, the large steady-state RGA elements may be less of a problem.*

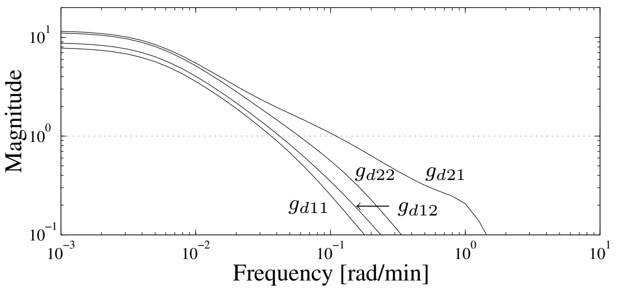


Figure 10.18: Disturbance gains $|g_{dik}|$ for assessing the effect of disturbance k on output i

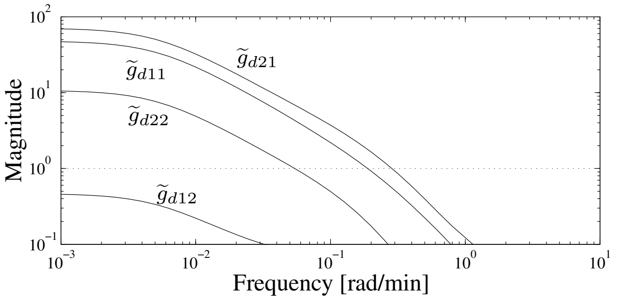


Figure 10.19: Closed-loop disturbance gains $|\widetilde{g}_{dik}|$ for assessing the effect of disturbance k on output i

The steady-state effect of the two disturbances is given by

$$G_d(0) = \begin{bmatrix} 7.88 & 8.81 \\ 11.72 & 11.19 \end{bmatrix} \tag{10.97}$$

and the magnitudes of the elements in $G_d(j\omega)$ are plotted as functions of frequency in Figure 10.18. From this plot the two disturbances seem to be equally difficult to reject with magnitudes larger than 1 up to a frequency of about 0.1 rad/min. We conclude that control is needed up to 0.1 rad/min. The magnitude of the elements in $G^{-1}G_d(j\omega)$ (not shown) are all less than 1 at all frequencies (at least up to 10 rad/min), and so it will be assumed that input constraints pose no problem.

Choice of pairings. *The selection of u_1 to control y_1 and u_2 to control y_2 corresponds to pairing on positive elements of $\Lambda(0)$ and $\Lambda(j\omega) \approx I$ at high frequencies. This seems sensible, and is used in the following.*

Analysis of decentralized control. *The elements in the CLDG and PRGA matrices are shown as functions of frequency in Figures 10.19 and 10.20. At steady-state we have*

$$\Gamma(0) = \begin{bmatrix} 35.1 & -27.6 \\ -43.2 & 35.1 \end{bmatrix}, \quad \widetilde{G}_d(0) = \Gamma(0)G_d(0) = \begin{bmatrix} -47.7 & -0.40 \\ 70.5 & 11.7 \end{bmatrix} \tag{10.98}$$

In this particular case, the off-diagonal elements of RGA (Λ) and PRGA (Γ) are quite similar. We note that $\widetilde{G}_d(0)$ is very different from $G_d(0)$, and this also holds at higher frequencies. For disturbance 1 (first column in $\widetilde{G}_d$) we find that the interactions increase the apparent effect of the disturbance, whereas they reduce the effect of disturbance 2, at least on output 1.

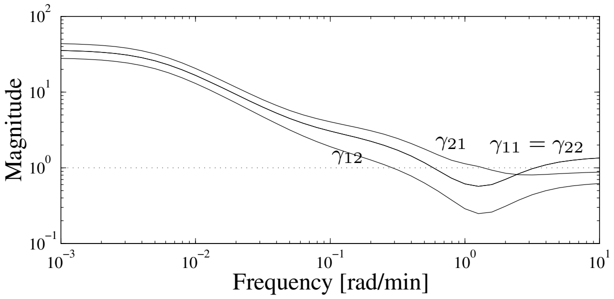


Figure 10.20: PRGA elements $|\gamma_{ij}|$ for effect of reference j on output i

We now consider one loop at a time to find the required bandwidth. For loop 1 (output 1) we consider γ_{11} and γ_{12} for references, and $\widetilde{g}_{d11}$ and $\widetilde{g}_{d12}$ for disturbances. Disturbance 1 is the most difficult, and we need $|1+L_1| > |\widetilde{g}_{d11}|$ at frequencies where $|\widehat{g}_{d11}|$ is larger than 1, which is up to about 0.2 rad/min. The magnitudes of the PRGA elements are somewhat smaller than $|\widetilde{g}_{d11}|$ (at least at low frequencies), so reference tracking will be achieved if we can reject disturbance 1. From $\widetilde{g}_{d12}$ we see that disturbance 2 has almost no effect on output 1 under feedback control.

Also, for loop 2 we find that disturbance 1 is the most difficult, and from $\widetilde{g}_{d12}$ we require a loop gain larger than 1 up to about 0.3 rad/min. A bandwidth of about 0.2 to 0.3 rad/min in each loop is required for rejecting disturbance 1, and should be achievable in practice.

Observed control performance. *To check the validity of the above results we designed two single-loop PI controllers:*

$$k_1(s) = 0.261\frac{1+3.76s}{3.76s}; \quad k_2(s) = -0.375\frac{1+3.31s}{3.31s} \tag{10.99}$$

The loop gains, $L_i = g_{ii}k_i$, with these controllers are larger than the closed-loop disturbance gains, $|\delta_{ik}|$, at frequencies up to crossover. Closed-loop simulations with these controllers are shown in Figure 10.21. The simulations confirm that disturbance 2 is more easily rejected than disturbance 1.

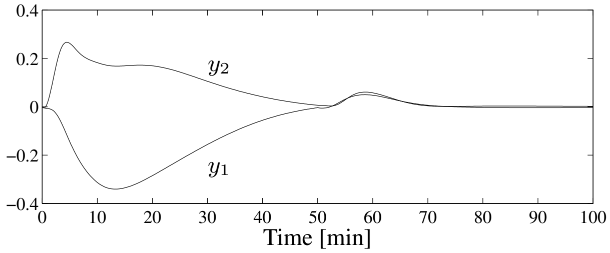


Figure 10.21: Decentralized PI control. Responses to a unit step in d_1 at $t = 0$ and a unit step in d_2 at $t = 50$ min.

In summary, there is an excellent agreement between the controllability analysis and the simulations, as has also been con£rmed by a number of other examples.

10.6.11 Conclusions on decentralized control

In this section, we have derived a number of conditions for the stability, e.g. (10.72) and (10.79), and performance, e.g. (10.87) and (10.88), of decentralized control systems. The conditions may be useful in determining appropriate pairings of inputs and outputs and the sequence in which the decentralized controllers should be designed. Recall, however, that in many practical cases decentralized controllers are tuned off-line, and sometimes on-line, using local models. In such cases, the conditions may be used in an input–output controllability analysis to determine the viability of decentralized control.

Some exercises which include a controllability analysis of decentralized control are given at the end of Chapter 6.

10.7 Conclusion

Control structure design is very important in applications, but it has traditionally received little attention in the control community. In this chapter, we have discussed the issues involved, and we have provided some results and rules, dos and don’ts, which we believe will be helpful in practice. However, there is still a need for improved tools and theory in this important area.