> solution, yet has to be done in ignorance of this solution, which can then turn out to be unsuitable in ways that were not foreseen.

Therefore, control system design usually proceeds iteratively through the steps of modelling, control structure design, controllability analysis, performance and robustness weights selection, controller synthesis, control system analysis and nonlinear simulation. Rosenbrock (1974) makes the following observation:

> Solutions are constrained by so many requirements that it is virtually impossible to list them all. The designer finds himself threading a maze of such requirements, attempting to reconcile conflicting demands of cost, performance, easy maintenance, and so on. A good design usually has strong aesthetic appeal to those who are competent in the subject.

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 available quantitative methods, for example, for decentralized control.

10.1 Introduction

Control structure design was considered by Foss (1973) in his paper entitled “Critique of 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. Later Morari et al. (1980) presented an overview of control structure design, hierarchical control and multilevel optimization in their paper “Studies in the synthesis of control structure for chemical processes”, but the gap still remained, and still does to some extent today.

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. For example, in the late 1980s Carl Nett (Nett, 1989; Nett and Minto, 1989) gave a number of lectures based on his experience on 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 outpaces 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.

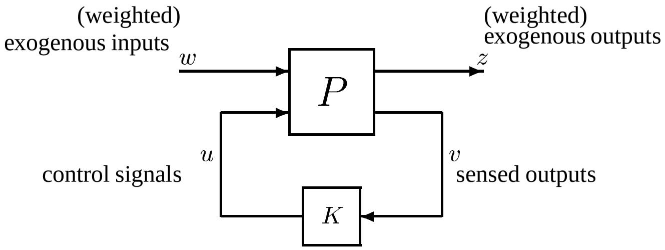


Figure 10.1: General control configuration

In Chapter 3.8 we considered the general control problem formulation in Figure 10.1, and stated that the controller design problem is to

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

However, if we go back to Chapter 1 (page 1), then we see that this is only Step 7 in the overall process of designing a control system. In this chapter we are concerned with the structural decisions (Steps 4, 5, 6 and 7) associated with the following tasks of *control structure design*:

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?*
2. The selection of manipulations 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 v in Figure 10.1?*
3. The selection of a *control configuration* (a structure interconnecting measurements/commands and manipulated variables; see sections 10.6, 10.7 and 10.8): *What is the structure of K in Figure 10.1, that is, how should we "pair" the variable sets u and v?*
4. The selection of a *controller type* (control law specification, e.g. PID-controller, decoupler, LQG, etc): *What algorithm is used for K in Figure 10.1?*

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. On the other hand, the *control configuration* refers only to the structuring (decomposition) of the controller K itself (also called the measurement/manipulation partitioning or input/output pairing). Control configuration issues are discussed in more detail in Section 10.6.

The selection of controlled outputs, manipulations and measurements (tasks 1 and 2 combined) is sometimes called *input/output selection*.

Ideally, the tasks involved in designing a complete control system are performed sequentially; first a "top-down" selection of controlled outputs, measurements and inputs (with little regard to the configuration of the controller K) and then a "bottom-up" design of the control system (in which the selection of the control configuration is the most important decision). However, in practice the tasks are closely related in that one decision directly influences the others, so the procedure may involve iteration.

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 may 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.

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 from physical insight obtain a reasonable choice of controlled outputs, measurements and manipulated inputs. In other cases, simple controllability measures as presented in Chapters 5 and 6 may be used for quickly evaluating or screening alternative control structures.

Some discussion on control structure design in the process industry is given by Morari (1982), Shinskey (1988), Stephanopoulos (1984) and Balchen and Mumme (1988). A survey on control structure design is given by van de Wal and de Jager (1995). 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 160) for an overview of the literature on input-output controllability analysis.

10.2 Optimization and control

In Sections 10.2 and 10.3 we are concerned with the selection of controlled variables (outputs). These are the variables z in Figure 10.1, but we will in these two sections call them y.

The selection of controlled outputs involves selecting the variables y to be controlled at given reference values, $y \approx r$. Here the reference value r is set at 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 which is often divided into two layers:

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

Additional layers are possible, as is illustrated in Figure 10.2 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. In general, the information flow in such a control hierarchy is based on the higher layer sending reference values (setpoints) to the layer below, and the lower layer reporting back any problems in achieving this, see Figure 10.3(b).

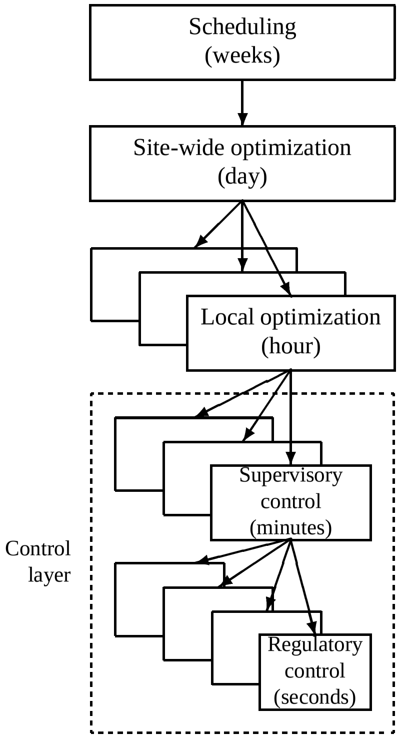


Figure 10.2: Typical control system hierarchy in a chemical plant

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 we often use linear dynamic models in the control layer (as we do throughout the book).

There is usually a time scale separation with faster lower layers as indicated in Figure 10.2. This means that the setpoints, as viewed from a given layer in the hierarchy, are updated only periodically. Between these updates, when the setpoints are constant, it is important that the system remains reasonably close to its optimum. This observation is the basis for Section 10.3 which deals with selecting outputs for the control layer.

From a theoretical point of view, the optimal coordination of the inputs and thus the optimal performance is obtained with a *centralized optimizing controller*, which combines the two layers of optimization and control; see Figure 10.3(c). 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.

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

Mesarovic (1970) reviews some ideas related to on-line multi-layer structures applied to large-scale industrial complexes. However, according to Lunze (1992), multilayer structures, although often used in practice, lack a formal analytical treatment. Nevertheless, in the next section we provide some ideas on how to select objectives (controlled outputs) for the control layer, such that the overall goal is satisfied.

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 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 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.

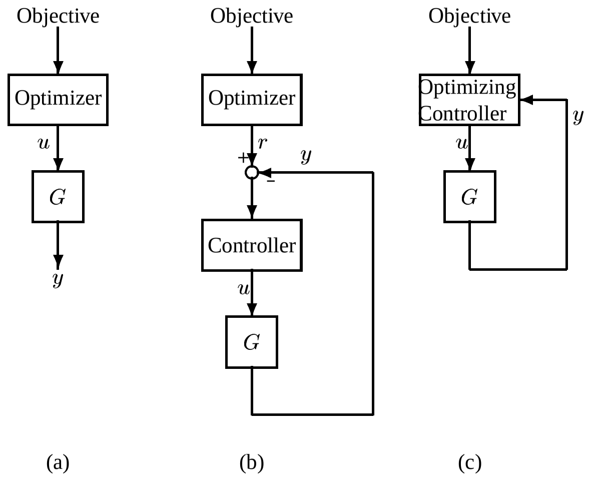


Figure 10.3: Alternative structures for optimization and control. (a) Open-loop optimization. (b) Closed-loop implementation with separate control layer. (c) Integrated optimization and control.

10.3 Selection of controlled outputs

A *controlled output* is an output variable (usually measured) with an associated control objective (usually a reference value). In many cases, it is clear from a physical understanding of the process what the controlled outputs should be. For example, if we consider heating or cooling a room, then we should select room temperature as the controlled output y. In other cases it is less obvious because each control objective may not be associated with a measured output variable. Then the controlled outputs y are selected to achieve the *overall system goal*, and may not appear to be important variables in themselves.

Example 10.1 Cake baking. *To get an idea of the issues involved in output selection let us consider the process of baking a cake. The overall goal is to make a cake which is well baked inside and with 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 fixed, e.g. at 15 minutes). Now, if we had never baked a cake before, and if we were to construct the stove ourselves, we might consider directly manipulating the heat input to the stove, 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, from opening the oven door or whatever else might be in the oven. In shortm the open-loop implementation is sensitive to uncertainty. An effective way of reducing the uncertainty is to use feedback. Therefore, in practice we look up the optimal oven temperature in a cook book, and use a* closed-loop *implementation where a thermostat is used to keep the temperature y at its predetermined value T.*

The (a) open-loop and (b) closed-loop implementations of the cake baking process are illustrated in Figure 10.3. In (b) the "optimizer" is the cook book which has a pre-computed table of the optimal temperature profile. The reference value r for temperature is then sent down to the control layer which consists of a simple feedback controller (the thermostat).

Recall that the title of this section is selection of controlled outputs. In the cake baking process we select *oven temperature* as the controlled output y 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.

In the following, we let y denote the selected controlled outputs in the control layer. Note that this may also include directly using the inputs (open-loop implementation) by selecting $y = u$. Two distinct questions arise:

1. What variables y should be selected as the controlled variables?
2. What is the optimal reference value $(y_{\rm opt})$ 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 first problem. We make the following *assumptions*:

(a) The overall goal can be quantified in terms of a scalar cost function J which we want to minimize.
(b) For a given disturbance d, there exists an optimal value $u_{\rm opt}(d)$ and corresponding value $y_{\rm opt}(d)$ which minimizes the cost function J.
(c) The reference values r for the controlled outputs y should be constant, i.e. r should be independent of the disturbances d. Typically, some average value is selected, e.g. $r = y_{\rm opt}(\bar{d})$

For example, in the cake baking process we may assign to each cake a number P on a scale from 0 to 10, based on cake quality. A perfectly baked cake achieves $P = 10$, and an acceptably baked cake achieves $P > 6$ (a completely burned cake may correspond to $P = 1$). In another case P could be the operating profit. In both cases we can select $J = -P$, and the overall goal of the control system is then to minimize J.

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

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

Ideally, we want $u = u_{\rm opt}$. However, this will not be achieved in practice, and we select controlled outputs y such that:

- *The input u (generated by feedback to achieve $y \approx r$) should be close to the optimal input $u_{\rm opt}(d)$.*

Note that we have assumed that r is independent of d.

What happens if $u \neq u_{\rm opt}$, e.g. due to a disturbance? Obviously, we then have a loss which can be quantified by $L = J - J_{\rm opt}$, and a reasonable objective for selecting controlled outputs y is to minimize some norm of the loss, for example, the worst-case loss

$$\text{Worst} - \text{case loss}: \quad \Phi \triangleq \max_{d \in \mathcal{D}} |\underbrace{J(u,d) - J(u_{\rm opt}, d)}_{L}| \qquad (10.2)$$

Here $\mathcal{D}$ is the set of possible disturbances. As “disturbances” we should also include changes in operating point and model uncertainty.

10.3.1 Selecting controlled outputs: Direct evaluation of cost

The “brute force” approach for selecting controlled variables is to evaluate the loss for alternative sets of controlled variables. Specically, by solving the nonlinear equations, we evaluate directly the cost function J for various disturbances d and control errors e, assuming $y = r + e$ where r is kept constant. The set of controlled outputs with smallest worst-case or average value of J is then preferred. This approach is may be time consuming because the solution of the nonlinear equations must be repeated for each candidate set of controlled outputs.

If we with constant references (setpoints) r can achieve an acceptable loss, then this set of controlled variables is said to be *self-optimizing*. 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”).

The special case of measurement selection for *indirect control* is covered on page 439.

10.3.2 Selecting controlled outputs: Linear analysis

We here use a linear analysis of the loss function. This results in the useful minimum singular value rule. However, 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 $L = J(u,d) - J_{\rm opt}(d)$ (10.2), where d is a fixed (generally non-zero) disturbance. We make the following additional assumptions:

(d) The cost function J is smooth, or more precisely twice differentiable.

(e) The optimization problem is unconstrained. If it is optimal to keep some variable at a constraint, then we assume that this is implemented and consider the remaining unconstrained problem.

(f) The dynamics of the problem can be neglected, that is, we consider the steady-state control and optimization.

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

$$\begin{aligned} J(u,d) \quad = \quad & J_{\rm opt}(d) \; + \; \underbrace{\left(\frac{\partial J}{\partial u}\right)^T_{\rm opt}}_{=0} (u - u_{\rm opt}(d)) \; + \\ & \frac{1}{2}(u - u_{\rm opt}(d))^T \left(\frac{\partial^2 J}{\partial u^2}\right)_{\rm opt} (u - u_{\rm opt(d)}) \; + \; \cdots \end{aligned} \qquad (10.3)$$

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 (10.3) is zero at the optimal point for an unconstrained problem.

Equation (10.3) quantifies how $u - u_{\rm opt}$ affects the cost function. Next, to study how this relates to output selection we use a linearized model of the plant, which for a fixed d becomes $y - y_{\rm opt} = G(u - u_{\rm opt})$ where G is the steady-state gain matrix. If G is invertible we then get

$$u - u_{\rm opt} = G^{-1}(y - y_{\rm opt}) \qquad (10.4)$$

(If G is not invertible we may use the pseudo-inverse $G^{\dagger}$ which results in the smallest possible $\|u - u_{\rm opt}\|_2$ for a given $y - y_{\rm opt}$.) We get

$$J - J_{\rm opt} \approx \frac{1}{2}\left(G^{-1}\left(y - y_{\rm opt}\right)\right)^T \left(\frac{\partial^2 J}{\partial u^2}\right)_{\rm opt} G^{-1}\left(y - y_{\rm opt}\right) \qquad (10.5)$$

where the term $(\partial^2 J/\partial u^2)_{\rm opt}$ is independent of y. Obviously, we would like to select the controlled outputs such that $y - y_{\rm opt}$ is zero. However, this is not possible in practice. To see this, write

$$y - y_{\rm opt} = y - r + r - y_{\rm opt} = e + e_{\rm opt} \qquad (10.6)$$

First, we have an optimization error $e_{\rm opt}(d) \triangleq r - y_{\rm opt}(d)$, because the algorithm (e.g. a cook book) pre-computes a desired r which is different from the optimal $y_{\rm opt}(d)$. In addition, we have a control error $e = y - r$ because the control layer is not perfect, for example due to poor control performance or an incorrect measurement or estimate (steady-state bias) of y. If the control itself is perfect then $e = n$ (measurement noise). In most cases the errors e and $e_{\rm opt}(d)$ can be assumed independent.

Example 10.1 **Cake baking, continued.** *Let us return to our initial question: Why select the oven temperature as a controlled output? We have two alternatives: a closed-loop implementation with $y = T$ (the oven temperature) and an open-loop implementation with $y = u = Q$ (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 $T_r = 190^\circ C$, 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 $100W$ and $5000W$. A cook book would then need to list a different value of Q_r 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 find that it is much easier to keep $e_{\rm opt} = T - T_{\rm opt}$ [°C] small than to keep $Q_r - Q_{\rm opt}$ [W] small. In summary, the main reason for controlling the oven temperature is to minimize the optimization error.*

From (10.5) and (10.6), we conclude that we should select the controlled outputs y such that:

1. *G^{-1} is small (i.e. G is large); the choice of y should be such that the inputs have a large effect on y.*
2. *$e_{\rm opt}(d) = r - y_{\rm opt}(d)$ is small; the choice of y should be such that its optimal value $y_{\rm opt}(d)$ depends only weakly on the disturbances and other changes.*
3. *$e = y - r$ is small; the choice of y should be such that it is easy to keep the control error e small.*

Note that $\bar\sigma(G^{-1}) = 1/\underline{\sigma}(G)$, and so we want the smallest singular value of the steady-state gain matrix to be large (but recall that singular values depend on scaling as is discussed below). The desire to have $\underline{\sigma}(G)$ large is consistent with our intuition that we should ensure that the controlled outputs are independent of each other.

To use $\underline{\sigma}(G)$ to select controlled outputs, we see from (10.5) that we should first scale the outputs such that the expected magnitude of $y_i - y_{i_{\rm opt}}$ is similar (e.g. 1) in magnitude for each output, and scale the inputs such that the effect of a given deviation $u_j - u_{j_{\rm opt}}$ on the cost function J is similar for each input (such that $\left(\partial^2 J/\partial u^2\right)_{\rm opt}$ is close to a constant times a unitary matrix). We must also assume that the variations in $y_i - y_{i_{\rm opt}}$ are uncorrelated, or more precisely, we must assume:

(g) The "worst-case" combination of output deviations, $y_i - y_{i_{\rm opt}}$, corresponding to the direction of $\underline{\sigma}(G)$, can occur in practice.

Procedure. The use of the minimum singular value to select controlled outputs may be summarized in the following procedure:

1. From a (nonlinear) model compute the optimal parameters (inputs and outputs) for various conditions (disturbances, operating points). (This yields a "look-up" table of optimal parameter values as a function of the operating conditions.)
2. From this data obtain for each candidate output the variation in its optimal value, $v_i = (y_{i_{\rm opt,max}} - y_{i_{\rm opt,min}})/2$.

3. Scale the candidate outputs such that for each output the sum of the magnitudes of v_i and the control error (e_i, including measurement noise n_i) is similar (e.g. $|v_i| + |e_i| = 1$).
4. Scale the inputs such that a unit deviation in each input from its optimal value has the same effect on the cost function J (i.e. such that $\left(\partial^2 J/\partial u^2\right)_{\rm opt}$ is close to a constant times a unitary matrix).
5. Select as candidates those sets of controlled outputs which correspond to a large value of $\underline{\sigma}(G)$. G is the transfer function for the effect of the scaled inputs on the scaled outputs.

Note that the disturbances and measurement noise enter indirectly through the scaling of the outputs (!).

Example. *The aero-engine application in Chapter 12 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 five alternative sets of three outputs are given. The outputs are scaled as outlined above, and a good output set is then one with a large value of $\underline{\sigma}(G)$, provided we can also achieve good dynamic control performance.*

Remark. Note that 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.

10.3.3 Selection of controlled variables: Summary

Generally, the optimal values of all variables will change with time during operation (due to disturbances and other changes). For practical reasons, we have considered a hierarchical strategy where the optimization is performed only periodically. The question is then: Which variables (*controlled outputs*) should be kept constant (between each optimization)? Essentially, we found that we should select variables y for which the variation in optimal value and control error is small compared to their controllable range (the range y may reach by varying the input u). We considered two approaches for selecting controlled outputs:

1. "Brute force" evaluation to find the set with the smallest loss imposed by using constant values for the setpoints r.
2. Maximization of $\underline{\sigma}(G)$ where G is appropriately scaled (see the above procedure).

If the loss imposed by keeping constant setpoints is acceptable then we have self-optimizing control. The objective of the control layer is then to keep the controlled outputs at their reference values (which are computed by the optimization layer). The controlled outputs are often measured, but we may also estimate their values based on other measured variables. We may also use other measurements to improve the control of the controlled outputs, for example, by use of cascade control. Thus, the selection of controlled and measured outputs are two separate issues, although the two decisions are obviously closely related.

The measurement selection problem is briefly discussed in the next section. Then in section 10.5 we discuss the relative gain array of the "big" transfer matrix (with all candidate outputs included), as a useful screening tool for selecting controlled outputs.

10.4 Selection of manipulations and measurements

We are here concerned with the variable sets u and v in Figure 10.1. Note that the measurements used by the controller (v) are in general different from the controlled variables (z), because 1) we may not be able to measure all the controlled variables, and 2) we may want to measure and control additional variables in order to

- stabilize the plant (or more generally change its dynamics)
- improve local disturbance rejection

Stabilization. We usually start the controller design by designing a (lower-layer) controller to stabilize the plant. The issue is then: Which outputs (measurements) and inputs (manipulatons) should be used for stabilization? We should clearly avoid saturation of the inputs, because this makes the system effective open-loop and stabilization is impossible. A reasonable objective is therefore to minimize the required input usage of the stabilizing control system. It turns out that this is achieved, for a single unstable mode, by selecting the output (measurement) and input (manipulation) corresponding to the largest elements in the output and input pole vectors (y_p and u_p), respectively (see remark on page 2) (Havre, 1998)(Havre and Skogestad, 1998b). This choice maximizes the (state) controllability and observability of the unstable mode.

Local disturbance rejection. For measurements, the rule is generally to select those which have a strong relationship with the controlled outputs, or which may quickly detect a major disturbance and which together with manipulations can be used for local disturbance rejection.

The selected manipulations should have a large effect on the controlled outputs, and should be located "close" (in terms of dynamic response) to the outputs and measurements.

For a more formal analysis we may consider the model $y_{\rm all} = G_{\rm all} u_{\rm all} + G_{d{\rm all}} d$. Here

- $y_{\rm all}$ = all candidate outputs (measurements)
- $u_{\rm all}$ = all candidate inputs (manipulations)

The model for a particular combination of inputs and outputs is then $y = Gu + G_d d$ where

$$G = S_O G_{\rm all} S_I; \quad G_d = S_O G_{d{\rm all}} \tag{10.7}$$

Here S_O is a non-square input "selection" matrix with a 1 and otherwise 0's in each row, and S_I is a non-square output "selection" matrix with a 1 and otherwise 0's in each column. For example, with $S_O = I$ all outputs are selected, and with $S_O = [\,0 \quad I\,]$ output 1 has *not* been selected.

To evaluate the alternative combinations, one may, based on G and G_d, perform an input-output controllability analysis as outlined in Chapter 6 for each combination (e.g, consider the minimum singular value, RHP-zeros, interactions, etc). At least this may be useful for eliminating some alternatives. A more involved approach, based on analyzing achievable robust performance by neglecting causality, is outlined by Lee et al. (1995). This approach is more involved both in terms of computation time and in the effort required to define the robust performance objective. An even more involved (and exact) approach would be to synthesize controllers for optimal robust performance for each candidate combination.

However, the number of combinations has a combinatorial growth, so even a simple input-output controllability analysis becomes very time-consuming if there are many alternatives. For a plant where we want to select m from M candidate manipulations, and l from L candidate measurements, the number of possibilities is

$$\binom{L}{l}\binom{M}{m} = \frac{L!}{l!(L-l)!}\,\frac{M!}{m!(M-m)!} \tag{10.8}$$

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; for $m = l = 5$ and $M = L = 10$ it is 63504; and for $m = M, l = 5$ and $L = 100$ (selecting 5 measurements out of 100 possible) there are 75287520 possible combinations.

Remark. The number of possibilities is much larger if we consider *all* possible combinations with 1 to M inputs and 1 to L outputs. The number is (Nett, 1989): $\sum_{m=1}^{M}\sum_{l=1}^{L}\binom{L}{l}\binom{M}{m}$.For example, with $M = L = 2$ there are 4+2+2+1=9 candidates (4 structures with one input and one output, 2 structures with two inputs and one output, 2 structures with one input and two outputs, and 1 structure with two inputs and two outputs).

One way of avoiding this combinatorial problem is to base the selection directly on the "big" models $G_{\rm all}$ and $G_{d{\rm all}}$. For example, one may consider the singular value decomposition and relative gain array of $G_{\rm all}$ as discussed in the next section. This rather crude analysis may be used, together with physical insight, rules of thumb and simple controllability measures, to perform a pre-screening in order to reduce the possibilities to a manageable number. These candidate combinations can then be analyzed more carefully.

10.5 RGA for non-square plant

A simple but effective screening tool for selecting inputs and outputs, which avoids the combinatorial problem just mentioned, is the relative gain array (RGA) of the "big" transfer matrix G_{all} with all candidate inputs and outputs included, $\Lambda = G_{\text{all}} \times G_{\text{all}}^{\dagger^T}$.

Essentially, for the case of many candidate manipulations (inputs) one may consider not using those manipulations corresponding to columns in the RGA where the sum of the elements is much smaller than 1 (Cao, 1995). Similarly, for the case of many candidate measured outputs (or controlled outputs) one may consider not using those outputs corresponding to rows in the RGA where the sum of the elements is much smaller than 1.

To see this, write the singular value decomposition of G_{all} as

$$G_{\text{all}} = U\Sigma V^H = U_r \Sigma_r V_r^H \tag{10.9}$$

where Σ_r consists only of the $r = \text{rank}(G)$ non-zero singular values, U_r consists of the r first columns of U, and V_r consists of the r first columns of V. Thus, V_r consists of the input directions with a non-zero effect on the outputs, and U_r consists of the output directions we can affect (reach) by use of the inputs.

Let $e_j = [0 \quad \cdots \quad 0 \quad 1 \quad 0 \quad \cdots \quad 0]^T$ be a unit vector with a 1 in position j and 0's elsewhere. Then the j'th input is $u_j = e_j^T u$. Define e_i in a similar way such that the i'th output is $y_i = e_i^T y$. We then have that $e_j^T V_r$ yields the projection of a unit input u_j onto the effective input space of G, and we follow Cao (1995) and define

$$\text{Projection for input } j = \|e_j^T V_r\|_2 \tag{10.10}$$

which is a number between 0 and 1. Similarly, $e_i^T U_r$ yields the projection of a unit output y_i onto the effective (reachable) output space of G, and we define

$$\text{Projection for output } i = \|e_i^T U_r\|_2 \tag{10.11}$$

which is a number between 0 and 1. The following theorem links the input and output (measurement) projection to the column and row sums of the RGA.

Theorem 10.1 (RGA and input and output projections.) *The i'th row sum of the RGA is equal to the square of the i'th output projection, and the j'th column sum of the RGA is equal to the square of the j'th input projection, i.e.*

$$\sum_{j=1}^{m} \lambda_{ij} = \|e_i^T U_r\|_2^2; \quad \sum_{i=1}^{l} \lambda_{ij} = \|e_j^T V_r\|_2^2 \tag{10.12}$$

For a square non-singular matrix both the row and column sums in (10.12) are 1.

Proof: See Appendix A.4.2. □

The RGA is a useful screening tool because it need only be computed once. It includes all the alternative inputs and/or outputs and thus avoids the combinatorial problem. From (10.12) we see that the row and column sums of the RGA provide a useful way of interpreting the information available in the singular vectors. For the case of extra inputs the RGA-values depend on the input scaling, and for extra outputs on the output scaling. The variables must therefore be scaled prior to the analysis.

Example 10.2 *Consider a plant with* 2 *inputs and* 6 *candidate outputs of which we want to select* 2. *The plant and its RGA-matrix are*

$$G_{\text{all}} = \begin{bmatrix} 10 & 10 \\ 10 & 9 \\ 2 & 1 \\ 2 & -1 \\ 2 & 2 \\ 0 & 2 \end{bmatrix}, \quad \Lambda = \begin{bmatrix} -0.1050 & 0.6303 \\ 0.5742 & -0.1008 \\ 0.1317 & -0.0616 \\ 0.4034 & 0.2101 \\ -0.0042 & 0.0252 \\ 0 & 0.2969 \end{bmatrix}$$

There exist $\binom{6}{2} = 15$ *combinations with* 2 *inputs and* 2 *outputs. The RGA may be useful in providing an initial screening. The six row sums of the RGA-matrix are* $0.5252, 0.4734, 0.0700, 0.6134, 0.0210$ *and* 0.2969. *To maximize the output projection we should select outputs 1 and 4. For this selection* $\underline{\sigma}(G) = 2.12$ *whereas* $\underline{\sigma}(G_{\text{all}}) = \sigma_2(G_{\text{all}}) = 2.69$ *with all outputs included. This shows that we have not lost much gain in the low-gain direction by using only* 2 *of the* 6 *outputs. However, there are a large number of other factors that determine controllability, such as RHP-zeros, sensitivity to uncertainty, and these must be taken into account when making the final selection.*

The following example shows that although the RGA is an efficient screening tool, it must be used with some caution.

Example 10.3 *Consider a plant with* 2 *inputs and* 4 *candidate outputs of which we want to select* 2. *We have:*

$$G_{\text{all}} = \begin{bmatrix} 10 & 10 \\ 10 & 9 \\ 2 & 1 \\ 2 & 1 \end{bmatrix}, \quad \Lambda = \begin{bmatrix} -2.57 & 3.27 \\ 1.96 & -1.43 \\ 0.80 & -0.42 \\ 0.80 & -0.42 \end{bmatrix}$$

The four row sums of the RGA-matrix are 0.70, 0.53, 0.38 *and* 0.38. *Thus, to maximize the output projection we should select outputs* 1 *and* 2. *However, this yields a plant* $G_1 = \begin{bmatrix} 10 & 10 \\ 10 & 9 \end{bmatrix}$ *which is ill-conditioned with large RGA-elements,* $\Lambda(G_1) = \begin{bmatrix} -9 & 10 \\ 10 & -9 \end{bmatrix}$, *and is likely to be difficult to control. On the other hand, selecting outputs* 1 *and* 3 *yields* $G_2 = \begin{bmatrix} 10 & 10 \\ 2 & 1 \end{bmatrix}$ *which is well-conditioned with* $\Lambda(G_2) = \begin{bmatrix} -1 & 2 \\ 2 & -1 \end{bmatrix}$. *For comparison, the minimum singular values are:* $\underline{\sigma}(G_{\text{all}}) = 1.05$, $\underline{\sigma}(G_1) = 0.51$, *and* $\underline{\sigma}(G_2) = 0.70$.

We discuss on page 435 the selection of extra measurements for use in a cascade control system.

10.6 Control configuration elements

We now assume that the measurements, manipulations and controlled outputs are fixed. The available synthesis theories presented in this book result in a multivariable controller K which connects all available measurements/commands (v) with all available manipulations (u),

$$u = Kv \tag{10.13}$$

(the variables v will mostly be denoted y in the following). 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 105). 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 (1988) 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 (10.13) has a fixed block-diagonal structure.

Cascade control *is 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.

Feedforward elements *link measured disturbances and 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. In particular, this is relevant for cascade control systems, and it is sometimes also used in the design of decentralized control systems.

The choice of control configuration leads to two different ways of partitioning the control system:

- *Vertical decomposition.* This usually results from a sequential design of the control system, e.g. based on cascading (series interconnecting) the controllers in a hierarchical manner.
- *Horizontal decomposition.* This usually involves a set of independent decentralized controllers.

Remark 1 Sequential design of a decentralized controller results in a control system which is decomposed both horizontally (since K is diagonal) as well as vertically (since controllers at higher layers are tuned with lower-layer controllers in place).

Remark 2 Of course, a performance loss is inevitable if we decompose the control system. For example, for a hierarchical decentralized control system, if we select a poor configuration at the lower (base) control layer, then this may pose fundamental limitations on the achievable performance which cannot be overcome by advanced controller designs at higher layers. These limitations imposed by the lower-layer controllers may include RHP-zeros (see the aero-engine case study in Chapter 12) or strong interactions (see the distillation case study in Chapter 12 where the LV-configuration yields large RGA-elements at low frequencies).

In this section, we discuss cascade controllers and selectors, and give some justification for using such "suboptimal" configurations rather than directly designing the overall controller K. Later, in Section 10.7, we discuss in more detail the hierarchical decomposition, including cascade control, partially controlled systems and sequential controller design. Finally, in Section 10.8 we consider decentralized diagonal control.

10.6.1 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 here use single-input single-output (SISO) controllers of the form

$$u_i = K_i(s)(r_i - y_i) \tag{10.14}$$

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 (10.14) 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:

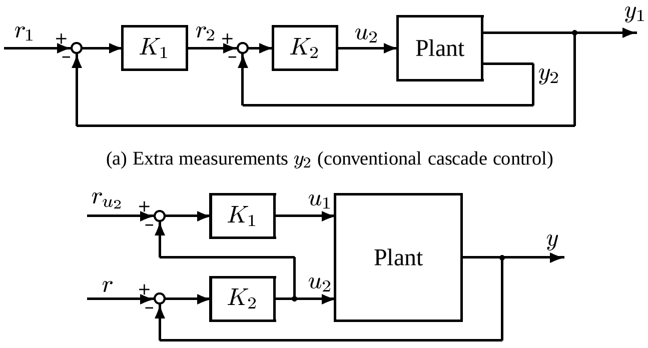
- The reference r_i is an output from another controller (typically used for the case of an extra measurement y_i), see Figure 10.4(a). This is *conventional cascade control*.
- The "measurement" y_i is an output from another controller (typically used for the case of an extra manipulated input u_j; e.g. in Figure 10.4(b) where u_2 is the "measurement" for controller K_1). This cascade scheme is referred to as *input resetting*.

10.6.2 Cascade control: Extra measurements

In many cases we make use of extra measurements y_2 (*secondary outputs*) to provide local disturbance rejection and linearization, or to reduce the effect of measurement noise. For example, velocity feedback is frequently used in mechanical systems, and local flow cascades are used in process systems. Let u be the manipulated input, y_1 the controlled output (with an associated control objective r_1) and y_2 the extra measurement.

Centralized (parallel) implementation. A centralized implementation $u = K(r - y)$, where K is a 2-input-1-output controller, may be written

$$u = K_{11}(s)(r_1 - y_1) + K_{12}(s)(r_2 - y_2) \tag{10.15}$$



(a) Extra measurements y_2 (conventional cascade control)

(b) Extra inputs u_2 (input resetting)

Figure 10.4: Cascade implementations

where in most cases $r_2 = 0$ (since we do not have a degree of freedom to control y_2).

Cascade implementation (conventional cascade control). To obtain an implementation with two SISO controllers we may cascade the controllers as illustrated in Figure 10.4(a):

$$r_2 = K_1(s)(r_1 - y_1), \tag{10.16}$$

$$u_2 = K_2(s)(r_2 - y_2),\ r_2 = \widehat{u}_1 \tag{10.17}$$

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.

With $r_2 = 0$ in (10.15) the relationship between the centralized and cascade implementation is $K_{11} = K_2K_1$ 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 more clearly 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} should have integral action in (10.15)).

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 12.2.

Remark. Consider conventional cascade control in Figure 10.4(a). In the general case y_1 and y_2 are not directly related to each other, and this is sometimes referred to as *parallel cascade*

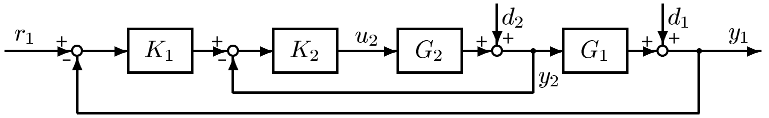


Figure 10.5: Common case of cascade control where the primary output y_1 depends directly on the extra measurement y_2.

control. However, it is common to encounter the situation in Figure 10.5 where y_1 depends directly on y_2. This is a special case of Figure 10.4(a) with "Plant"$= \begin{bmatrix} G_1G_2 \\ G_2 \end{bmatrix}$, and it is considered further in Example 10.1.

Exercise 10.1 Conventional cascade control. *With reference to the special (but common) case of conventional cascade control shown in Figure 10.5, Morari and Zafiriou (1989) conclude that the use of extra measurements is useful under the following circumstances:*

(a) The disturbance d_2 is significant and G_1 is non-minimum phase.
(b) The plant G_2 has considerable uncertainty associated with it – e.g. 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. We want to derive conclusions (a) and (b) from an input-output controllability analysis, and also, (c) explain 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_1G_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 $L_2 = G_2K_2$ removes the uncertainty if it is sufficiently fast (high gain feedback) and yields a transfer function $(I + L_2)^{-1}L_2$ close to I at frequencies where K_1 is active. (c) In most cases, such as when PID controllers are used, the practical bandwidth is limited by the frequency w_u where the phase of the plant is $-180°$ (see section 5.12), 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_1G_2.*

Exercise 10.2 *To illustrate the benefit of using inner cascades for high-order plants, case (c) in the above example, consider Figure 10.5 and let*

$$G_1 = \frac{1}{(s+1)^2}, \quad G_2 = \frac{1}{s+1}$$

We use a fast proportional controller $K_2 = 25$ in the inner loop, whereas a somewhat slower PID-controller is used in the outer loop,

$$K_1(s) = K_c\frac{(s+1)^2}{s(0.1s+1)}, \quad K_c = 5$$

Sketch the closed-loop response. What is the bandwidth for each of the two loops?

Compare this with the case where we only measure y_1, so $G = G_1G_2$, and use a PID-controller $K(s)$ with the same dynamics as $K_1(s)$ but with a smaller value of K_c. What is the achievable bandwidth? Find a reasonable value for K_c (starting with $K_c = 1$) and sketch the closed-loop response (you will see that it is about a factor 5 slower without the inner cascade).

10.6.3 Cascade control: 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 sufficient 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").

Centralized (parallel) implementation. A centralized implementation $u = K(r - y)$ where K is a 1-input 2-output controller, may be written

$$u_1 = K_{11}(s)(r-y), \quad u_2 = K_{21}(s)(r-y) \tag{10.18}$$

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 $t \to \infty$.

Cascade implementation (input resetting). To obtain an implementation with two SISO controllers we may cascade the controllers as shown in Figure 10.4(b). We again let input u_2 take care of the fast control and u_1 of the long-term control. The fast control loop is then

$$u_2 = K_2(s)(r-y) \tag{10.19}$$

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}:

$$u_1 = K_1(s)(r_{u_2} - y_1), \quad y_1 = u_2 \tag{10.20}$$

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

With $r_{u_2} = 0$ the relationship between the centralized and cascade implementation is $K_{11} = -K_1K_2$ and $K_{21} = K_2$.

The cascade implementation again has the advantage 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).

Remark 1 Typically, the controllers in a cascade system are tuned one at a time starting with the fast loop. For example, for the control system in Figure 10.6 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).

Remark 2 In process control, the cascade implementation of input resetting is sometimes referred to as *valve position control*, because the extra input u_2, usually a valve, is reset to a desired position by the outer cascade.

Exercise 10.3 *Draw the block diagrams for the two centralized (parallel) implementations corresponding to Figure 10.4.*

Exercise 10.4 *Derive the closed-loop transfer functions for the effect of r on y, u_1 and u_2 for the cascade input resetting scheme in Figure 10.4(b). As an example use $G = [G_{11} \;\; G_{12}] = [1 \;\; 1]$ 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.*

Example 10.4 **Two layers of cascade control.** *Consider the system in Figure 10.6 with two manipulated 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 (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.*

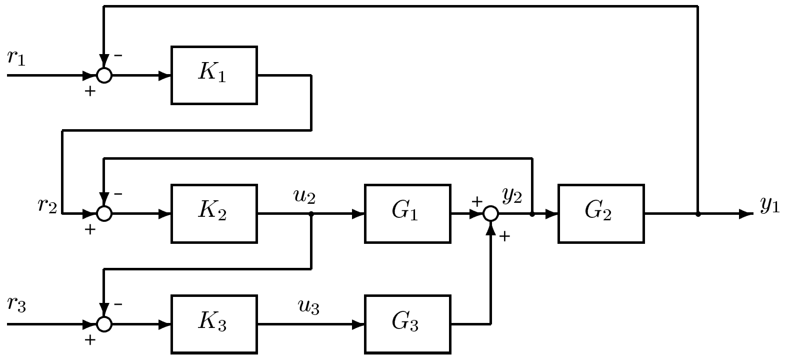


Figure 10.6: Control configuration with two layers of cascade control.

In Figure 10.6 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 corresponding equations are

$$\widehat{u}_1 = K_1(s)(r_1 - y_1) \tag{10.21}$$

$$u_2 = K_2(s)(r_2 - y_2), \quad r_2 = \widehat{u}_1 \tag{10.22}$$

$$u_3 = K_3(s)(r_3 - y_3), \quad y_3 = u_2 \tag{10.23}$$

Controller K_1 controls the primary output y_1 at its reference r_1 by adjusting the "input" $\widehat{u}_1$, 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.

Exercise 10.5 **Process control application.** *A practical case of a control system like the one in Figure 10.6 is in the use of a pre-heater to keep the 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). Make a process flowsheet with instrumentation lines (not a block diagram) for this heater/reactor process.*

10.6.4 Extra inputs and outputs (local feedback)

In many cases performance may be improved with local feedback loops involving extra manipulated inputs and extra measurements. However, the improvement must be traded off against the cost of the extra actuator, measurement and control system. An example where local feedback is required to counteract the effect of high-order lags is given for a neutralization process in Figure 5.24 on page 208. The use of local feedback is also discussed by Horowitz (1991).

10.6.5 Selectors

Split-range control for extra inputs. We assumed above that the extra input is used to improve dynamic performance. Another situation is when input constraints make it necessary to add a manipulated input. 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, this 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.

10.6.6 Why use cascade and decentralized control?

As is evident from Figure 10.6(a), decomposed control configurations can easily become quite complex and difficult to maintain and understand. 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 is 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 each controller is 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). *A fundamental reason for applying cascade and decentralized control is thus to save on modelling effort*. 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 be tuned to respond quickly.

Other advantages of cascade and decentralized control include the following: they are often easier to understand by operators, they reduce the need for control links and allow for decentralized implementation, their tuning parameters have a direct and “localized” effect, and they tend to be less sensitive to uncertainty, for example, in the input channels. The issue of simplified implementation and reduced computation load is also important in many applications, but is 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 alternatives 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.

Why do we need a theory for cascade and decentralized control? We just argued that the main advantage of decentralized control was its saving on the modelling effort, but any theoretical treatment of decentralized control requires a plant model. This seems to be a contradiction. However, even though we may *not* want to use a model to tune the controllers, we may still want to use a model to decide on a control structure and to decide on whether acceptable control with a decentralized configuration is possible. The modelling effort in this case is less, because the model may be of a more “generic” nature and does not need to be modified for each particular application.

10.7 Hierarchical and partial control

A hierarchical control system results when we design the subcontrollers in a sequential manner, usually starting with the fast loops (“bottom-up”). This means that the controller at some higher layer in the hierarchy is designed based on a partially controlled plant. In this section we derive transfer functions for partial control, and provide some guidelines for designing hierarchical control systems.

10.7.1 Partial control

Partial control involves controlling only a subset of the outputs for which there is a control objective. We divide the outputs into two classes:

- y_1 – (temporarily) uncontrolled output (for which there is an associated control objective)
- y_2 – (locally) measured and controlled output

We also subdivide the available manipulated inputs in a similar manner:

- u_2 – inputs used for controlling y_2
- u_1 – remaining inputs (which *may* be used for controlling y_1)

We have inserted the word *temporarily* above, since y_1 is normally a controlled output at some higher layer in the hierarchy. However, we here consider the partially controlled system as it appears after having implemented only a local control system where u_2 is used to control y_2. In most of the development that follows we assume that the outputs y_2 are tightly controlled.

Four applications of partial control are:

1. *Sequential design of decentralized controllers.* The outputs y (which include y_1 and y_2) all have an associated control objective, and we use a hierarchical control system. 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.
2. *Sequential design of conventional cascade control.* The outputs y_2 are additional measured (“secondary”) variables which are not important variables in themselves. The reason for controlling y_2 is to improve the control of y_1. The

references r_2 are used as degrees of freedom for controlling y_1 so the set u_1 is often empty.

3. *"True" partial control*. The outputs y (which include y_1 and y_2) all have an associated control objective, and we consider whether by controlling only the subset y_2; we can indirectly achieve acceptable control of y_1, that is, the outputs y_1 remain uncontrolled and the set u_1 remains unused.
4. *Indirect control*. The outputs y_1 have an associated control objective, but they are not measured. Instead, we aim at indirectly controlling y_1 by controlling the "secondary" measured variables y_2 (which have no associated control objective). The references r_2 are used as degrees of freedom and the set u_1 is empty. This is similar to cascade control, but there is no "outer" loop involving y_1. Indirect control was discussed in Section 10.7.4.

The following table shows more clearly the difference between the four applications of partial control. In all cases there is a control objective associated with y_1 and a feedback loop involving measurement and control of y_2.

	Measurement and control of y_1 ?	Control objective for y_2 ?
Sequential decentralized control	Yes	Yes
Sequential cascade control	Yes	No
"True" partial control	No	Yes
Indirect control	No	No

The four problems are closely related, and in all cases we (1) want the effect of the disturbances on y_1 to be small (when y_2 is controlled), and (2) want it to be easy to control y_2 using u_2 (dynamically). . Let us derive the transfer functions for y_1 when y_2 is controlled. One difficulty is that this requires a separate analysis for each choice of y_2 and u_2, and the number of alternatives has a combinatorial growth as illustrated by (10.8).

By partitioning the inputs and outputs, the overall model $y = Gu$ may be written

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

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

Assume now that feedback control $u_2 = K_2(r_2 - y_2 - n_2)$ is used for the "secondary" subsystem involving u_2 and y_2, see Figure 10.7. By eliminating u_2 and y_2, we then get the following model for the resulting partially controlled system:

$$\begin{aligned} y_1 \;=\; & \big(G_{11} - G_{12}K_2(I + G_{22}K_2)^{-1}G_{21}\big)\,u_1 \;+ \\ & \big(G_{d1} - G_{12}K_2(I + G_{22}K_2)^{-1}G_{d2}\big)\,d \;+ \\ & G_{12}K_2(I + G_{22}K_2)^{-1}(r_2 - n_2) \end{aligned} \tag{10.26}$$

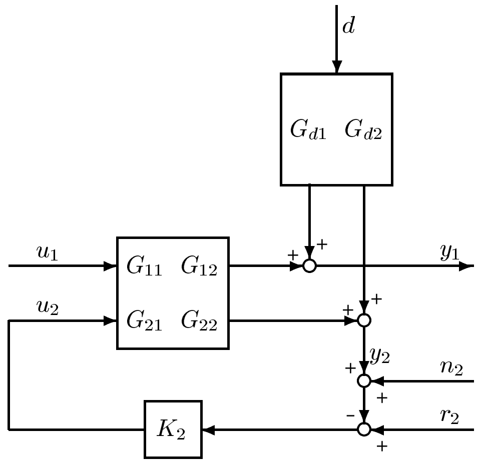


Figure 10.7: Partial control

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

$$F_l(G, -K_2) = G_{11} - G_{12}K_2(I + G_{22}K_2)^{-1}G_{21} \tag{10.27}$$

Tight control of y_2. In some cases we can assume that the control of y_2 is fast compared to the control of y_1. To obtain the model we may formally let $K_2 \to \infty$ in (10.26), but it is better to solve for u_2 in (10.25) to get

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

We have here assumed that G_{22} is square and invertible, otherwise we can get the least-square solution by replacing G_{22}^{-1} by the pseudo-inverse, $G_{22}^{\dagger}$. On substituting this into (10.24) we get

$$y_1 = \underbrace{\big(G_{11} - G_{12}G_{22}^{-1}G_{21}\big)}_{\triangleq P_u} u_1 + \underbrace{\big(G_{d1} - G_{12}G_{22}^{-1}G_{d2}\big)}_{\triangleq P_d} d + \underbrace{G_{12}G_{22}^{-1}}_{\triangleq P_r} \underbrace{(r_2 - e_2)}_{y_2} \tag{10.28}$$

where P_d is called the *partial disturbance gain*, which is the disturbance gain for a system under perfect partial control, and P_u is the effect of u_1 on y_1 with y_2 perfectly controlled. In many cases the set u_1 is empty (there are no extra inputs). The advantage of the model (10.28) over (10.26) is that it is independent of K_2, but we stress that it only applies at frequencies where y_2 is tightly controlled. For the case of tight control we have $e_2 \triangleq y_2 - r_2 = n_2$, i.e., the control error e_2 equals the measurement error (noise) n_2.

Remark. 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.

10.7.2 Hierarchical control and sequential design

A hierarchical control system arises when we apply a sequential design procedure to a cascade or decentralized control system.

The idea is to first implement a local *lower-layer* (or inner) control system for controlling the outputs y_2. Next, with this lower-layer control system in place, we design a controller K_1 to control y_1. The appropriate model for designing K_1 is given by (10.26) (for the general case) or (10.28) (for the case when we can assume y_2 perfectly controlled).

The objectives for this hierarchical decomposition may vary:

1. To allow for simple or even on-line tuning of the lower-layer control system (K_2).
2. To allow the use of longer sampling intervals for the higher layers (K_1).
3. To allow simple models when designing the higher-layer control system (K_1). The high-frequency dynamics of the models of the partially controlled plant (e.g. P_u and P_r) may be simplified if K_1 is mainly effective at lower frequencies.
4. To "stabilize"[1] the plant using a lower-layer control system (K_2) such that it is amenable to manual control.

The latter is the case in many process control applications where we first close a number of faster "regulatory" loops in order to "stabilize" the plant. The higher layer control system (K_1) is then used mainly for optimization purposes, and is not required to operate the plant.

Based on these objectives, Hovd and Skogestad (1993) proposed some criteria for selecting u_2 and y_2 for use in the lower-layer control system:

1. The lower layer must quickly implement the setpoints computed by the higher layers, that is, the input-output controllability of the subsystem involving use of u_2 to control y_2 should be good (consider G_{22} and G_{d2}).
2. The control of y_2 using u_2 should provide local disturbance rejection, that is, it should minimize the effect of disturbances on y_1 (consider P_d for y_2 tightly controlled).
3. The control of y_2 using u_2 should not impose unnecessary control limitations on the remaining control problem which involves using u_1 and/or r_2 to control y_1. By "unnecessary" we mean limitations (RHP-zeros, ill-conditioning, etc.) that did not exist in the original overall problem involving u and y. Consider the controllability of P_u for y_2 tightly controlled, which should not be much worse than that of G.

These three criteria are important for selecting control configurations for distillation columns as is discussed in the next example.

Example 10.5 **Control configurations for distillation columns**. *The overall control problem for the distillation column in Figure 10.8 has 5 inputs*

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

(these are all flows: reflux L, boilup V, distillate D, bottom flow B, overhead vapour V_T) and 5 outputs

$$y = [\,y_D \quad x_B \quad M_D \quad M_B \quad p\,]^T$$

(these are compositions and inventories: top composition y_D, bottom composition x_B, condenser holdup M_D, reboiler holdup M_B, pressure p) see Figure 10.8. This problem usually has no inherent control limitations caused by RHP-zeros, but the plant has poles fin or close to the origin and needs to be stabilized. In addition, for high-purity separations the 5×5 RGA-matrix may have some large elements. Another complication is that composition measurements are often expensive and unreliable.

In most cases, the distillation column is first stabilized by closing three decentralized SISO loops for level and pressure so

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

and the remaining outputs are

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

The three SISO loops for controlling y_2 usually interact weakly and may be tuned independently of each other. However, since each level (tank) has an inlet and two outlet flows, there exists many possible choices for u_2 (and thus for u_1). By convention, each choice ("configuration") 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 we use

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

to control y_1 (and we assume that there is a control system in place which uses $u_2 = [\,D \quad B \quad V_T\,]^T$ to control y_2). The LV-configuration is good from the point of view that control of y_1 using u_1 is nearly independent of the tuning of the controller K_2 involving y_2 and u_2. However, the problem of controlling y_1 by u_1 ("plant" P_u) is often strongly interactive with large steady-state RGA-elements in P_u.

Another configuration is the DV-configuration where

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

and thus $u_2 = [\,L \quad B \quad V_T\,]^T$. In this case, the steady-state interactions from u_1 to y_1 are generally much less, and P_u has small RGA-elements. But the model in (10.26) depends

[1] The terms "stabilize" and "unstable" as used by process operators may not refer to a plant that is unstable in a mathematical sense, but rather to a plant that is *sensitive* to disturbances and which is difficult to control manually.

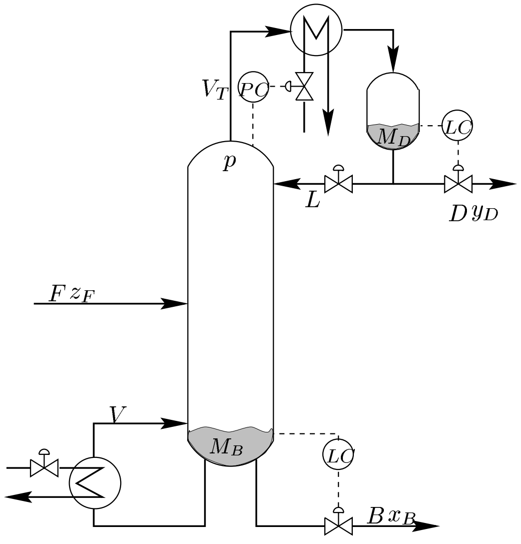


Figure 10.8: Typical distillation column controlled with the LV-configuration

strongly on K_2 (i.e. on the tuning of the level loops), and a slow level loop for M_D may introduce unfavourable dynamics for the response from u_1 to y_1.

There are also many other possible configurations (choices for the two inputs in u_1); with five inputs there are 10 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.

Expressions which directly *relate the models for various configurations, e.g. relationships between P_u^{LV}, P_d^{LV} and P_u^{DV}, P_d^{DV} etc., are given in Haggblom and Waller (1988) and Skogestad and Morari (1987a). However, it may be simpler to start from the overall 5×5 model G, and derive the models for the configurations using (10.26) or (10.28), see also the MATLAB file on page 501.*

To select a good distillation control configuration, one should first consider the problem of controlling levels and pressure (y_2). This eliminates a few alternatives, so the final choice is based on the 2×2 composition control problem (y_1). If y_2 is tightly controlled then none of the configurations seem to yield RHP-zeros in P_u. Important issues to consider then are disturbance sensitivity (the partial disturbance gain P_d should be small) and the interactions (the RGA-elements of P_u). These issues are discussed by, for example, Waller et al. (1988) and Skogestad et al. (1990). Another important issue is that it is often not desirable to have tight level loops and some configurations, like the DV-configuration mentioned above, are sensitive to the tuning of K_2. Then the expressions for P_u and P_d, which are used in the references mentioned above, may not apply. This is further discussed in Skogestad (1997).

Because of the problems of interactions and the high cost of composition measurements, we often find in practice that only one of the two product compositions is controlled ("true" partial control). This is discussed in detail in Example 10.7 below. Another common solution is to make use of additional temperature measurements from the column, where their reference values are set by a composition controller in a cascade manner.

In summary, the overall 5×5 distillation control problem is solved by first designing a 3×3 controller K_2 for levels and pressure, and then designing a 2×2 controller K_1 for the composition control. This is then a case of (block) decentralized control where the controller blocks K_1 and K_2 are designed sequentially (in addition, the blocks K_1 and K_2 may themselves be decentralized).

Sequential design is also used for the design of cascade control systems. This is discussed next.

Sequential design of cascade control systems

Consider the conventional cascade control system in Figure 10.4(a), where we have additional "secondary" measurements y_2 with no associated control objective, and the objective is to improve the control of the primary outputs y_1 by locally controlling y_2. The idea is that this should reduce the effect of disturbances and uncertainty on y_1.

From (10.28), it follows that we should select secondary measurements y_2 (and inputs u_2) such that $\|P_d\|$ is small and at least smaller than $\|G_{d1}\|$. In particular, these arguments apply at higher frequencies. Furthermore, it should be easy to control y_1 by using as degrees of freedom the references r_2 (for the secondary outputs) or the unused inputs u_1. More precisely, we want the input-output controllability of the "plant" $[\,P_u \quad P_r\,]$ (or P_r if the set u_1 is empty) with disturbance model P_d, to be better than that of the plant $[\,G_{11} \quad G_{12}\,]$ (or G_{12}) with disturbance model G_{d1}.

Remark. Most of the arguments given in Section 10.2, for the separation into an optimization and a control layer, and in Section 10.3, for the selection of controlled outputs, apply to cascade control if the term "optimization layer" is replaced by "primary controller", and "control layer" is replaced by "secondary controller".

Exercise 10.6 *The block diagram in Figure 10.5 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 $G_{d2} = [\,0 \quad I\,]$. 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.*

10.7.3 “True” partial control

We here consider the case where we attempt to leave a set of primary outputs y_1 uncontrolled. This “true” partial control may be possible in cases where the outputs are correlated such that controlling the outputs y_2 indirectly gives acceptable control of y_1. One justification for partial control is that measurements, actuators and control links cost money, and we therefore prefer control schemes with as few control loops as possible.

To analyze the feasibility of partial control, consider the effect of disturbances on the *uncontrolled* output(s) y_1 as given by (10.28). Suppose all variables have been scaled as discussed in Section 1.4. Then we have that:

- *A set of outputs y_1 may be left uncontrolled only if the effects of all disturbances on y_1, as expressed by the elements in the corresponding partial disturbance gain matrix P_d, are less than 1 in magnitude at all frequencies.*

Therefore, to evaluate the feasibility of partial control one must for each choice of controlled outputs (y_2) and corresponding inputs (u_2), rearrange the system as in (10.24) and (10.25) and compute P_d using (10.28).

There may also be changes in r_2 (of magnitude R_2) which may be regarded as disturbances on the uncontrolled outputs y_1. From (10.28) then, we also have that:

- *A set of outputs y_1 may be left uncontrolled only if the effects of all reference changes in the controlled outputs (y_2) on y_1, as expressed by the elements in the matrix $G_{12}G_{22}^{-1}R_2$, are less than* 1 *in magnitude at all frequencies.*

One uncontrolled output and one unused input. “True” partial control is often considered if we have an $m \times m$ plant $G(s)$ where acceptable control of all m outputs is difficult, and we consider leaving one input u_j unused and one output y_i uncontrolled. In this case, as an alternative to rearranging y into $\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$ and u into $\begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$ for each candidate control configuration and computing P_d from (10.28), we may directly evaluate the partial disturbance gain based on the overall model $y = Gu + G_d d$. The effect of a disturbance d_k on the uncontrolled output y_i is

$$P_{d_k} = \left(\frac{\partial y_i}{\partial d_k}\right)_{u_j=0, y_{l\neq i}=0} = \frac{[G^{-1}G_d]_{jk}}{[G^{-1}]_{ji}} \tag{10.29}$$

where “$u_j = 0, y_{l\neq i} = 0$” means that input u_j is constant (unused) and the remaining outputs $y_{l\neq i}$ are constant (perfectly controlled).

Proof of (10.29): The proof is from Skogestad and Wolff (1992). Rewrite $y = Gu + [G_d]_k d_k$ as $u = G^{-1}y - [G^{-1}]_k G_d d$. Set $y_l = 0$ for all $l \neq i$. Then $u_j = [G^{-1}]_{ji} y_i - [G^{-1}G_d]_{jk} d_k$ and by setting $u_j = 0$ we find $y_i/d_k = [G^{-1}G_d]_{jk}/[G^{-1}]_{ji}$. □

We want P_{d_k} small so from (10.29) we derive direct insight into how to select the uncontrolled output and unused input:

1. Select the unused input u_j such that the j’th row in $G^{-1}G_d$ has small elements. That is, keep the input constant (unused) if its desired change is small.
2. Select the uncontrolled output y_i and unused input u_j such that the ji’th element in G^{-1} is large. That is, keep an output uncontrolled if it is insensitive to changes in the unused input with the other outputs controlled.

Example 10.6 *Consider the FCC process in Exercise 6.16 on page 250 with*

$$G(0) = \begin{bmatrix} 16.8 & 30.5 & 4.30 \\ -16.7 & 31.0 & -1.41 \\ 1.27 & 54.1 & 5.40 \end{bmatrix}, \quad G^{-1}(0) = \begin{bmatrix} 0.09 & 0.02 & -0.06 \\ 0.03 & 0.03 & -0.02 \\ -0.34 & -0.32 & 0.38 \end{bmatrix}$$

where we want to leave one input unused and one output uncontrolled. From the second rule, since all elements in the third row of G^{-1} are large, it seems reasonable to let input u_3 be unused, as is done in Exercise 6.16. (The outputs are mainly selected to avoid the presence of RHP-zeros, see Exercise 6.16).

(10.29) may be generalized to the case with several uncontrolled outputs / unused inputs (Zhao and Skogestad, 1997). We first reorder G such that the upper left 11-subsystem contains the uncontrolled and unused variables. If G (and thus G_{11}) is square, we then have

$$P_d = \left([G^{-1}]_{11}\right)^{-1} [G^{-1}G_d]_1 \tag{10.30}$$

This result is derived from the definition of P_d in (10.28) by making use of the Schur complement in (A.7).

We next consider a 2×2 distillation process where it is difficult to control both outputs independently due to strong interactions, and we leave one output (y_1) uncontrolled. To improve the performance of y_1 we also consider the use of feedforward control where u_1 is adjusted based on measuring the disturbance (but we need no measurement of y_1).

Example 10.7 Partial and feedforward control of 2×2 distillation process. *Consider a distillation process with* 2 *inputs (reflux L and boilup V),* 2 *outputs (product compositions y_D and x_B) and* 2 *disturbances (feed flowrate F and feed composition z_F). We assume that changes in the reference (r_1 and r_2) are infrequent and they will not be considered. At steady-state ($s = 0$) we have*

$$G = \begin{bmatrix} 87.8 & -86.4 \\ 108.2 & -109.6 \end{bmatrix}, \; G_d = \begin{bmatrix} 7.88 & 8.81 \\ 11.72 & 11.19 \end{bmatrix}, \; G^{-1}G_d = \begin{bmatrix} -0.54 & -0.005 \\ -0.64 & -0.107 \end{bmatrix} \tag{10.31}$$

Since the row elements in $G^{-1}G_d$ are similar in magnitude as are also the elements of G^{-1} (between 0.3 *and* 0.4*), the rules following (10.29) do not clearly favour any particular partial* *control scheme. This is confirmed by the values of P_d, which are seen to be quite similar for the four candidate partial control schemes:*

$$P_{d1}^{2,2} = \begin{bmatrix} -1.36 \\ -0.011 \end{bmatrix}^T,\ P_{d1}^{2,1} = \begin{bmatrix} -1.63 \\ -0.27 \end{bmatrix}^T,\ P_{d2}^{1,2} = \begin{bmatrix} 1.72 \\ 0.014 \end{bmatrix}^T,\ P_{d2}^{1,1} = \begin{bmatrix} 2.00 \\ 0.33 \end{bmatrix}^T$$

The superscripts denote the controlled output and corresponding input. Importantly, in all four cases, the magnitudes of the elements in P_d are much smaller than in G_d, so control of one output significantly reduces the effect of the disturbances on the uncontrolled output. In particular, this is the case for disturbance 2, for which the gain is reduced from about 10 to 0.33 and less.

Let us consider in more detail scheme 1 which has the smallest disturbance sensitivity for the uncontrolled output ($P_{d1}^{2,2}$). This scheme corresponds to controlling output y_2 (the bottom composition) using u_2 (the boilup V) and with y_1 (the top composition) uncontrolled. We use a dynamic model which includes liquid flow dynamics; the model is given in Section 12.4. Frequency-dependent plots of G_d and P_d show that the conclusion at steady state also applies at higher frequencies. This is illustrated in Figure 10.9, where we show for the uncontrolled output y_1 and the worst disturbance d_1 both the open-loop disturbance gain (G_{d11}, Curve 1) and the partial disturbance gain ($P_{d11}^{2,2}$, Curve 2). For disturbance d_2 the partial disturbance gain (not shown) remains below 1 at all frequencies.

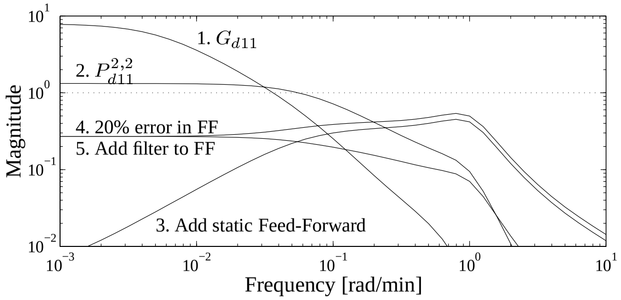


Figure 10.9: Effect of disturbance 1 on output 1 for distillation column example

The partial disturbance gain for disturbance d_1 (the feed flowrate F) is somewhat above 1 at low frequencies ($P_d(0) = -1.36$), so let us next consider how we may reduce its effect on y_1. One approach is to reduce the disturbance itself, for example, by installing a buffer tank (as in pH-example in Chapter 5.16.3). However, a buffer tank has no effect at steady-state, so it does not help in this case.

Another approach is to install a feedforward controller based on measuring d_1 and adjusting u_1 (the reflux L) which is so far unused. In practice, this is easily implemented as a ratio controller which keeps L/F constant. This eliminates the steady-state effect of d_1 on y_1 (provided the other control loop is closed). In terms of our linear model, the mathematical equivalence of this ratio controller is to use $u_1 = 0.54d_1$, where 0.54 is the 1, 1-element in $-G^{-1}G_d$. The effect of the disturbance after including this static feedforward controller is shown as curve 3 in Figure 10.9. However, due to measurement error we cannot achieve perfect feedforward control, so let us assume the error is 20%, and use $u_1 = 1.2 \cdot 0.54d_1$. The steady-state effect of the disturbance is then $P_d(0)(1 - 1.2) = 1.36 \cdot 0.2 = 0.27$, which is still acceptable. But, as seen from the frequency-dependent plot (curve 4), the effect is above 0.5 at higher frequencies, which may not be desirable. The reason for this undesirable peak is that the feedforward controller, which is purely static, reacts too fast, and in fact makes the response worse at higher frequencies (as seen when comparing curves 3 and 4 with curve 2). To avoid this we filter the feedforward action with a time constant of 3 min resulting in the following feedforward controller:

$$u_1 = \frac{0.54}{3s+1} d_1 \tag{10.32}$$

To be realistic we again assume an error of 20%. The resulting effect of the disturbance on the uncontrolled output is shown by curve 5, and we see that the effect is now less than 0.27 at all frequencies, so the performance is acceptable.

Remark. *In the example there are four alternative partial control schemes with quite similar disturbance sensitivity for the uncontrolled output. To decide on the best scheme, we should also perform a controllability analysis of the feedback properties of the four 1 × 1 problems. Performing such an analysis, we find that schemes 1 (the one chosen) and 4 are preferable, because the input in these two cases has a more direct effect on the output, and with less phase lag.*

In conclusion, for this example it is difficult to control both outputs simultaneously using feedback control due to strong interactions. However, we can almost achieve acceptable control of both outputs by leaving y_1 uncontrolled. The effect of the most difficult disturbance on y_1 can be further reduced using a simple feedforward controller (10.32) from disturbance d_1 to u_1.

10.7.4 Measurement selection for indirect control

Assume the overall goal is to keep some variable y_1 at a given value (setpoint) r_1, e.g. our objective is to minimize $J = \|y_1 - r_1\|$. We assume we cannot measure y_1, and instead we attempt to achieve our goal by controlling y_2 at a constant value r_2. For small changes we may assume linearity and write

$$y_1 = G_1 u + G_{d1} d \tag{10.33}$$

$$y_2 = G_2 u + G_{d2} d \tag{10.34}$$

With feedback control of y_2 we get $y_2 = r_2 + e_2$ where e_2 is the control error. We now follow the derivation that led to P_d in (10.28): Solving for u_2 in (10.34) and substituting into (10.33) yields

$$y_1 = (G_{d1} - G_1 G_2^{-1} G_{d2})d + G_1 G_2^{-1}(r_2 + e_2)$$

With $e_2 = 0$ and $d = 0$ this gives $y_1 = G_1 G_2^{-1} r_2$, so r_2 must be chosen such that

$$r_1 = G_1 G_2^{-1} r_2 \tag{10.35}$$

he control error in the primary output is then

$$y_1 - r_1 = \underbrace{(G_{d1} - G_1 G_2^{-1} G_{d2})}_{P_d} d + \underbrace{G_1 G_2^{-1}}_{P_r} e_2 \tag{10.36}$$

To minimize $J = \|y_1 - r_1\|$ we should therefore select controlled outputs such that $\|P_d d\|$ and $\|P_r e_2\|$ are small. Note that P_d depends on the scaling of disturbances d and "primary" outputs y_1 (and is independent of the scaling of inputs u and selected outputs y_2, at least for square plants). The magnitude of the control error e_2 depends on the choice of outputs y_2. Based on (10.36) a procedure for selecting controlled outputs y_2 may be suggested:

> Scale the disturbances d to be of magnitude 1 (as usual), and scale the outputs y_2 so that the expected control error e_2 (measurement noise) is of magnitude 1 for each output (this is different from the output scaling used in other cases). Then to minimize the control error for the primary outputs, $J = \|y_1 - r_1\|$, we should select sets of controlled outputs which:
>
> $$\text{Minimizes } \|[\,P_d \quad P_r\,]\| \tag{10.37}$$

Remark 1 The choice of norm in (10.37) is usually of secondary importance. The maximum singular value arises if $\|d\|_2 \leq 1$ and $\|e_2\|_2 \leq 1$, and we want to minimize $\|y_1 - r_1\|_2$.

Remark 2 For the choice $y_2 = y_1$ we have that $r_1 = r_2$ is independent of d and the matrix P_d in (10.36) and (10.37) is zero. However, P_r is still non-zero.

Remark 3 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 too close to the column end due to sensitivity to measurement error ($\|P_r\|$ becomes large), and we cannot place it too far from the column end due to sensitivity to disturbances ($\|P_d\|$ becomes large).

Remark 4 Indirect control is related to the idea of *inferential control* which is commonly used in the process industry. However, in 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.

Remark 5 The problem of indirect control is closely related to that of *cascade control* discussed in Section 10.7.2. The main difference is that in cascade control we also measure and control y_1 in an outer loop. In this case we want $\|[\,P_d \quad P_r\,]\|$ small only at high frequencies beyond the bandwidth of the outer loop involving y_1.

10.8 Decentralized feedback control

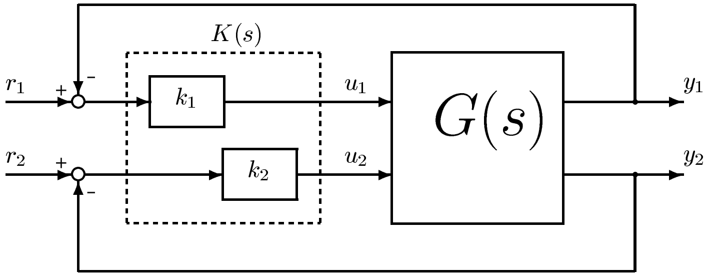


Figure 10.10: Decentralized diagonal control of a 2 × 2 plant

In this section, $G(s)$ is a square plant which is to be controlled using a diagonal controller (see Figure 10.10)

$$K(s) = \mathrm{diag}\{k_i(s)\} = \begin{bmatrix} k_1(s) & & & \\ & k_2(s) & & \\ & & \ddots & \\ & & & k_m(s) \end{bmatrix} \tag{10.38}$$

This is the problem of decentralized diagonal feedback control. The design of decentralized control systems involves two steps:

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

The optimal solution to this problem is very difficult mathematically, because the optimal controller is in general of infinite order and may be non-unique; we do not address it in this book. The reader is referred to the literature (e.g. Sourlas and Manousiouthakis, 1995) for more details. Rather we aim at providing simple tools for pairing selections (step 1) and for analyzing the achievable performance (controllability) of diagonally controlled plants (which may assist in step 2).

Notation for decentralized diagonal control. $G(s)$ denotes a square $m \times m$ plant with elements g_{ij}. $G^{ij}(s)$ denotes the remaining $(m-1) \times (m-1)$ plant obtained by removing row i and column j in $G(s)$. With a particular choice of pairing 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 ($\mathrm{diag}\{k_i\}$), and we also introduce

$$\widetilde{G} \triangleq \mathrm{diag}\{g_{ii}\} = \begin{bmatrix} g_{11} & & & \\ & g_{22} & & \\ & & \ddots & \\ & & & g_{mm} \end{bmatrix} \tag{10.39}$$

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 $L = GK$.

10.8.1 RGA as interaction measure for decentralized control

We here follow Bristol (1966), and show that the RGA provides a measure of the interactions caused by decentralized diagonal control. Let u_j and y_i denote a particular input and output 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:

- Other loops open: All other inputs are constant, i.e. $u_k = 0, \forall k \neq j$.
- Other loops closed: All other outputs are constant, i.e. $y_k = 0, \forall k \neq i$.

In the latter case, it is assumed that the other loops are closed with perfect control. 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 the effect $\partial y_i / \partial u_j$ of "our" given input u_j on "our" given output y_i for the two extreme cases. We get

$$\text{Other loops open:} \quad \left(\frac{\partial y_i}{\partial u_j}\right)_{u_k=0, k\neq j} = g_{ij} \tag{10.40}$$

$$\text{Other loops closed:} \quad \left(\frac{\partial y_i}{\partial u_j}\right)_{y_k=0, k\neq i} \triangleq \widehat{g}_{ij} \tag{10.41}$$

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}

$$\widehat{g}_{ij} = 1/[G^{-1}]_{ji} \tag{10.42}$$

To derive (10.42) note that

$$y = Gu \quad \Rightarrow \quad \left(\frac{\partial y_i}{\partial u_j}\right)_{u_k=0, k\neq j} = [G]_{ij} \tag{10.43}$$

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

$$u = G^{-1}y \quad \Rightarrow \quad \left(\frac{\partial u_j}{\partial y_i}\right)_{y_k=0, k\neq i} = [G^{-1}]_{ji} \tag{10.44}$$

and (10.42) follows. Bristol argued that the ratio between the gains in (10.40) and (10.41), corresponding to the two extreme cases, is a useful measure of interactions, and he introduced the term, ij'th relative gain defined as

$$\lambda_{ij} \triangleq \frac{g_{ij}}{\widehat{g}_{ij}} = [G]_{ij}[G^{-1}]_{ji} \tag{10.45}$$

The Relative Gain Array (RGA) is the corresponding matrix of relative gains. From (10.45) we get $\Lambda(G) = G \times (G^{-1})^T$ where × denotes element-by-element multiplication (the Schur product). This is identical to our definition of the RGA-matrix in (3.69).

Intuitively, we would like to pair variables u_j and y_i so that λ_{ij} is close to 1, because this means that the gain from u_j to y_i is unaffected by closing the other loops. More precicely, we would like to pair such that the rearranged system, with the pairings along the diagonal, has a RGA matrix close to identity (see Pairing Rule 1, page 445).

10.8.2 Factorization of sensitivity function

The magnitude of the off-diagonal elements in G (the interactions) relative to its diagonal elements are given by the matrix

$$E \triangleq (G - \widetilde{G})\widetilde{G}^{-1} \tag{10.46}$$

An important relationship for decentralized control is given by the following factorization of the return difference operator:

$$\underbrace{(I + GK)}_{\text{overall}} = \underbrace{(I + E\widetilde{T})}_{\text{interactions}} \quad \underbrace{(I + \widetilde{G}K)}_{\text{individual loops}} \tag{10.47}$$

or equivalently in terms of the sensitivity function $S = (I + GK)^{-1}$,

$$\boxed{S = \widetilde{S}(I + E\widetilde{T})^{-1}} \tag{10.48}$$

Here

$$\widetilde{S} \triangleq (I + \widetilde{G}K)^{-1} = \mathrm{diag}\{\frac{1}{1 + g_{ii}k_i}\} \quad \text{and} \quad \widetilde{T} = I - \widetilde{S} \tag{10.49}$$

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. (10.48) follows from (A.139) with $G = \widetilde{G}$ and $G' = G$. The reader is encouraged to confirm that (10.48) is correct, because most of the important results for stability and performance using decentralized control may be derived from this expression.

10.8.3 Stability of decentralized control systems

Consider a square plant with single-loop controllers. For a 2 × 2 plant there are two alternative pairings, a 3 × 3 plant offers 6, a 4 × 4 plant 24, and an $m \times m$ plant has $m!$ alternatives. Thus, tools are needed which are capable of quickly evaluating alternative pairings. In this section we first derive sufficient conditions for stability which may be used to select promising pairings. These give rules in terms of diagonal dominance. We then derive necessary conditions for stability which may be used to *eliminate* undesirable pairings.

A. Sufficient conditions for stability

For decentralized diagonal control, it is desirable that the system can be tuned and operated one loop at a time. Assume therefore that G is stable and each individual loop is stable by itself ($\widetilde{S}$ and $\widetilde{T}$ are stable). Then from the factorization $S = \widetilde{S}(I + E\widetilde{T})^{-1}$ in (10.47) and the generalized Nyquist theorem in Lemma A.5 (page 540), it follows that the overall system is stable (S is stable) if and only if $\det(I + E\widetilde{T}(s))$ does not encircle the origin as s traverses the Nyquist D-contour. From the spectral radius stability condition in (4.107) we then have that the overall system is stable if

$$\rho(E\widetilde{T}(j\omega)) < 1, \forall\omega \tag{10.50}$$

This sufficient condition for overall stability can, as discussed by Grosdidier and Morari (1986), be used to obtain a number of even *weaker* stability conditions.

Sufficient conditions in terms of E. The least conservative approach is to split up $\rho(E\widetilde{T})$ using the structured singular value. From (8.92) we have $\rho(E\widetilde{T}) \leq \mu(E)\bar{\sigma}(\widetilde{T})$ and from (10.50) we get the following theorem (as first derived by Grosdidier and Morari, 1986):

Theorem 10.2 *Assume G is stable and that the individual loops are stable ($\widetilde{T}$ is stable). Then the entire system is closed-loop stable (T is stable) if*

$$\bar{\sigma}(\widetilde{T}) = \max_i |\widetilde{t}_i| < 1/\mu(E) \quad \forall\omega \tag{10.51}$$

Here $\mu(E)$ is called the structured singular value interaction measure, and is computed with respect to the *diagonal structure* of $\widetilde{T}$, where we may view $\widetilde{T}$ as the "design uncertainty". We would like to use integral action in the loops, that is, we want $\widetilde{T} \approx I$ at low frequencies, i.e. $\bar{\sigma}(\widetilde{T}) \approx 1$. Thus, in order to satisfy (10.51) we need $\mu(E) \leq 1$ at low frequencies where we have tight control. This gives the following **rule**:

Prefer pairings for which we have $\mu(E) < 1$ ("generalized diagonal dominance") at low frequencies.

Another approach is to use Gershgorin's theorem, see page 514. From (10.50) we then derive the following sufficient condition for overall stability in terms of the rows of G:

$$|\widetilde{t}_i| < |g_{ii}| / \sum_{j \neq i} |g_{ij}| \quad \forall i, \forall\omega \tag{10.52}$$

or alternatively, in terms of the columns,

$$|\widetilde{t}_i| < |g_{ii}| / \sum_{j \neq i} |g_{ji}| \quad \forall i, \forall\omega \tag{10.53}$$

This gives the important insight that we prefer to pair on large elements in G.

Remark 1 We cannot say that (10.51) is always less conservative than (10.52) and (10.53). It is true that the *smallest* of the $i = 1, \ldots m$ upper bounds in (10.52) or (10.53) is always smaller (more restrictive) than $1/\mu(E)$ in (10.51). However, (10.51) imposes the *same* bound on $|\widetilde{t}_i|$ for each loop, whereas (10.52) and (10.53) give *individual* bounds, some of which may be less restrictive than $1/\mu(E)$.

Remark 2 Another definition of generalized diagonal dominance is that $\rho(|E|) < 1$, where $\rho(|E|)$ is the Perron root; see (A.127). However, since $\mu(E) = \mu(DED^{-1})$, see (8.84) where D in this case is diagonal, it follows from (A.127) that $\mu(E) \leq \rho(|E|)$, and it is better (less restrictive) to use $\mu(E)$ to define diagonal dominance.

Remark 3 Condition (10.51) and the use of $\mu(E)$ 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).

Sufficient conditions for stability in terms of RGA. We now want to show that for closed-loop stability it is desirable to select pairings such that the RGA is close to the identity matrix in the crossover region. The next simple theorem, which applies to a triangular plant, will enable us to do this:

Theorem 10.3 *Suppose the plant $G(s)$ is stable. If the RGA-matrix $\Lambda(G) = I\ \forall\omega$ then stability of each of the individual loops implies stability of the entire system.*

Proof: From the definition of the RGA it follows that $\Lambda(G) = I$ can only arise from a triangular $G(s)$ or from $G(s)$-matrices that can be made triangular by interchanging rows and columns in such a way that the diagonal elements remain the same but in a different order (the pairings remain the same). A plant with a "triangularized" transfer matrix (as described above) controlled by a diagonal controller has only *one-way coupling* and will always yield a stable system provided the individual loops are stable. Mathematically, $E = (G - \widetilde{G})\widetilde{G}^{-1}$ can be made triangular, and since the diagonal elements of E are zero, it follows that all eigenvalues of $E\widetilde{T}$ are zero, so $\rho(E\widetilde{T}) = 0$ and (10.50) is satisfied. □

RGA at crossover frequencies. In most cases, it is sufficient for overall stability to require that $G(j\omega)$ is close to triangular (or $\Lambda(G) \approx I$) at crossover frequencies:

Pairing Rule 1. *To achieve stability with decentralized control prefer pairings such that at frequencies ω around crossover, the rearranged matrix $G(j\omega)$ (with the paired elements along the diaginal) is close to triangular. This is equivalent to requiring $\Lambda(G(j\omega)) \approx I$, i.e. the RGA-number $\|\Lambda(G(j\omega)) - I\|_{\text{sum}}$ should be small.*

Derivation of Pairing rule 1. Assume that $\widetilde{S}$ is stable, and that $\widetilde{S}\Gamma(s) = \widetilde{S}\widetilde{G}(s)G(s)^{-1}$ is stable and has no RHP-zeros (which is always satisfied if both G and $\widetilde{G}$ are stable and have no RHP-zeros). Then from (10.60) the overall system is stable (S is stable) if and only if $(I + \widetilde{S}(\Gamma - I))^{-1}$ is stable. Here $\widetilde{S}(\Gamma - I)$ is stable, so from the spectral radius stability condition in (4.107) the overall system is stable if

$$\rho(\widetilde{S}(\Gamma - I)(j\omega)) < 1, \quad \forall\omega \tag{10.54}$$

At low frequencies, this condition is usually satisfied because $\widetilde{S}$ is small. At higher frequencies, where the elements in $\widetilde{S} = \mathrm{diag}\{\widetilde{s}_i\}$ approach and possibly exceed 1 in magnitude, (10.54) may be satisfied if $G(j\omega)$ is close to triangular. This is because $\Gamma - I$ and thus $\widetilde{S}(\Gamma - I)$ are then close to triangular, with diagonal elements close to zero, so the eigenvalues of $\widetilde{S}(\Gamma - I)(j\omega)$ are close to zero, (10.54) is satisfied and we have stability of S. This conclusion also holds for plants with RHP-zeros provided they are located beyond the crossover frequency range. □

Example. *Consider a plant and its RGA-matrix*

$$G = \begin{bmatrix} -5 & 1 \\ 6 & 2 \end{bmatrix}; \quad \Lambda(G) = \begin{bmatrix} 0.625 & 0.375 \\ 0.375 & 0.625 \end{bmatrix}$$

The RGA indicates that G is diagonally dominant and that we would prefer to use the diagonal pairing. This is confirmed by computing the relative interactions as given by the matrix E:

$$\widetilde{G} = \begin{bmatrix} -5 & 0 \\ 0 & 2 \end{bmatrix}; \quad E = (G - \widetilde{G})\widetilde{G}^{-1} = \begin{bmatrix} 0.375 & 0.3125 \\ -0.75 & 0.375 \end{bmatrix}$$

. The SSV-interaction measure is $\mu(E) = 0.6124$, so the plant is diagonally dominant, and from (10.51) stability of the individual loops will guarantee stability of the overall closed-loop system. Note that the Perron root $\rho(|E|) = 0.8591$ which shows that the use of $\mu(E)$ is less conservative.

It is not possible in this case to conclude from the Gershgorin bounds in (10.52) and (10.53) that the plant is diagonally dominant, because the off-diagonal element of 6 is larger than any of the diagonal elements.

B. Necessary steady-state conditions for stability

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

An even stronger requirement is that the system remains stable as the gain in various loops are reduced (detuned) by an arbitrary factor, i.e. $0 \le \epsilon_i \le 1$ ("complete detunability"). Decentralized integral controllability (DIC) is concerned with whether this is *possible* integral control:

Definition 10.1 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 ($0 \le \epsilon_i \le 1$) 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 first proved by Grosdidier et al. (1985):

Theorem 10.4 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:

$$\text{A stable (reordered) plant } G(s) \text{ is DIC only if } \lambda_{ii}(0) \ge 0 \text{ for all } i. \tag{10.55}$$

Proof: The theorem may be proved by setting $\widetilde{T} = I$ in (10.47) and applying the generalized Nyquist stability condition. Alternatively, we can use Theorem 6.5 on page 245 and select $G' = \mathrm{diag}\{g_{ii}, G^{ii}\}$. Since $\det G' = g_{ii} \det G^{ii}$ and from (A.77) $\lambda_{ii} = \frac{g_{ii} \det G^{ii}}{\det G}$ we have $\det G'/detG = \lambda_{ii}$ and Theorem 10.4 follows. □

Each of the three possible instabilities in Theorem 10.4 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, for example, 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.

Remarks on DIC and RGA.

1. DIC was introduced by Skogestad and Morari (1988b). A detailed survey of conditions for DIC and other related properties is given by Campo and Morari (1994).
2. Unstable plants are not DIC. The reason 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.

3. For $\epsilon_i = 0$ we assume that the integrator of the corresponding SISO controller has been removed, otherwise the integrator would yield internal instability.
4. For 2×2 and 3×3 plants we have even tighter conditions for DIC than (10.55). For 2×2 plants (Skogestad and Morari, 1988b)

$$\text{DIC} \quad \Leftrightarrow \quad \lambda_{11}(0) > 0 \tag{10.56}$$

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)

$$\text{DIC} \quad \Leftrightarrow \quad \sqrt{\lambda_{11}(0)} + \sqrt{\lambda_{22}(0)} + \sqrt{\lambda_{33}(0)} \geq 1 \tag{10.57}$$

(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 identical to 1, but this has little practical significance).
5. One cannot expect tight conditions for DIC in terms of the RGA for 4×4 systems or higher. The reason is that the RGA essentially only considers "corner values", $\epsilon_i = 0$ or $\epsilon_i = 1$ (integrity), for the detuning factor in each loop in the definition of DIC. 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 $\epsilon_k = 1$.
6. **Determinant conditions 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 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 DIC.*

   This determinant condition follows by applying Theorem 6.5 to all possible combinations of $\epsilon_i = 0$ or 1 as illustrated in the proof of Theorem 10.4, and is equivalent to requiring that the so-called Niederlinski indices,

$$N_I = \det G(0) / \Pi_i g_{ii}(0) \tag{10.58}$$

of $G(0)$ and its principal submatrices are all positive. Actually, this yields more information than the RGA, because in the RGA the terms are combined into $\lambda_{ii} = \frac{g_{ii} \det G^{ii}}{\det G}$ 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.
7. 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.
8. If we assume that the controllers have integral action, then $T(0) = I$, and we can derive from (10.51) that a sufficient condition for DIC is that G is generalized diagonally dominant at steady-state, that is,

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

This is proved by Braatz (1993, p.154). However, the requirement is only sufficient for DIC and therefore cannot be used to eliminate designs. Specifically, for a 2×2 system

it is easy to show (Grosdidier and Morari, 1986) that $\mu(E(0)) < 1$ is equivalent to $\lambda_{11}(0) > 0.5$, which is conservative when compared with the necessary and sufficient condition $\lambda_{11}(0) > 0$ in (10.56).
9. 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.
10. Since Theorem 6.5 applies to unstable plants, we may also easily extend Theorem 10.4 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 (1994a). Alternatively, one may first implement a stabilizing controller and then analyze the partially controlled system as if it were the plant $G(s)$.
11. The above results only address stability. Performance is analyzed in Section 10.8.5.

10.8.4 The RGA and right-half plane zeros: Further reasons for not pairing on negative RGA elements

Bristol (1966) claimed that negative values of $\lambda_{ii}(0)$ implied the presence of RHP-zeros. This is indeed true as illustrated by the following two theorems:

Theorem 10.5 *(Hovd and Skogestad, 1992) Consider a transfer function matrix $G(s)$ with no zeros or poles at $s = 0$. Assume $\lim_{s \to \infty} \lambda_{ij}(s)$ 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 10.6 *(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 $\widehat{g}_{ij}(s)$ has an odd number of RHP-poles and RHP-zeros.

Negative RGA-elements and decentralized performance With decentralized control we usually design and implement the controller by tuning and closing one loop at a time in a sequential manner. Assume that we pair on a *negative* steady-state RGA-element, $\lambda_{ij}(0) < 0$, assume that $\lambda_{ij}(\infty)$ is positive (it is usually close to 1, see Pairing rule 1), and assume that the element g_{ij} has no RHP-zero. Then taken together the above two theorems then have the following implications:

(a) If we start by closing this loop (involving input u_i and output y_j), then we will get a RHP-zero (in $G^{ij}(s)$) which will limit the performance in the other outputs (follows from Theorem 10.5 by assuming that G has no RHP-zero and that $\lambda_{ij}(\infty) > 0$).

(b) If we end by closing this loop, then we will get a RHP-zero i(in $\widehat{g}_{ij}(s)$) which will limit the performance in output y_i (follows from Theorem 10.6).

In conclusion, pairing on a negative RGA-element will, in addition to resulting in potential instability as given in Theorem 10.4, also limit the closed-loop decentralized performance.

We have then firmly established the following rule:

Pairing Rule 2. *For a stable plant avoid pairings that correspond to negative steady-state RGA-elements, $\lambda_{ij}(0) < 0$.*

The RGA is a very efficient 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 alternative pairings by a simple glance at the RGA-matrix. This is illustrated by the following example.

Example 10.8 *Consider a 3×3 plant with*

$$G(0) = \begin{bmatrix} 10.2 & 5.6 & 1.4 \\ 15.5 & -8.4 & -0.7 \\ 18.1 & 0.4 & 1.8 \end{bmatrix}, \quad \Lambda(0) = \begin{bmatrix} 0.96 & \mathbf{1.45} & -1.41 \\ \mathbf{0.94} & -0.37 & 0.43 \\ -0.90 & -0.07 & \mathbf{1.98} \end{bmatrix} \tag{10.59}$$

For a 3×3 plant there are 6 alternative 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 DIC we can from a quick glance at the steady-state RGA eliminate five of the six alternative pairings.

Example 10.9 *Consider the plant and RGA*

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

Note that the RGA is constant, independent of frequency. Only two of the six possible pairings give positive steady-state RGA-elements (Pairing Rule 2): (a) The (diagonal) pairing on all $\lambda_{ii} = 1$. (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, $\|\Lambda - I\|_{\mathrm{sum}}$, is 30 for both alternatives. Thus, none of the two alternatives satisfy Pairing Rule 1, and we are led to conclude that decentralized control should not be used for this plant.

Remark. *(Hovd and Skogestad, 1992) confirm this conclusion by designing PI controllers for the two cases. Surprisingly, they found pairing (a) corresponding to $\lambda_{ii} = 1$ to be significantly worse than (b) with $\lambda_{ii} = 5$. They found the achieveable 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 10.7 *(a) Assume that the 4×4 matrix in (A.82) 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 6.16 on page 250. Show that 5 of the 6 possible pairings can be eliminated by requiring DIC.*

10.8.5 Performance of decentralized control systems

Above we used the factorization $S = \widetilde{S}(I + E\widetilde{T})^{-1}$ in (10.48) to study stability. Here we want to consider perfomance. A related factorization which follows from (A.140) is

$$S = (I + \widetilde{S}(\Gamma - I))^{-1}\widetilde{S}\Gamma \tag{10.60}$$

where Γ is the Performance Relative Gain Array (PRGA),

$$\Gamma(s) \triangleq \widetilde{G}(s)G^{-1}(s) \tag{10.61}$$

which is a scaled inverse of the plant. Note that $E = \Gamma^{-1} - I$. At frequencies where feedback is effective ($\widetilde{S} \approx 0$), (10.60) yields $S \approx \widetilde{S}\Gamma$ which shows that Γ is important when evaluating performance with decentralized control. 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 measures also one-way interaction, whereas the RGA only measures two-way interaction.

We will also make use of the related Closed-Loop Disturbance Gain (CLDG) matrix, defined as

$$\widetilde{G}_d(s) \triangleq \Gamma(s)G_d(s) = \widetilde{G}(s)G^{-1}(s)G_d(s) \tag{10.62}$$

The CLDG depends on both output and disturbance scaling.

In the following, we consider performance in terms of the control error

$$e = y - r = Gu + G_d d - r \tag{10.63}$$

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, $|e_i| < 1$.

For SISO systems, we found in Section 5.10 that in terms of scaled variables we must at all frequencies require

$$|1+L| > |G_d| \quad \text{and} \quad |1+L| > |R| \tag{10.64}$$

for acceptable disturbance rejection and command tracking, respectively. Note that L, G_d and R are all scalars in this case. For decentralized control these requirements may be directly generalized by introducing the PRGA-matrix, $\Gamma = \widetilde{G}G^{-1}$, and the CLDG-matrix, $\widetilde{G}_d = \Gamma G_d$. These generalizations will be presented and discussed next, and then subsequently proved.

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 (10.67) is valid). Then for acceptable disturbance rejection ($|e_i| < 1$) we must with decentralized control require for each loop i,

$$|1+L_i| > |\widetilde{g}_{di}| \quad \forall i \tag{10.65}$$

which is the same as the SISO-condition (5.52) 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. Similarly, consider a change in reference for output j of magnitude R_j. Consider frequencies where feedback is effective (and (10.67) is valid). Then for acceptable reference tracking ($|e_i| < 1$) we must require for each loop i

$$|1+L_i| > |\gamma_{ij}| \cdot |R_j| \quad \forall i \tag{10.66}$$

which is the same as the SISO-condition 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 γ_{ii} close to 1 (recall Pairing Rule 1 on page 445).

Proofs of (10.65) and (10.66): At frequencies where feedback is effective, $\widetilde{S}$ is small, so

$$I + \widetilde{S}(\Gamma - I) \approx I \tag{10.67}$$

and from (10.60) we have

$$S \approx \widetilde{S}\Gamma \tag{10.68}$$

The closed-loop response then becomes

$$e = SG_d d - Sr \approx \widetilde{S}\widetilde{G}_d d - \widetilde{S}\Gamma r \tag{10.69}$$

and the response in output i to a single disturbance d_k and a single reference change r_j is

$$e_i \approx \widetilde{s}_i \widetilde{g}_{dik} d_k - \widetilde{s}_i \gamma_{ik} r_k \tag{10.70}$$

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 (10.65) follows. Similarly, to achieve $|e_i| < 1$ for $|r_j| = |R_j|$ we must require $|s_i \gamma_{ik} R_j| < 1$ and (10.66) follows. Also note that $|s_i \gamma_{ik}| < 1$ will imply that assumption (10.67) is valid. Since R usually has all of its elements larger than 1, in most cases (10.67) will be automatically satisfied if (10.66) is satisfied, so we normally need not check assumption (10.67). □

Remark 1 (10.68) may also be derived from (10.48) 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)):

$$\beta_i \triangleq \widetilde{g}_{di}/g_{di} = [\widetilde{G}G^{-1}g_d]_i/[g_d]_i \tag{10.71}$$

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.

10.8.6 Summary: 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. If the plant is unstable, then as usual the unstable modes must be controllable and observable. In addition, the unstable modes must not be *decentralized fixed modes*, otherwise the plant cannot be stabilized with a diagonal controller (Lunze, 1992). For example, this is the case for a triangular plant if the unstable mode appears only in the off-diagonal elements.

The next step is to compute the RGA-matrix as a function of frequency, and to determine if one can find a good set of input-output pairs bearing in mind the following:

1. Prefer pairings which have the RGA-matrix close to identity at frequencies around crossover, i.e. the RGA-number $\|\Lambda(j\omega) - I\|$ should be small. This rule is to ensure that interactions from other loops do not cause instability as discussed following (10.54).
2. Avoid a pairing ij with negative steady-state RGA elements, $\lambda_{ij}(G(0))$.
3. Prefer a pairing ij where g_{ij} puts minimal restrictions on the achievable bandwidth. Specifically, the frequency ω_{uij} where $\angle g_{ij}(j\omega_{uij}) = -180°$ should be as large as possible.

This rule favours pairing on variables “close to each other”, which makes it easier to satisfy (10.65) and (10.66) physically while at the same time achieving stability. It is also consistent with the desire that $\Lambda(j\omega)$ is close to I.

When a reasonable choice of pairings has been made, one should rearrange G to have the paired elements along the diagonal and perform a controllability analysis.

4. Compute the CLDG and PRGA, and plot these as a function of frequency.
5. For systems with many loops, it is best to perform the analysis one loop at the 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, we need $|1+L_i|$ to be larger than each of these

$$\text{Performance}: \quad |1+L_i| > \max_{k,j}\{|\widetilde{g}_{dik}|, |\gamma_{ij}|\} \tag{10.72}$$

   To achieve stability of the individual loops one must analyze $g_{ii}(s)$ to ensure that the bandwidth required by (10.72) 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 frequency where $\angle g_{ii}$ is $-180°$ (recall Section 5.12).
6. As already mentioned one may check for constraints by considering the elements of $G^{-1}G_d$ and making sure that they do not exceed one in magnitude within the frequency range where control is needed. Equivalently, one may for each loop i plot $|g_{ii}|$, and the requirement is then that

$$\text{To avoid input constraints}: \quad |g_{ii}| > |\widetilde{g}_{dik}|, \quad \forall k \tag{10.73}$$

   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 (10.73) 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, then one may consider another choice of pairings and go back to Step 4. If one still cannot find any pairings which are controllable, then one should consider multivariable control.

7. If the chosen pairing *is* controllable then the analysis based on (10.72) tells us directly how large $|L_i| = |g_{ii}k_i|$ must be, and can be used as a basis for designing the controller $k_i(s)$ for loop i.

Remark. In some cases, pairings which violate the above rules may be chosen. For example, one may even choose to pair on elements with $g_{ii} = 0$ which yield $\lambda_{ii} = 0$. One then relies on the interactions to achieve the desired performance as loop i by itself has no effect. An example of this is in distillation control when the LV-configuration is *not* used, see Example 10.5.

Example 10.10 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 consider again the distillation process used in Example 10.7. The LV configuration is used, that is, the manipulated inputs are reflux L (u_1) and boilup V (u_2). Outputs are the product compositions y_D (y_1) and x_B (y_2). Disturbances in feed flowrate 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 5 state dynamic model is given in Section 12.4.*

Initial controllability analysis. *$G(s)$ is stable and has no RHP-zeros. The plant and RGA-matrix at steady-state are*

$$G(0) = \begin{bmatrix} 87.8 & -86.4 \\ 108.2 & -109.6 \end{bmatrix}, \quad \Lambda(0) = \begin{bmatrix} 35.1 & -34.1 \\ -34.1 & 35.1 \end{bmatrix} \tag{10.74}$$

The RGA-elements are much larger than 1 and indicate a plant that is fundamentally difficult to control. 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.

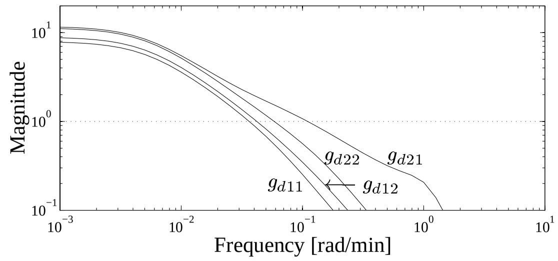


Figure 10.11: Disturbance gains $|g_{dik}|$, for effect of disturbance k on output i

The steady-state effect of the two disturbances is given by

$$G_d(0) = \begin{bmatrix} 7.88 & 8.81 \\ 11.72 & 11.19 \end{bmatrix} \tag{10.75}$$

and the magnitudes of the elements in $G_d(j\omega)$ are plotted as a function of frequency in Figure 10.11. 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.

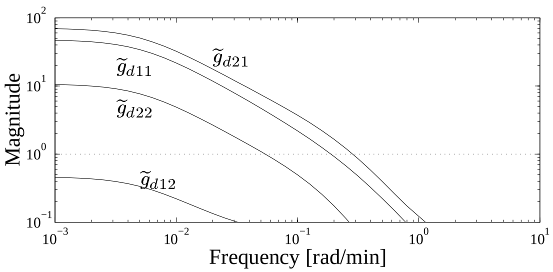


Figure 10.12: Closed-loop disturbance gains, $|\widetilde{g}_{dik}|$, for effect of disturbance k on output i

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 10.12 and 10.13. At steady-state we have*

$$\Gamma(0) = \begin{bmatrix} 35.1 & -27.6 \\ -43.2 & 35.1 \end{bmatrix}, \quad \widetilde{G}_d(0) = \Gamma(0)G_d(0) = \begin{bmatrix} -47.7 & -0.40 \\ 70.5 & 11.7 \end{bmatrix} \tag{10.76}$$

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.

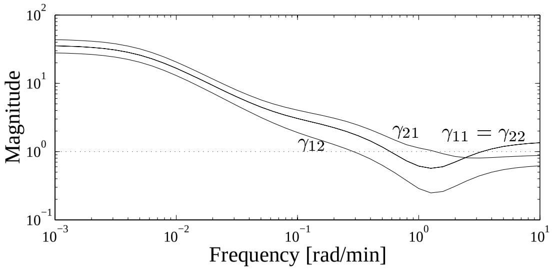


Figure 10.13: PRGA-elements $|\gamma_{ij}|$ for effect of reference j on output i

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 $|1 + L_1| > |\widehat{g}_{d11}|$ at frequencies where $|\widehat{g}_{d11}|$ is larger than 1, which is up to about 0.2 rad/min. The magnitude 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.

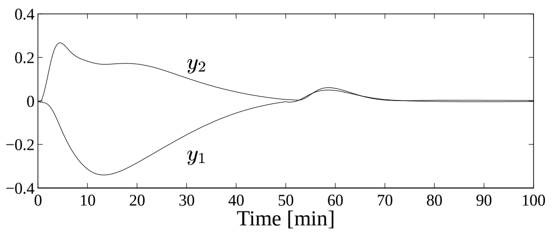


Figure 10.14: Decentralized PI-control. Responses to a unit step in d_1 at $t = 0$ and a unit step in d_2 at $t = 50$ min

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:*

$$k_1(s) = 0.261\frac{1+3.76s}{3.76s}; \quad k_2(s) = -0.375\frac{1+3.31s}{3.31s} \tag{10.77}$$

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 10.14. The simulations confirm that disturbance 2 is more easily rejected than disturbance 1.

In summary, there is an excellent agreement between the controllability analysis and the simulations, as has also been confirmed by a number of other examples.

10.8.7 Sequential design of decentralized controllers

The results presented in this section on decentralized control are most useful for the case when the local controllers $k_i(s)$ are designed *independently*, that is, each controller is designed locally and then all the loops are closed. As discussed above, one problem with this is that the interactions may cause the overall system (T) to be unstable, even though the local loops ($\widetilde{T}$) are stable. This will not happen if the plant is diagonally dominant, such that we satisfy, for example, $\bar{\sigma}(\widetilde{T}) < 1/\mu(E)$ in (10.51).

The stability problem is avoided if the controllers are designed *sequentially* as is commonly done in practice when, for example, the bandwidths of the loops are quite different. In this case the outer loops are tuned with the inner (fast) loops in place, and each step may be considered as a SISO control problem. In particular, overall stability is determined by m SISO stability conditions. However, the issue of performance is more complicated because the closing of a loop may cause "disturbances" (interactions) into a previously designed loop. The engineer must then go back and redesign a loop that has been designed earlier. Thus sequential design may involve many iterations; see Hovd and Skogestad (1994b). The performance bounds in (10.72) are useful for determining the required bandwidth in each loop and may thus suggest a suitable sequence in which to design the controllers.

Although the analysis and derivations given in this section apply when we design the controllers sequentially, it is often useful, after having designed a lower-layer controller (the inner fast loops), to redo the analysis based on the model of the partially controlled system using (10.26) or (10.28). For example, this is usually done for distillation columns, where we base the analysis of the composition control problem on a 2×2 model of the partially controlled 5×5 plant, see Examples 10.5 and 10.10.

10.8.8 Conclusions on decentralized control

In this section, we have derived a number of conditions for the stability, e.g. (10.51) and (10.55), and performance, e.g. (10.65) and (10.66), 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 based on local models or even on-line. The conditions/bounds are also useful in an input-output controllability analysis for determining the viability of decentralized control.

Some exercises which include a controllability analysis of decentralized control are given at the end of Chapter 6.

10.9 Conclusion

The issue of control structure design is very important in applications, but it has received relatively little attention in the control community during the last 40 years. In this chapter, we have discussed the issues involved, and we have provided some ideas and tools. There is clearly a need for better tools and theory in this area.

11

MODEL REDUCTION

This chapter describes methods for reducing the order of a plant or controller model. We place considerable emphasis on reduced order models obtained by residualizing the less controllable and observable states of a balanced realization. We also present the more familiar methods of balanced truncation and optimal Hankel norm approximation.

11.1 Introduction

Modern controller design methods such as $\mathcal{H}_\infty$ and LQG, produce controllers of order at least equal to that of the plant, and usually higher because of the inclusion of weights. These control laws may be too complex with regards to practical implementation and simpler designs are then sought. For this purpose, one can either reduce the order of the plant model prior to controller design, or reduce the controller in the final stage, or both.

The central problem we address is: given a high-order linear time-invariant stable model G, find a low-order approximation G_a such that the infinity ($\mathcal{H}_\infty$ or $\mathcal{L}_\infty$) norm of the difference, $||G - G_a||_\infty$, is small. By model order, we mean the dimension of the state vector in a minimal realization. This is sometimes called the McMillan degree.

So far in this book we have only been interested in the infinity ($\mathcal{H}_\infty$) norm of stable systems. But the error $G - G_a$ may be unstable and the definition of the infinity norm needs to be extended to unstable systems. $\mathcal{L}_\infty$ defines the set of rational functions which have no poles on the imaginary axis, it includes $\mathcal{H}_\infty$, and its norm (like $\mathcal{H}_\infty$) is given by $||G||_\infty = \sup_w \bar{\sigma}\,(G(jw))$.

We will describe three main methods for tackling this problem: balanced truncation, balanced residualization and optimal Hankel norm approximation. Each method gives a stable approximation and a guaranteed bound on the error in the approximation. We will further show how the methods can be employed to reduce the order of an *unstable* model G. All these methods start from a special state-space