Design of Robust Decentralized Controllers

Morten Hovd Sigurd Skogestad*
Chemical Engineering, University of Trondheim, NTH,
N-7034 Trondheim, Norway.

Presented at AIChE Annual Meeting, Miami Beach, Florida, November 1-6 1992
Session: Advances in Process Control 1, Paper No. 127d

Abstract

The procedure for independent design of robust decentralized controllers proposed by Skogestad and Morari [20] is improved by requiring the controller to be a decentralized Internal Model Control (IMC) type controller. It is shown how to find bounds on the magnitude of the IMC filter time constants such that robust stability or performance is guaranteed. This allows the use of real perturbation blocks for modeling the uncertainty associated with the controllers. In contrast, Skogestad and Morari [20] found bounds on the sensitivity functions and complementary sensitivity functions for the individual loops, and therefore allowed a much larger class of designs, resulting in more conservative conditions.

The concept of Robust Decentralized Detunability is introduced. If a system is Robust Decentralized Detunable, any subset of the loops can be detuned independently and to an arbitrary degree without endangering robust stability. A simple test for Robust Decentralized Detunability is developed for systems controlled by a decentralized IMC controller.

The problem of sequential design of robust decentralized controllers is also addressed. It is shown how to include into the design problem for loop k a simple estimate the effect of closing subsequent loops will have on loop k and the loops that have already been closed.

*To whom correspondence should be addressed. FAX: +47-7-594080, e-mail: skoge@kjemi.unit.no

1 Introduction

Decentralized control remains popular in the chemical process industry, despite developments of advanced controller synthesis procedures leading to full multivariable controllers. Some of the reasons for the continued popularity of decentralized control are:

1. Decentralized controllers are easy to implement.
2. They are easy for operators to understand.
3. The operators can be allowed to retune the controllers to take account of changing process conditions (as a result of 2 above).
4. Some measurements or manipulated variables may fail. Tolerance of such failures are more easily incorporated into the design of decentralized controllers than full controllers.
5. The control system can be brought gradually into service during process startup and taken gradually out of service during shutdown.

The design of a decentralized control system consists of two main steps:

a) Control structure selection, that is, choosing manipulated inputs and controlled outputs, and pairing inputs and outputs.

b) Design of each single-input single-output (SISO) controller.

In this paper we will consider b), and assume that a) has already been done (e.g. by using the tools in [8, 9]). Standard controller synthesis algorithms (e.g. H_2 or H_∞ synthesis) lead to multivariable controllers, and cannot handle requirements for controllers with a specified structure. Instead, some practical approaches to the design of decentralized controllers have evolved:

- Parameter optimization.
- Sequential design [3, 14, 16].
- Independent design [20].

We discuss all these three approaches to the design of decentralized controllers, with reference to the potential advantages of decentralized control listed above. Parameter optimization will be considered only briefly, whereas we discuss and illustrate the unique problems associated with sequential design in more detail. New results on independent design are presented which represent improvements over the existing design procedure. Throughout this work we will use the structured singular value (see below) as the measure of control quality.

A system is said to have robust stability if it is stable regardless of whatever uncertainty is contained within the system. Because of items 4 and 5 above, we would like the system to remain stable if any subset of the control loops are out of service, or if the individual controllers have been detuned. Furthermore, we would like this stability to be a *robust* property, We define such systems to be Robust Decentralized Detunable:

Definition 1 *A closed loop system is said to be Robust Decentralized Detunable if each controller element can be detuned independently by an arbitrary amount without endangering robust stability.*

Decentralized detunability for a given controller should not be confused with decentralized integral controllability (DIC), which is a property of the plant only. DIC implies that there for a given plant *exists* a decentralized controller with integral action in all channels that is decentralized detunable.

2 Notation

In this paper, $G(s)$ will denote the plant, which is assumed to be of dimension $n \times n$. The matrix consisting of the diagonal elements of $G(s)$ is denoted $\tilde{G}(s)$, and $g_{ij}(s)$ is the ij'th element of $G(s)$. The reference signal (setpoint) is denoted r, manipulated inputs are denoted u and outputs are denoted y. If disturbances are present, $G_d(s)$ denotes the (open loop) transfer function from disturbances d to outputs y. Throughout this work, all controllers are assumed to be completely decentralized. The decentralized conventional feedback controller is denoted $C(s)$, with i'th diagonal element $c_i(s)$ (Fig. 1a). Likewise, the decentralized IMC controller is denoted Q, with i'th diagonal element $q_i(s)$ (Fig. 1b). The controllers $C(s)$ and $Q(s)$ are related by

$$C(s) = Q(s)(I - \tilde{G}(s)Q(s))^{-1} \tag{1}$$

The sensitivity function is $S(s) = (I + G(s)C(s))^{-1}$ and the $H(s) = I - S(s) = G(s)C(s)(I + G(s)C(s))^{-1}$ is the complementary sensitivity function. The sensitivity functions and complementary sensitivity functions for the individual loops are collected in the diagonal matrices $\tilde{S}(s) = (I - \tilde{G}(s)C(s))^{-1}$ and $\tilde{H}(s) = \tilde{G}(s)C(s)(I - \tilde{G}(s)C(s))^{-1}$. Note that the diagonal elements of $\tilde{S}(s)$ and $\tilde{H}(s)$ do not equal the diagonal elements of $S(s)$ and $H(s)$, respectively. The i'th element on the diagonal of $\tilde{S}$ and $\tilde{H}$ are $\tilde{s}_i$ and $\tilde{h}_i$, respectively.

3 Robust Control and the Structured Singular Value

Since no model is a perfect representation of the system, the control system stability and performance should be little affected by the uncertainties of the model. In this paper we use the structured singular value, μ, introduced by Doyle [5], as a measure of the robustness of feedback systems. Within the μ framework, one accepts that it is the impossible to find a perfect model, and instead require information about the structure, location and estimates of the magnitude of the model uncertainties.

In Fig. 2 we have drawn an example of a feedback system with uncertainty in the inputs and outputs[1], represented by the perturbation blocks Δ_I and Δ_O, respectively. Note that the individual perturbations can be restricted to have a certain structure.

[1]Many other types of uncertainties possible, see [5] for details on how to represent different uncertainties with perturbation blocks.

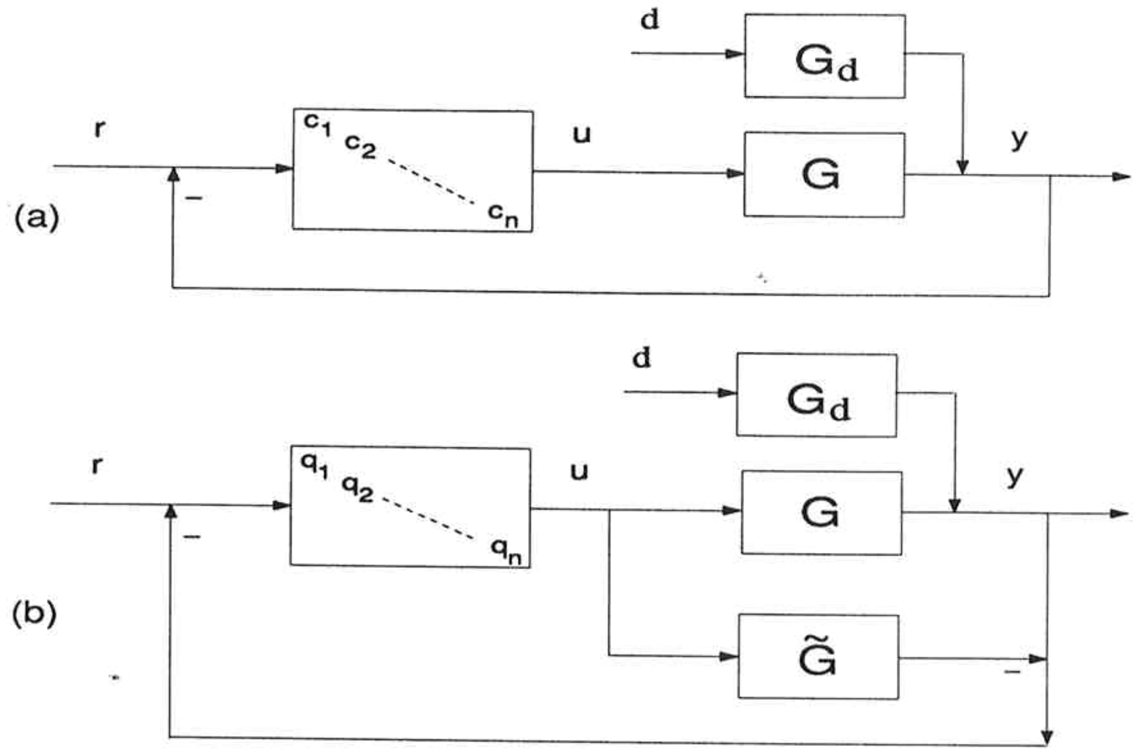

Figure 1: Block diagram of feedback systems. (a) Conventional decentralized controller. (b) Decentralized IMC controller.

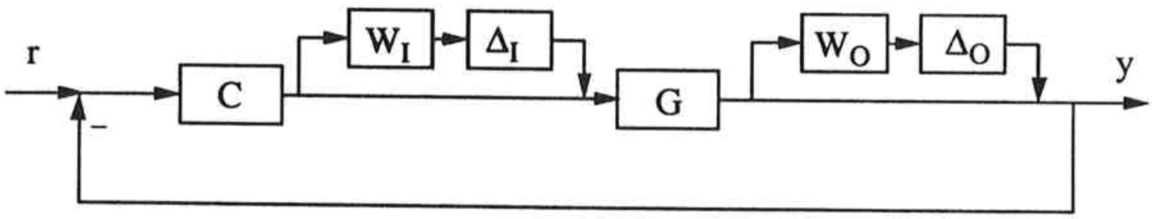

Figure 2: Block diagram for feedback system with uncertainty in the inputs and outputs.

For instance, as individual inputs and outputs usually do not affect each other, both Δ_I and Δ_O are assumed to be diagonal. The weights W_I and W_O are frequency-dependent and normalize the maximum magnitude of Δ_I and Δ_O to unity.

Any block diagram with uncertainties represented by perturbation blocks can be rearranged into the $M - \Delta$ structure of Fig. 3, if external inputs and outputs are neglected. In Fig. 3, Δ is a block diagonal matrix with the perturbation blocks of the original block diagram on the diagonal, and M contains all the other blocks in the block diagram (plant, controller, weights). For the specific case in Fig. 2, we have that

$$\Delta = \text{diag}\{\Delta_I, \Delta_O\}; \qquad M = \begin{bmatrix} -W_I CG(I+CG)^{-1} & -W_I C(I+GC)^{-1} \\ W_O G(I+CG)^{-1} & -W_O GC(I+GC)^{-1} \end{bmatrix}$$

Provided M is stable (the system has Nominal Stability, NS) and Δ is norm bounded and stable (stable perturbation blocks), it follows from the Nyquist stability criterion

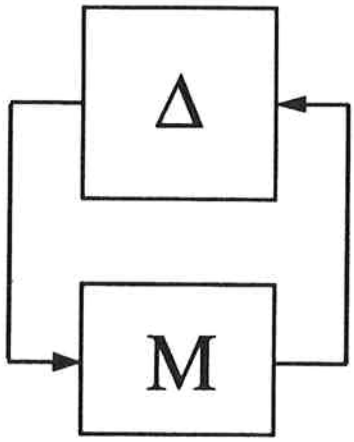

Figure 3: Feedback system rearranged into a perturbation block Δ and an interconnection matrix M.

[5] that the overall system is stable provided $\det(I - M\Delta) \neq 0 \quad \forall\Delta, \forall\omega$. In this case the system is said to have Robust Stability (RS). The structured singular value is defined such that

$$\mu_\Delta^{-1} = \min_\delta\{\delta | \det(I - M\Delta) = 0 \quad \text{for some } \Delta, \bar{\sigma}(\Delta) \leq \delta\} \tag{2}$$

If weights are used to normalize the maximum value of the largest singular value of Δ to unity ($\bar{\sigma}(\Delta) = 1$) at all frequencies, like in Fig. 2, the system will remain stable for any allowable perturbation Δ provided $\mu_\Delta(M) < 1$.

Doyle [5] showed that performance can be analyzed in the μ framework by considering an equivalent stability problem of larger dimension. In this paper we use a performance specification of the type $\bar{\sigma}(W_P S_p) \leq 1 \quad \forall\omega$ where S_p is the worst sensitivity function (S) made possible by the perturbation blocks. This performance specification can be incorporated in the μ framework by closing the loop from outputs to output disturbances with the performance weight W_P and a *full* perturbation block Δ_P. If $\mu_\Delta(M) \leq 1 \quad \forall\omega$ (after normalizing the magnitude of the perturbation blocks) and M is stable for the corresponding $M - \Delta$ structure of increased dimension (in our specific example, $\Delta = \text{diag}\{\Delta_I, \Delta_O, \Delta_P\}$), the system is said to have Robust Performance (RP), as the performance specification is fulfilled for all the possible model uncertainties.

To simplify notation, we will use "$\mu(M)$" in the meaning $\sup_\omega \mu_\Delta(M)$. Doyle and Chu [6] proposed an algorithm for the synthesis of controllers which minimizes μ, known as $D - K$ iteration. However, $D - K$ iteration results in full controllers, and the problem of synthesizing μ-optimal decentralized controllers has not been solved.

4 Parameter Optimization

When using parameter optimization, an *à priori* parametrization of the controller and the chosen measure of control quality (in our case μ) is optimized with respect to the controller parameters, using some optimization routine. Controller design using parameter optimization is easily formulated in a computer program and often gives satisfactory designs (e.g. [21]). However, the optimization is not necessarily convex, and problems with local minima may be encountered. Another problem is that, since all the loops are assumed to be in service at each step in the optimization, advantages 4 and 5 in the introduction may not be achieved. One must therefore check specifically whether these advantages are achieved after the optimization is finished. More importantly, the parameter optimization approach gives no guidelines for how to achieve advantages 4 and 5 if analysis of a proposed controller shows that they are not achieved.

5 Independent Design

Independent design of robust decentralized controllers was introduced by Skogestad and Morari [20]. It is based on Theorem 1 in [19], which we state here:

Theorem 1 *Let the μ interconnection matrix M be written as a lower Linear Fractional Transformation (LFT) of the transfer function matrix T*

$$M = F_l(N,T) = N_{11} + N_{12}T(I - N_{22}T)^{-1}N_{21} \tag{3}$$

and let k be a given constant. Assume $\mu_\Delta(N_{11}) < 1$ and $det(I - N_{22}T) \neq 0$ then

$$\mu_\Delta(M) \leq 1 \tag{4}$$

if

$$\bar{\sigma}(T) \leq c_T \tag{5}$$

where c_T solves

$$\mu_{\tilde{\Delta}} \begin{bmatrix} N_{11} & N_{12} \\ c_T N_{21} & c_T N_{22} \end{bmatrix} = 1 \tag{6}$$

and $\tilde{\Delta} = \mathrm{diag}\{\Delta, T\}$

Proof: See [19].

The condition $\mu_\Delta(M)$ is typically the RP condition we want to satisfy, and T is some important transfer function which depends on the controller. Skogestad and Morari [20] uses Thm. 1 to find bounds on the sensitivity function and complementary sensitivity functions for the individual loops (i.e. $T = \tilde{S}$ and $T = \tilde{H}$ are used). The bounds on $\tilde{S}$ and $\tilde{H}$ can be combined over different frequency ranges. Thus, if either the bound on $\tilde{S}$ or the bound on $\tilde{H}$ is fulfilled for all loops at all frequencies, then $\mu_\Delta(M) \leq 1$ is achieved.

The rationale behind Thm. 1 is to treat the transfer functions (T) as a "class of possible designs" (i.e. as uncertainty), and finds bounds on the magnitude of this fictitious uncertainty which guarantees that $\mu_\Delta(M) \leq 1$. One is faced with finding controllers such that the bounds on the transfer functions are fulfilled. It is therefore important for the success of independent design that T introduces as little additional uncertainty as possible. It turns out that parametrizing the class of possible designs as $T = \tilde{S}$ and $T = \tilde{H}$ are not ideal for this purpose.

5.1 Example 1

Consider Example 1 in Chiu and Arkun [3]:

$$G(s) = \begin{bmatrix} \frac{1.66}{39s+1} & \frac{-1.74e^{-2s}}{4.4s+1} \\ \frac{0.34e^{-s}}{8.9s+1} & \frac{1.4e^{-s}}{3.8s+1} \end{bmatrix} \tag{7}$$

There is independent input uncertainty with input uncertainty weight $W_I(s) = 0.07I_2$, and the performance requirement is given by the performance weight $W_p(s) = 0.25\frac{7s+1}{7s}I_2$

Chiu and Arkun [3] attempted independent design for this example, using $T = \tilde{S}$ and $T = \tilde{H}$, but were unable to find a controller which fulfilled the resulting bounds. In [3] it was therefore claimed that independent design can not be performed for this example. We will however demonstrate below that independent design can be performed for this example, by parametrizing the class of possible designs within the framework of Internal Model Control.

5.2 Independent Design with Decentralized IMC Controllers

We use the Internal Model Control (IMC) technique [7] to parametrize the individual controller elements, and select T not as a transfer function, but rather as a parametrization of the tuning constant ϵ_i in the IMC controller. Our approach is similar to that of Lee and Morari [10], but we use ϵ_i as the parameter rather than the filter f_i. The relationship between the elements q_i of the IMC controller and the elements c_i of the conventional controller is given by

$$c_i = q_i(1 - g_{ii}c_i)^{-1} \tag{8}$$

In the IMC design procedure [15], q_i has the form

$$q_i = \hat{g}_{ii}^{-1} f_i \tag{9}$$

where $\hat{g}_{ii}$ is the minimum phase part of g_{ii}, and f_i is a low pass filter used to make q_i realizable and to detune the system for robustness. In order to simplify the exposition, we will assume the plant G to be open loop stable, and use a low pass filter of the form

$$f_i = \frac{1}{(\epsilon_i s + 1)^{n_f}} \tag{10}$$

That is, the f_i is taken to be a low pass filter of order n_f, consisting of n_f *identical* first order low pass filters in series. For details on IMC design, and on filter form for unstable systems, the reader is referred to Morari and Zafiriou [15].

Choice of T for Independent Design. After fixing n_f, the only thing which remains uncertain in the IMC technique is the value of ϵ_i. To fulfill performance requirements at low frequencies, the closed loop system must be sufficiently fast, which means that the filter time constant ϵ must be smaller than a certain value. On the other hand,

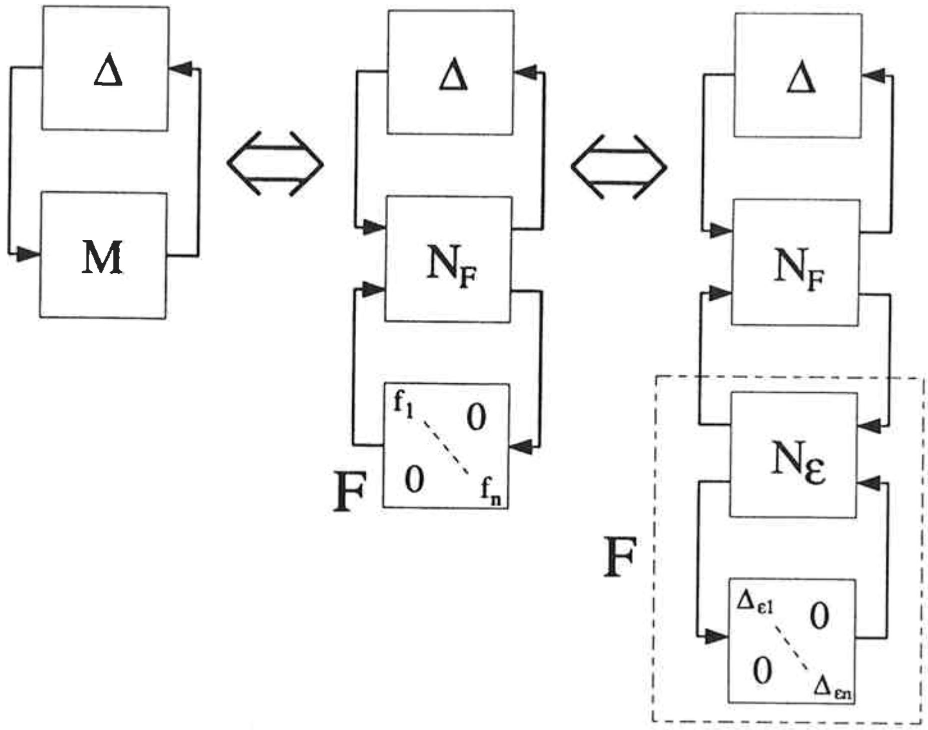

Figure 4: The interconnection matrix M expressed as an LFT of the IMC filter F and as an LFT of the "uncertainty" associated with the filter time constants.

the closed loop system must be sufficiently detuned to avoid robustness problems at higher frequencies, thus requiring ϵ to be larger than a certain value, meaning that $1/\epsilon$ must be smaller than some value. We will therefore use Thm. 1 to find bounds on ϵ and $e_i \stackrel{\text{def}}{=} 1/\epsilon$ which can be combined over different frequency ranges. Since we are using a specific control structure the class of possible designs is much smaller than if we use Thm. 1 to find bounds on $\tilde{S}$ and $\tilde{H}$. Bounds on $\tilde{S}$ and $\tilde{H}$ are therefore potentially much more conservative.

To derive conditions on ϵ_i and e_i that guarantee $\mu_\Delta(M) \leq 1$ we will proceed as follows: First we parametrize the μ interconnection matrix M as an LFT of the IMC filter F and then as an LFT of the "uncertainty" in the filter time constant. We refer the readers to [19] or [15] for details on how to find the LFT's needed in Fig. 4. Below we will only elaborate on how to express f_i as an LFT of the "uncertainty" associated with ϵ_i or e_i. We then solve Eq. (6) at each frequency point to find the desired bound. Note that it is sufficient at each frequency to satisfy the bound either for ϵ_i or for $e_i = 1/\epsilon_i$, but we must of course use the same bound for ϵ_i (and e_i) at all frequencies. Note that although $\hat{g}_{ii}^{-1}$ in Eq. (9) will normally not be realizable, its frequency response is easily calculated. Also note that since we work with the frequency response, we will have to check à posteriori for the (internal) stability of the μ interconnection matrix.

First Order Low Pass Filters. Consider first the case $n_f = 1$. We then have $f_i = 1/(\epsilon_i s + 1)$. The objective is to find the allowable ranges for ϵ_i and $e_i = 1/\epsilon_i$ that at each frequency guarantee $\mu(M) \leq 1$. Since we do not allow negative values for ϵ_i

we should not write $|\epsilon_i| \leq c_\epsilon$. Instead write

$$\epsilon_i = \frac{\epsilon^*}{2}(1+\Delta_\epsilon) \qquad |\Delta_\epsilon| \leq 1 \tag{11}$$

$$e_i = \frac{e^*}{2}(1+\Delta_e) \qquad |\Delta_e| \leq 1 \tag{12}$$

and fix $c_\epsilon = 1$ and $c_e = 1$ in Eq. (6). The μ interconnection matrix in Eq. (6) then depends only on ϵ^* or e^* (depending on whether we try to find bounds on ϵ_i or e_i). Note that all quantities, including Δ_ϵ and Δ_e are real. In order to use Thm. 1 we now need to write f_i as an LFT of Δ_ϵ and Δ_e. For $T = \Delta_\epsilon$ we have

$$f_i = \frac{1}{\frac{\epsilon^*}{2}(1+\Delta_\epsilon)s+1} \tag{13}$$

which may be written as an LFT, $f_i = F_l(N_{\epsilon_i}, \Delta_\epsilon)$, with

$$N_{\epsilon_i} = \frac{1}{\frac{\epsilon^*}{2}s+1}\begin{bmatrix} 1 & -1 \\ \frac{\epsilon^*}{2}s & \frac{-\epsilon^*}{2}s \end{bmatrix} \tag{14}$$

Similarly, for $T = \Delta_e$, we have

$$N_{e_i} = \frac{1}{\frac{e^*}{2}+s}\begin{bmatrix} \frac{e^*}{2} & \frac{e^*}{2} \\ s & \frac{-e^*}{2} \end{bmatrix} \tag{15}$$

This shows how to express an individual filter element f_i as an LFT of the real "uncertainty" in the filter time constant in that filter element. The LFT for the overall IMC filter $F = \mathrm{diag}\{f_i\}$ is then just a simple diagonal augmentation of the corresponding blocks of the LFT for the individual filter elements. For example, let N_{11}^i denote the N_{11} block for the LFT of element i. The block N_{11} for the LFT of the overall IMC filter will then be given by $N_{11} = \mathrm{diag}\{N_{11}^i\}$.

We use $|\Delta_\epsilon| \leq 1$ and $|\Delta_e| \leq 1$ and correspondingly fix $c_\epsilon = 1$ and $c_e = 1$ in Eq. (6). The required bounds for ϵ_i and e_i are then found by iterating on the value of ϵ^* or e^* (as appropriate) until $\mu_{\tilde{\Delta}} = 1$ (the matrix N in Eq. (6) will depend on ϵ^* when we iterate on ϵ^*, and depend on e^* when we iterate on e^*). Denote the values of ϵ^* and e^* which give $\mu_{D\tilde{e}lta} = 1$ ϵ_s^* and e_s^*, respectively. For a fixed frequency, we are then guaranteed that $\mu_\Delta(M) \leq 1$ provided

$$\epsilon_i \leq \epsilon_s^* \,\forall i; \quad \text{or} \tag{16}$$

$$e_i \leq e_s^* \,\forall i \iff \epsilon_i \leq \frac{1}{e_s^*} \,\forall i \tag{17}$$

Note that although the ϵ_i's are independent, we get the same bound for all ϵ_i.

The iterations are performed using only positive real values for ϵ^* and e^*. Note that $\mu_{\tilde{\Delta}}$ in Eq. (6) is non-decreasing with increasing values of ϵ^* or e^*. This follows from the fact that we use $|\Delta_\epsilon| \leq 1$ and $|\Delta_e| \leq 1$, therefore the set of possible values for $\epsilon_i = \frac{\epsilon^*}{2}(1+\Delta_\epsilon)$ for any fixed value of ϵ^* contains all the possible values for ϵ_i for any smaller ϵ^* (and similarly for e_i and e^*). A very simple iteration scheme, e.g. bisection, can therefore be used.

Higher Order Low Pass Filters. In IMC design, one will often use filters of order higher than one. We therefore need to be able to express the higher order filters as LFT's of Δ_ϵ and Δ_e. For this we can use the rules for series interconnection of linear dynamical systems. First note that $G(s) = C(sI - A)^{-1}B + D$ may be written as an LFT of $\frac{1}{s}I$, with

$$N_{11} = D; \quad N_{12} = C; \quad N_{21} = B; \quad N_{22} = A$$

The formulae for series interconnection $G = G_1 G_2$ of dynamical systems $G_1(s) = C_1(sI - A_1)^{-1}B_1 + D_1$ and $G_2(s) = C_2(sI - A_2)^{-1}B_2 + D_2$ are (e.g. [13]):

$$\begin{aligned} A &= \begin{bmatrix} A_1 & 0 \\ B_2 C_1 & A_2 \end{bmatrix} \\ B &= \begin{bmatrix} B_1 \\ B_2 D_1 \end{bmatrix} \\ C &= \begin{bmatrix} D_2 C_1 & C_2 \end{bmatrix} \\ D &= D_2 D_1 \end{aligned}$$

The formulae for series interconnection of dynamical systems can be used directly to express an n_f'th order low pass filter as LFT's of $\text{diag}\{\Delta_{\epsilon 1}, \cdots, \Delta_{\epsilon n_f}\}$ and $\text{diag}\{\Delta_{e1}, \cdots, \Delta_{en_f}\}$. As we will normally use the same time constant for all first order factors of the n_f'th order filter, we will have $\Delta_{\epsilon 1} = \Delta_{\epsilon 2} = \cdots = \Delta_{\epsilon n_f}$ and $\Delta_{e1} = \Delta_{e2} = \cdots = \Delta_{en_f}$, and we have repeated scalar, real "uncertainty" associated with the filter in each IMC controller element.

Note that although we have repeated scalar "uncertainties" for each individual filter element, the filter time constants may differ in different filter elements, and the "uncertainties" in different filter elements are therefore independent. For a plant of dimension $n \times n$ we therefore end up with n repeated scalar uncertainty blocks for the IMC filter, each of these blocks being repeated n_f times[2].

5.3 Independent Design Procedure

With the preliminaries above, we can now propose an independent design algorithm:

1. Find the matrices N_ϵ, expressing the μ interconnection matrix M as an LFT of Δ_ϵ, and the matrix N_e, expressing M as an LFT of Δ_e. N_ϵ will depend on the value of ϵ^*, and N_e will depend on the value of e^*, and we must therefore recompute N_ϵ and N_e for every new value of ϵ^* and e^*, respectively.

2. We get

$$\mu(M) \le 1$$

if

$$0 \le \epsilon_i \le \epsilon_s^* \quad \forall i \tag{18}$$

[2]One may use low pass filter of different orders in the different filter elements, in which case the value of n_f will differ for different filter elements.

where ϵ_s^* solves

$$\mu(N_\epsilon) = 1 \tag{19}$$

Similarly, let e_s^* solve $\mu(N_e) = 1$, giving the bound

$$0 \leq e_i \leq e_s^* \quad \forall i \quad \Longleftrightarrow 1/e_s^* \leq \epsilon_i \quad \forall i \tag{20}$$

3. From 2 and Thm. 1 we know that $\mu(M) < 1$ for the range of values of ϵ which at *all* frequencies is *either* within the range of values in Eq. (18) *or* within the range of values in Eq. (20).

4. Choose a value of ϵ within the range of values found in point 3, and verify the stability of M for this choice of ϵ. [3]

If we are successful in Steps 3 and 4, the controller design is completed. Since we have both real and complex perturbations, Step 2 requires μ calculations for mixed real and complex perturbations [24], which is still a research topic. However, the existing μ software has proved to be acceptable in many cases.

5.4 Examples

5.4.1 Example 1 (continued)

Consider again Example 1 studied above. For this problem we choose a second order low pass filter in each element of the decentralized IMC controller. Since we have a 2×2 system, this will add two real, repeated scalar perturbations, each repeated twice. Solving Eq. (6), we obtain the results in Fig. 5. We see that values of ϵ between

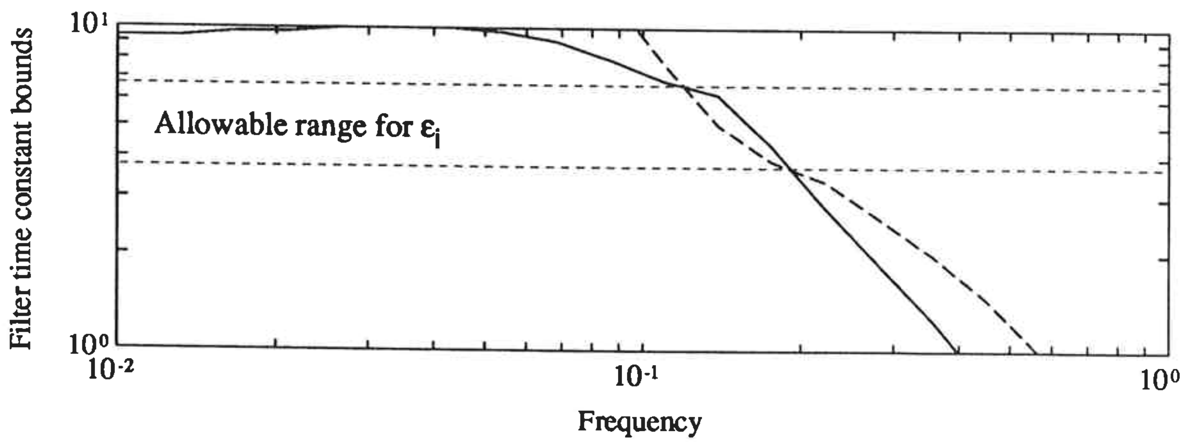

Figure 5: Filter time constant bounds for Example 1. Solid: ϵ_s^* (upper bound). Dashed: $1/e_s^*$ (lower bound).

[3] For any value of ϵ within the range found in point 3, the map under the Nyquist D-contour of $\det(I - M\Delta)$ will encircle the origin the same number of times. Thus, if M is found to be unstable in Step 4, it is "robustly unstable".

3.7 and 6.6 are at all frequencies either below the upper bound or above the lower bound. Choosing $\epsilon = 5$ for both loops, it is easily verified that the system is nominally (internally) stable. We have thus completed an independent design for this example.

5.4.2 Example 2

Here we consider Example 2 in [3].

$$G(s) = \begin{bmatrix} \frac{0.66}{6.7s+1} & \frac{-0.61}{8.4s+1} & \frac{-0.005}{9.06s+1} \\ \frac{1.11}{3.25s+1} & \frac{-2.36}{5s+1} & \frac{-0.01}{7.09s+1} \\ \frac{34.7}{8.15s+1} & \frac{46.2}{10.9s+1} & \frac{0.87(11.61s+1)}{(3.89s+1)(18.8s+1)} \end{bmatrix} \tag{21}$$

In this example only robust stability is considered, with independent, multiplicative input uncertainty with uncertainty weight $W_I(s) = 0.13\frac{5s+1}{0.25s+1}$. In [3] it is found that independent design using Thm. 1 with $T = \tilde{H}$ and $T = \tilde{S}$ cannot be used to design a robust controller for this example. Since the process is stable and only multiplicative uncertainty is considered, this clearly illustrates the shortcomings of that method.

As for Example 1, a second order low pass filter is used in each diagonal element of the IMC controller. This will add three real, repeated scalar perturbations, each repeated twice. From Step 2 in the independent design procedure we obtain the results

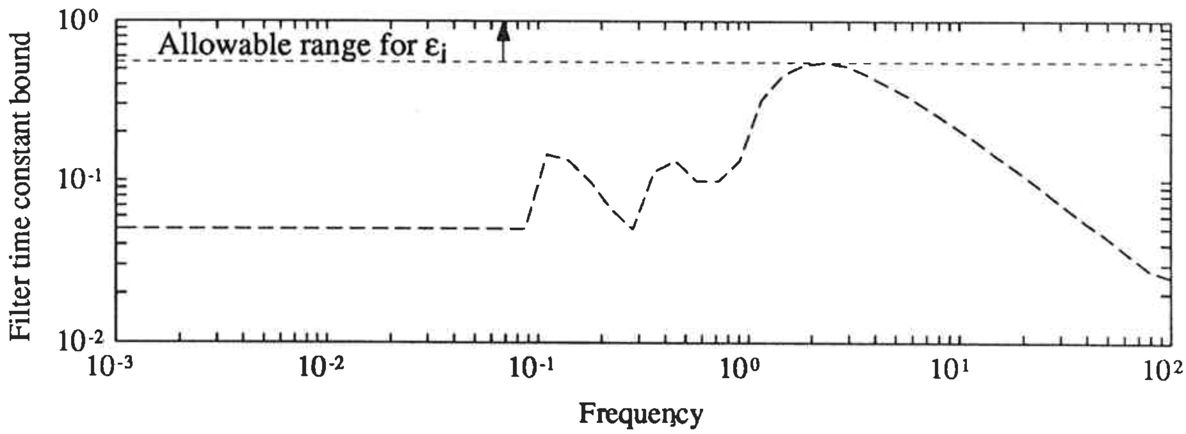

Figure 6: Lower bound on filter time constant ($1/\epsilon_s^*$) for Example 2.

in Fig. 6. From Fig. 6 we see that any value of ϵ larger than 0.55 will be acceptable. Choosing $\epsilon = 1$ for all loops, we find that the system is stable. We thus find that the system will be robustly stable for any value of $\epsilon_i > 0.55$. In general we want ϵ to be small for a faster nominal response.

For both Example 1 and Example 2, Chiu and Arkun [3] were unable to perform an independent design, using the procedure of Skogestad and Morari [20]. This demonstrates the importance of introducing as little conservatism as possible in the description of the uncertainty associated with the controllers when performing an independent design.

5.4.3 Robust Decentralized Detunability in the IMC Framework

In the IMC framework, controllers are detuned by increasing the filter time constants. We have thus found for Example 2 above that the loops can be detuned independently of each other, without endangering robust stability, provided all loops have $\epsilon_i > 0.55$. Thus the closed loop system in Example 2 with $\epsilon_i > 0.55$ in all loops is found to be robust decentralized detunable according to Definition 1. After removing the performance requirement from Example 1 and redoing the calculations for robust stability, we find that it is robust decentralized detunable provided $\epsilon_i > 0.16$ for both loops.

A requirement for Robust Decentralized Detunability is that the individual loops are stable. A decentralized IMC controller as parametrized in Eq. (9) will make the individual loops stable, which in most cases is an advantage. However, integral action is inherent in IMC controllers, and integral action and stability of the individual loops is known to be incompatible with stability of the overall system for certain plants. We would like to emphasize Step 4 in the Independent design procedure, that nominal stability must be checked explicitly for one value of ϵ within the bounds found. The Niederlinski Index criterion [17] gives a necessary condition for obtaining stability both of the individual loops and the overall system when there is integral action in all channels. The Niederlinski Index criterion has recently been generalized to open loop unstable plants [9]. Let the number of Right Half Plane (RHP) poles in G be n_U (including multiplicities), and the number of RHP poles in $\tilde{G}$ be $\tilde{n}_U$. Note that in general $\tilde{n}_U \neq n_U$. If all the individual loops are stable, a necessary condition for the stability of the overall system is that

$$\text{sign}\{N_I\} = \text{sign}\{\frac{\det G(0)}{\det \tilde{G}(0)}\} = \text{sign}\{(-1)^{-n_U+\tilde{n}_U}\} \tag{22}$$

Thus, before attempting to perform an independent design, one should check that overall stability can be achieved with integral action in all channels and having stable individual loops.

5.4.4 Example 3

Consider the process

$$G(s) = \begin{bmatrix} \frac{5}{20s^2+12s+1} & \frac{8}{20s^2+12s+1} \\ \frac{6}{40s^2+12s+1} & \frac{2}{40s^2+12s+1} \end{bmatrix} \tag{23}$$

with independent actuator uncertainty with uncertainty weight $W_I(s) = 0.2\frac{10s+1}{s+1}I_2$. Since this plant is stable and the Niederlinski Index is negative, $N_I = -3.8$, we know that we cannot have the individual loops stable and at the same time achieve overall system stability. Nevertheless, we proceed with independent design, and choose third order low pass filters for both loops. We find that Step 3 in the independent design procedure indicates that any value of $\epsilon > 4$ (approximately) will give robust stability (figure omitted). Calculating μ for $\epsilon = 5$ for both loops, we do indeed obtain a value of $\mu < 1$ at all frequencies. The reason, which we find in Step 4 in the independent design procedure, is that the overall system is nominally unstable. The μ test merely tells us that this instability is a robust property. For other cases, it may not be this easy to tell *à priori* that the overall system will be unstable with the individual loops stable.

5.5 Conclusions on Independent Design

We have proposed a parametrization of the class of allowable decentralized designs which is based on the following four key steps:

1. Use an IMC controller design for each loop.
2. Select the filter time constant ϵ_i as the "uncertain" parameter.
3. Parametrize ϵ_i and $e_i = 1/\epsilon_i$ such that only positive values are allowed.
4. Obtain bounds on both ϵ_i and e_i that guarantee robust stability/performance.

We have found that:

- The result of considering only decentralized IMC controllers with a specified filter structure, is that the set of possible controller designs considered is much smaller than the set of possible controller designs when trying to find bounds on $\tilde{S}$ and $\tilde{H}$, and the resulting bound are therefore less conservative.
- One can derive a bound on the IMC filter time constants which ensures that the system is robust decentralized detunable.
- It is critical that real perturbations are used for the parametrization of ϵ_i and e_i. μ software capable of handling real perturbations is therefore needed.

The bounds obtained are common to all the filter elements, and it is not obvious how to take advantage of the possibility of having differing filter time constants in the different filter elements. However, one may easily use constant ratios between the filter time constants in the independent design procedure (e.g. choosing $\epsilon_1 = \epsilon^*$, $\epsilon_2 = 10\epsilon^*$, etc.).

If independent design fails in the first step with our improved independent design procedure, the "uncertainty" associated with the filter time constants can be reduced even further by assuming all filter time constants to have fixed values relative to each other (e.g. assuming all filter time constants in *all* filter elements to be equal). This may be termed "simultaneous design". For a plant of size $n \times n$ and low pass filters of order n_f, this will reduce the "uncertainty" associated with the filter time constants from n real scalar uncertainties repeated n_f times, to one real scalar uncertainty repeated $n \times n_f$ times.

The independent design procedure proposed here can also be applied to other types of controllers, for example, one can find bounds on the ratio of gain to integral time (k/T_i) for PID controllers. However, decentralized IMC controllers have only one tuning parameter, and are therefore preferable for our independent design procedure.

One can easily use parameter optimization to find the filter time constants ϵ_i that minimize $\mu(M)$. Independent design has the advantage of providing a *range* of values for which robust stability/performance is fulfilled, and can also guarantee that the system is robust decentralized detunable.

6 Sequential Design

Sequential design of decentralized controllers was introduced in the control literature by Mayne [14], but it is probably fair to say that it has always been the most common way of designing decentralized controllers in industry. Sequential design involves closing and tuning one loop at the time. This is the advantage of sequential design: each step in the design procedure may be considered as a single-input single-output (SISO) control problem. The loops that have already been designed are (assumed to be) kept in service when closing and tuning subsequent loops. However, if the subsequent closing of other loops makes a loop perform badly, the engineer must go back and redesign a loop that has been closed earlier. Thus sequential design may involve iteration.

If the loops of a decentralized control system have been designed in the order $1, 2, \cdots, k, k+1, \cdots, n$ without having to redesign any controller element, and stability has been achieved after the design of each loop, sequential design will automatically ensure a limited degree of failure tolerance: The system will remain stable if *all* the loops $k+1, \cdots, n$ are to fail or be taken out of service simultaneously. Similarly, during startup the system will be stable if the loops are brought into service in the same order as they have been designed.

Sequential design of decentralized controllers has been addressed by several authors (e.g. [2, 3, 14, 16, 23]). However, the only published procedure for robust (in terms of μ) sequential design of decentralized controllers appear to be the one due to Chiu and Arkun [3, 4].

6.1 The Robust Sequential Design Procedure of Chiu and Arkun

The sequential design procedure of Chiu and Arkun [3, 4] involves performing the *independent* design procedure of Skogestad and Morari [20] at each step in the design. One loop is then closed with tuning parameters in accordance with the bounds obtained from independent design. A new independent design is then performed with this loop closed, and so on. If the independent design procedure fails (the bounds conflict and Eq. (5) cannot be satisfied) in the first step, Chiu and Arkun propose to close a sufficient number of loops to enable independent design to be performed. No guidelines are given for how to choose tuning parameters for these loops that have to be closed prior to the application of independent design.

Another weakness with their design method is that the controller design for loop k is done by finding bounds for all remaining loops $k - n$, which guarantee robust stability or performance if the controllers in *all* loops $k - n$ fulfill the bounds. Closing *only* loop k with a controller fulfilling the bounds found, will not necessarily mean that the subsystem consisting of loops $1 - k$ is stable.

The improved independent design procedure presented in Section 5 can readily be incorporated into the sequential design procedure of Chiu and Arkun. However, even with this improvement we may encounter all the problems mentioned above. There is therefore a need for a robust sequential design procedure starting from one single loop rather than a procedure involving a large number of loops simultaneously.

6.2 Problems Unique to Sequential Design

In sequential design, three problems arise which are not encountered using independent design or parametric optimization:

1. The final controller design, and thus the control quality achieved, may depend on the order in which the controller in the individual loops are designed.
2. The optimal design for the controller in loop k depends on the design of the controllers in *all* the other loops, some of which are still not designed.
3. The individual elements of the plant transfer function G may contain right half plane (RHP) zeros that do not correspond to RHP transmission zeros of G

The conventional rule for dealing with problem 1 is to close the fast loops first, the reason being that the loop gain and phase in the bandwidth region of the fast loops is relatively insensitive to the tuning of the slower loops. While this argument is reasonable for ***loop*** k, ***output*** k may still be sensitive to the tuning of the controller in a slower loop l, if u_l has a large effect on y_k.

We will attempt to reduce the severity of problem 2 by using simple estimates of how the undesigned loops will affect the output of the loop to be designed.

We require the system to be stable after the closing of each loop, but it may not be possible to close the fast loop (k) first, if the corresponding transfer function element has a significant RHP zero that is not a transmission zero of the plant G. However, such RHP zeros in the individual elements of G may disappear when the other loops are closed (as the RHP zero is not an transmission zero), and it may therefore be possible to achieve fast control in loop k if the controller for this loop is designed at a later stage. This is illustrated in Example 4 below.

6.3 Preliminaries

In the following, we will assume without loss of generality that the loops are closed (and controllers designed) in the order $1, 2, \cdots, k, k+1, \cdots$, and that the loop to be designed is k. Let G_k denote the submatrix of dimension $k \times k$ in the upper left corner of G. Introduce $\hat{G}_k = \text{diag}\{G_k, g_{ii}\}$, $i = k+1, k+2, \cdots, n$, $\hat{S}_k = (I + \hat{G}_k C)^{-1}$ and $\hat{H}_k = I - \hat{S}_k = \hat{G}_k C(I + \hat{G}_k C)^{-1}$. We then have

$$S = \hat{S}_k (I + E_k \hat{H}_k)^{-1} \tag{24}$$

where $E_k = (G - \hat{G}_k)\hat{G}_k^{-1}$. When performing sequential design, one should keep in mind that the effective transfer function from u_k to y_k can change when subsequent loops are closed. This is due to the interaction between the loops. We see that rows 1 to k of $(I + E_k \hat{H}_k)^{-1}$ expresses how interaction affects loops 1 to k, and can be considered as an *input weight* to $S_k = (I + G_k C_k)^{-1}$.

For the special case $k = 1$ we have $\hat{G}_1 = \tilde{G}$, $\hat{S}_1 = \tilde{S}$ and $\hat{H}_1 = \tilde{H}$ (see Notation). Recall that $\tilde{S}$ and $\tilde{H}$ consist of the closed loop transfer functions of the individual loops.

6.3.1 Loop Gain Requirements for Setpoint Following and Disturbance Rejection

Consider the feedback system in Fig. 1a. Assume that the plant transfer function G and the disturbance transfer function G_d are scaled such that the largest tolerable offset in any controlled variable has magnitude 1 and the largest individual disturbance expected has magnitude 1 at any frequency. Prior to designing the first loop we have $k = 1$ and $\hat{G}_1 = \tilde{G} = \text{diag}\{g_{ii}\}$ in Eq. 24. Eventually all the loops will be closed and at low frequencies we will have $\tilde{h}_i \approx 1 \forall i$. We use this information to predict the overall response in terms of the individual loop responses. Consider only frequencies below the bandwidths of all the loops ($\tilde{h}_i \approx 1 \; \forall i$ and $(I + E_1\tilde{H})^{-1} \approx \tilde{G}G^{-1}$) and find

$$e = y - r = -Sr + SG_d d \approx -\tilde{S}\Gamma r + \tilde{S}\Gamma G_d d; \quad \omega < \omega_B \tag{25}$$

where $\Gamma = \tilde{G}G^{-1} = \{\gamma_{ij}\}$ is known as the Performance Relative Gain Array (PRGA) [8] and $\Gamma G_d = \{\delta_{ik}\}$ is known as the Closed Loop Disturbance Gain (CLDG). Thus $|\gamma_{ij}(j\omega)|$ gives the loop gain requirement at frequency ω for a change in setpoint j to cause an acceptably small offset in output i. Likewise, $|\delta_{ik}(j\omega)|$ gives the loop gain requirement in loop i for rejecting disturbance k. The PRGA and CLDG are introduced in [8, 22], and a more detailed explanation of their uses can be found therein. Here we will use the PRGA for two purposes:

1. Determine the *order* of loop closing (closing first loops that are required to be fast).

2. Estimate loop gain requirements for counteracting interactions and disturbances, thereby finding an estimate of the complementary sensitivity functions ($\tilde{h}_i$'s) for the loops that are not closed.

6.4 Sequential Design Procedure

The proposed sequential design procedure is outlined here. We assume that the design specifications include a performance requirement of the type

$$\bar{\sigma}(W_p S[I \; G_d]) < 1 \; \forall \omega$$

i.e. we want to optimize *robust performance* (for some specified model uncertainty) both with respect to setpoint changes and disturbances[4]. Note that S can be expressed in terms of $\hat{S}_k$ as shown in Eq. (24). Obviously, we can only have a performance requirement for an output where we have a controller. For this reason, define W_{pk} as the matrix of dimension $k \times k$ consisting of the upper left corner of W_p. Likewise, define W_{1k} as the matrix consisting of the first k rows of $(I + E_k\hat{H}_k)^{-1}[I \; G_d]$, using an estimate of $\hat{H}_k$.

Our sequential design procedure is then for step k to design a SISO controller that minimizes $\sup_\omega \bar{\sigma}(W_{pk}S_k W_{1k})$. If model uncertainty is included the problem is to minimize

[4]If disturbances are not considered, an empty matrix can be substituted for G_d.

the structured singular value of some matrix, in which W_{1k} is used as an input weight for performance.

Comments to the Sequential Design Method:

Step 1. When designing loop 1, we have $\hat{G}_k = \tilde{G} = \text{diag}\{g_{ii}\}; \quad i = 1, \cdots, n$. An estimate of $\tilde{H} = \text{diag}\{\tilde{h}_i\}$ is needed for calculating $(I + E_k\tilde{H})^{-1}$. The loop gain requirements given above in terms of the PRGA and CLDG are helpful for this purpose, as will be demonstrated in the examples. W_{p1} consists of the first element on the main diagonal of W_p, and W_{11} is the first row of $(I+E_k\tilde{H})^{-1}[I \; G_d]$. For a plant of dimension $n \times n$ and with n_d disturbances, the perturbation block for performance will thus be of dimension $(n + n_d) \times 1$.

Step k. Here $\hat{G}_k = \text{diag}\{G_k, g_{ii}\}; \; i = k+1, \cdots, n\}$. Controllers for loops $1, \cdots, k-1$ have now been found, and $\hat{H}_k$ is estimated to be $\hat{H}_k = \text{diag}\{H_{k-1}, \tilde{h}_i\}; \; i = k, \cdots, n$, where $\tilde{h}_i$ is the original estimate of the complementary sensitivity functions for loop i. Here W_{1k} consists of the k first rows of $(I + E_k\tilde{H})^{-1}[I \; G_d]$, and the perturbation block for performance is of dimension $(n + n_d) \times k$.

Step n. The controllers in all the other loops have been designed, and we therefore have $W_{1n} = [I \; G_d]$, and the perturbation block for performance is of dimension $(n + n_d) \times n$.

6.4.1 Design Method for the Controllers in the Individual Loops

A choice has to be made as to what design method should be used for designing the controllers in the individual loops. We will consider μ-synthesis and parametric optimization.

μ synthesis is relatively fast, but it results in controllers with a high number of states. The number of states in controller element c_i will be equal to the number of states in the μ interconnection matrix for the design problem, plus the number of states in the D-scaling matrices that are used to scale the interconnection matrix. Controller element c_i will become a part of the interconnection matrix when designing subsequent loops, and the number of states will therefore accumulate. Model reduction for the reduction of the number of states in the controller is therefore necessary. In our experience, performance may suffer severely when the number of states in each controller element is reduced to a number normally considered acceptable for process control (typically three states or less).

Parameter optimization is relatively slow on the computer. The controller has to be parametrized à priori, e.g. using a PID structure, and the achievable control quality is dependent on the controller parametrization. However, the number of states in the controller is fixed, and no model reduction is necessary. In this work parametric optimization is therefore chosen for the design of the individual loops.

6.4.2 Iteration

Iteration (redesigning loops) is in general undesirable, both because it is time consuming and because one is no longer guaranteed the limited degree of failure tolerance normally associated with sequential design when one has to resort to iteration. One objective with our procedure is that the estimate of W_{1k} (using the estimate of $\hat{H}_k$) should reduce the need for iteration. However, it will of course be possible to reduce the value of μ further using iteration, but the improvement has been small for the examples we have considered.

6.5 Examples

The sequential design procedure outlined above will be demonstrated in two examples.

6.5.1 Example 2 (continued)

Consider again Example 2 from [3], and add the performance requirement $\bar{\sigma}(W_p S) < 1$. This should be satisfied for all possible plants allowed by the input uncertainty. We choose the performance weight

$$W_p(s) = w_p(s)I; \quad w_p(s) = 0.4\frac{\tau_{cl}s + 1}{\tau_{cl}}$$

The objective is to make the system as fast as possible in a robust sense, by minimizing τ_{cl} subject to $\mu_{RP} \leq 1$ (μ_{RP} meaning μ for robust performance).

We choose to pair on the diagonal elements of G as done previously. The PRGA for this example is shown in Fig. 7, together with the uncertainty weight. PRGA elements larger than 1 imply interactions, and the figure shows that there is severe interaction from loops 1 and 2 into loop 3. The loop gain in loop 3 must consequently be high at the frequencies where the feedback in loops 1 and 2 is effective. This means that the bandwidth in loop 3 has to be higher than the bandwidths in loops 1 and 2. The bandwidth in loop 3 will be limited by the input uncertainty. When attempting to minimize τ_{cl}, we must therefore minimize the difference in bandwidths between loop 3 and the two other loops. From Fig. 7 and Eq. (24) (with $\tilde{H}$ substituted for $\hat{H}_k$) we also see that we want $\tilde{h}_1$ and $\tilde{h}_2$ to roll off quickly at frequencies beyond their respective loop bandwidths, as this will reduce the interactions from loops 1 and 2 into loop 3 in the frequency range above the bandwidths of loops 1 and 2. The initial estimates for the complementary sensitivity functions for the individual loops are therefore chosen to be of the form

$$\tilde{h}_i(s) = \frac{1}{(\frac{s}{\omega_i} + 1)^2} \tag{26}$$

Estimating $\tilde{h}_i$ to be a first order low pass filter would imply that loops 1 and 2 would roll off more slowly beyond their respective bandwidths, and the interactions from loops 1 and 2 into loop 3 would therefore make it necessary to have a larger difference in bandwidths between loop 3 and loops 1 and 2. The approach taken here to minimize τ_{cl} may be considered somewhat naive: τ_{cl} is minimized (subject to $\mu \leq 1$) during

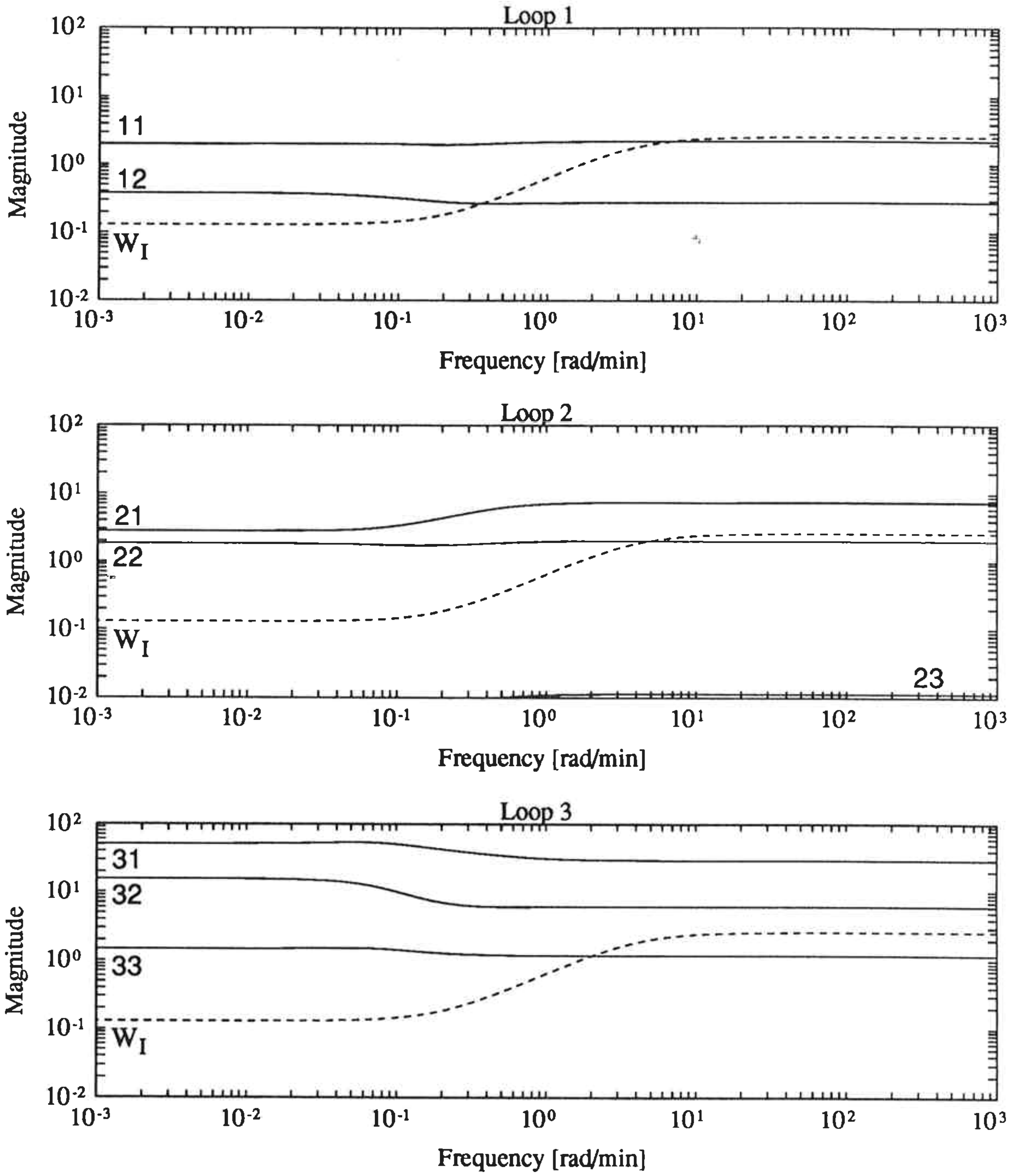

Figure 7: Performance relative gain array and uncertainty weight for Example 2. Element 13 is smaller than 10^{-2} at all frequencies.

the controller design for each loop. However, as the same τ_{cl} in the end will apply for all outputs, we choose ω_i in Eq. (26) to be consistent with τ_{cl}, that is, we specify a $\tilde{h}_i = 1 - \tilde{s}_i$ such that $|w_p(j\omega)\tilde{s}_i(j\omega)| < 1 \forall \omega$. We thus choose $\omega_1 = 1/\tau_{cl}$. From the PRGA's in Fig. 7 we see that $|\gamma_{31}| \approx 5|\gamma_{32}|$ at frequencies around 1 and assuming the magnitude of the loop gain in loop 3 to have a slope of -2 on a log-log plot this indicates $\omega_2/\omega_1 = 2.2$. Loop 3 is the fast loop, and considering the uncertainty weight we therefore fix $\omega_3 = 1rad/min$. It is clear from Fig. 7 that loop 3 must be closed first, and probably loop 2 second as there is some interaction from loop 1 to loop 2, especially at high frequencies.

The controller parametrization is chosen to be

$$c_i(s) = k\frac{T_1 s + 1}{T_1 s}\frac{T_2 s + 1}{10T_2 s + 1} \tag{27}$$

Note that this is not on the PID form since the pole in the last term is at a lower frequency ($s = 0.1/T_2$) than the zero. This controller parametrization allows loops 1 and 2 to roll off quickly beyond their respective bandwidths, whereas the loop gain in loop 3 can increase rapidly over one decade at frequencies slightly lower than the loop bandwidth.

Step 1: Loop 3. W_{11} is the third row of $(I + E_k\hat{H}_k)^{-1}$, and there is one 1×1 perturbation block for the input uncertainty, and one 3×1 perturbation block for the performance specification. Iterating on τ_{cl} (and changing ω_1 and ω_1 correspondingly, as explained above) $\mu = 0.992$ is obtained for $\tau_{cl} = 8.5$, and the corresponding controller is

$$c_3(s) = 84.9\frac{4.70s + 1}{4.70s}\frac{4.01s + 1}{40.1s + 1} \tag{28}$$

Step 2: Loop 2. $\tilde{h}_3$ is updated in $\hat{H}_k$, and W_{12} is the second and third rows of $(I + E_k\hat{H}_k)^{-1}$. There is one diagonal perturbation block of dimension 2×2 for the input uncertainty, and a 3×2 perturbation block for performance. $\mu = 0.998$ is obtained for $\tau_{cl} = 11$, and we find

$$c_2(s) = -0.079\frac{1.32s + 1}{1.32s}\frac{0.186s + 1}{1.86s + 1} \tag{29}$$

Step 3: Loop 1. Now all loops are included in the design problem (and there are no disturbances present), consequently $W_{13} = I_3$, and there is one diagonal 3×3 perturbation block for the input uncertainty and a full 3×3 perturbation block for performance. $\mu = 1.000$ is obtained for $\tau_{cl} = 18$ and

$$c_1(s) = 0.04\frac{0.385s + 1}{0.385s}\frac{0.898s + 1}{8.98s + 1} \tag{30}$$

In comparison, the best decentralized controller found using simultaneous parametric optimization with the same controller parametrization gave $\mu = 1$ for $\tau_{cl} = 16$.

6.5.2 Example 4

We consider the polypropylene reactor studied by Lie and Balchen [11, 12] and Hovd and Skogestad [9]. A schematic outline of the process is shown in Fig. 8. There are three inputs, three outputs and four disturbances. The process has a complex pair of RHP poles, but no RHP transmission zeros. However, there are RHP zeros in all

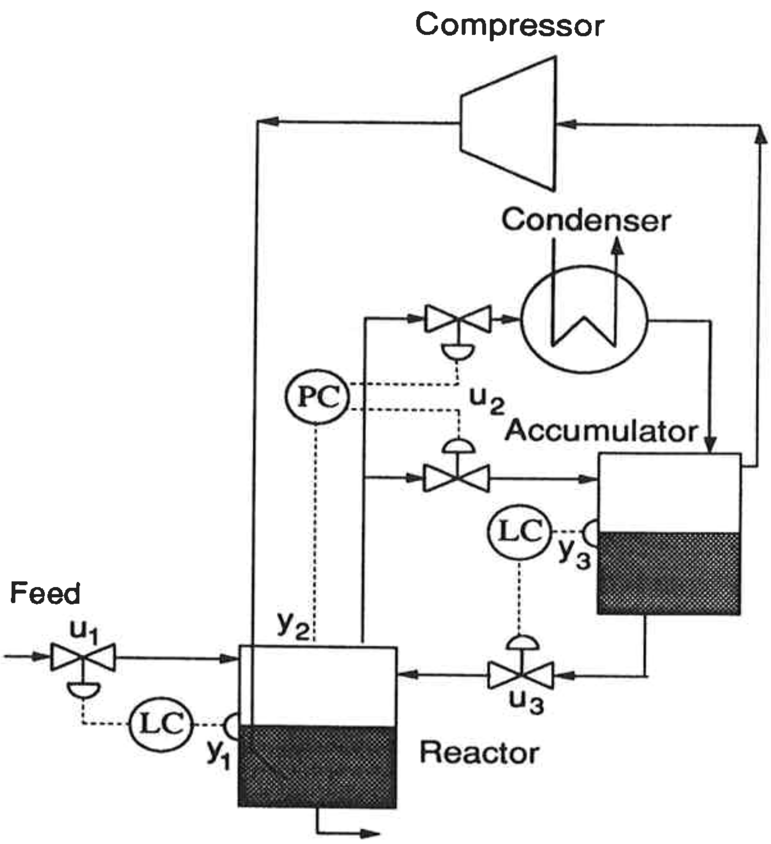

Figure 8: Schematic outline of the process in Example 4.

elements of G except in g_{11} at frequencies close to the frequency corresponding to the RHP poles. A more detailed description of the process and details on the scalings used are given in the appendix, together with a state space description of the process. Only input uncertainty is considered, and the uncertainty weight is

$$W_I(s) = w_I(s)I; \quad w_I(s) = 0.2\frac{\frac{1}{24}s+1}{\frac{1}{240}s+1}$$

which reflects a steady state uncertainty of 20% and a maximum neglected time delay of 0.5 minute. The performance requirement, in terms of the scaled outputs and disturbances, is $\bar{\sigma}(W_pS[I\ G_d]) < 1$ at all frequencies and for all uncertainties allowed by the uncertainty weight. The performance weight is given by

$$W_p(s) = w_p(s)I; \quad w_p(s) = 0.4\frac{0.2s+1}{0.2s}$$

We will use decentralized control with pairings as indicated in Fig. 8. This pairing was found to be preferable in [11, 9], and corresponds to industrial practice. The PRGA and CLDG for this process are shown in Fig. 9 and Fig. 10 respectively, together with the uncertainty and performance weights. Fig. 9 shows that the interaction is relatively modest in the bandwidth region, except for loop 3 where the interaction between the loops causes increased loop gain requirement at low frequencies. From Fig. 10 we see that loop 3 is also most severely affected by the disturbances. Loop

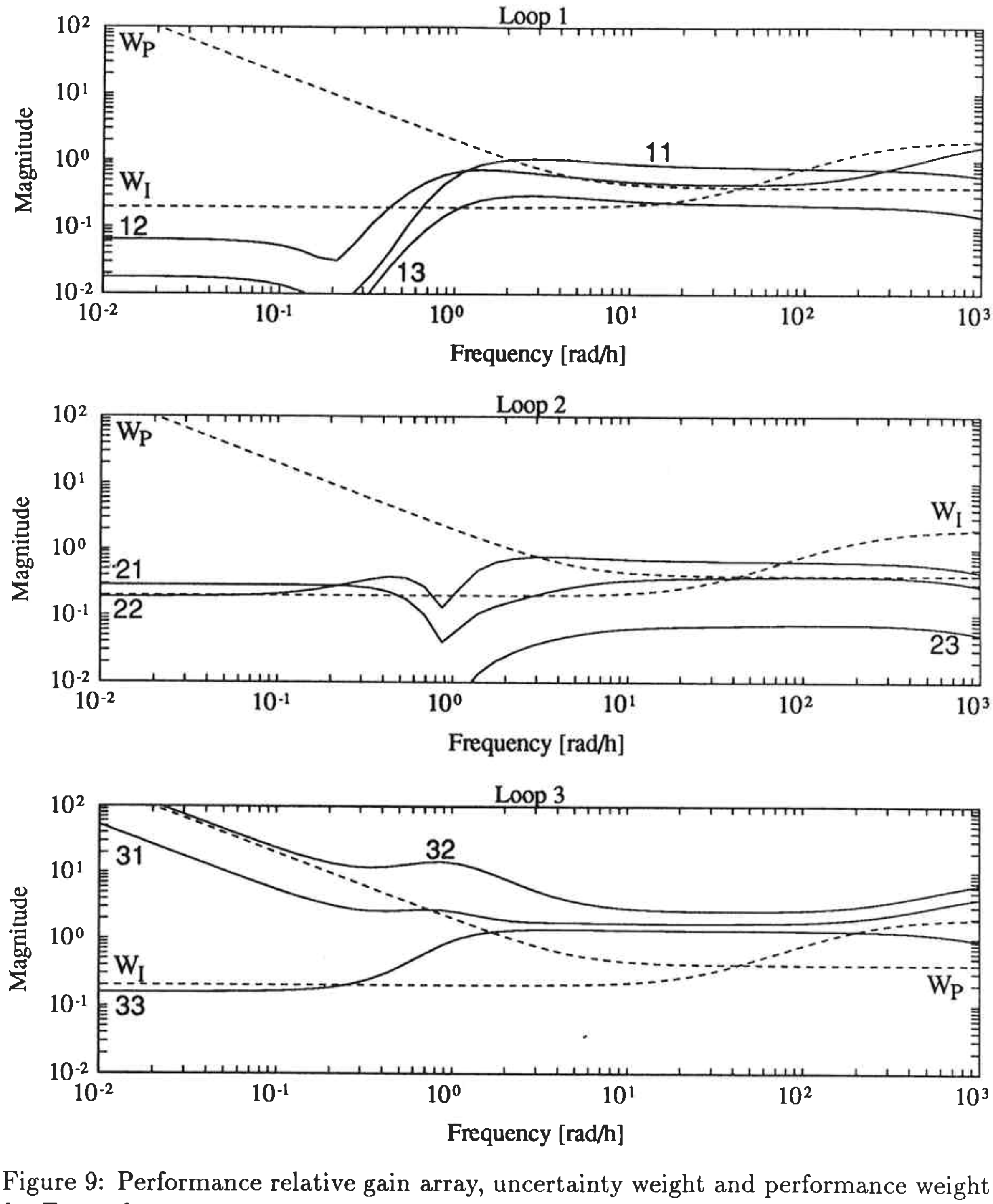

Figure 9: Performance relative gain array, uncertainty weight and performance weight for Example 4.

3 must therefore be the fastest loop, and loop 1 is chosen as the second fastest loop as it is somewhat more affected by disturbances than loop 2 at frequencies around 2 [rad/min]. The uncertainty and performance weights give upper and lower bounds for the bandwidths of the individual loops. We choose to spread out the bandwidths of the individual loops between these bounds, and choose the following initial estimated

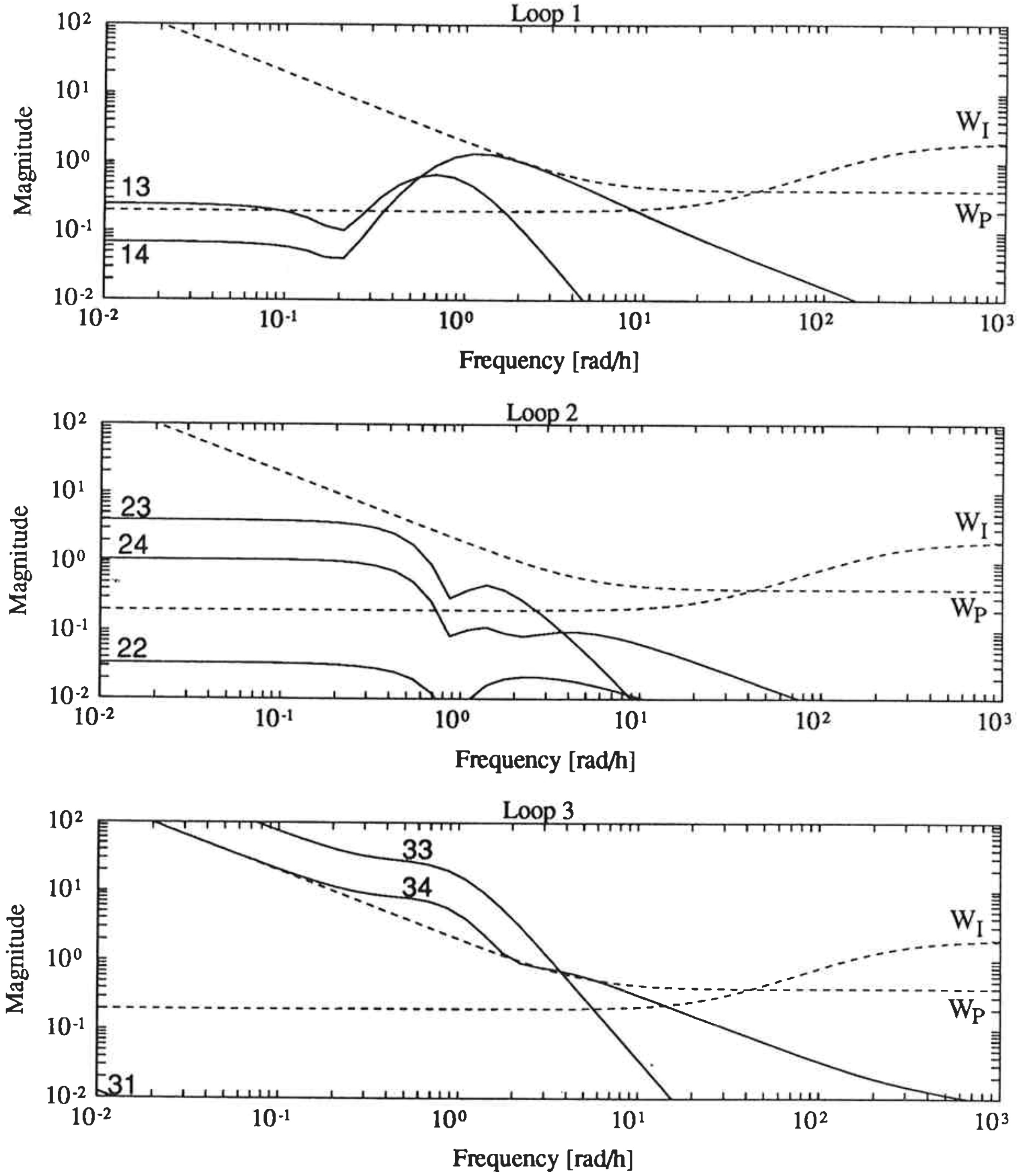

Figure 10: Closed loop disturbance gains, uncertainty weight and performance weight for Example 4. Elements of the CLDG not shown are smaller than 10^{-2} at all frequencies.

for the complementary sensitivity functions for the individual loops:

$$\tilde{h}_1 = \frac{1}{0.04s+1}; \quad \tilde{h}_2 = \frac{1}{0.2s+1}; \quad \tilde{h}_3 = \frac{1}{0.015s+1}$$

PI controllers, $c_i(s) = k\frac{T_i s+1}{T_i s}$ are used for all loops. Because of RHP zeros in individual elements and subsystems, the design sequence for this example must be 1-2-3, i.e., we are required to design the fastest loop last. In contrast to Example 2, the performance weight for this example is fixed, and we therefore minimize μ at each stage in the design.

Step 1: Loop 1. W_{11} is the first row of $(I + E_2\tilde{H})^{-1}[I\ G_d]$, and there is one 1×1 perturbation block for the input uncertainty and a 7×1 perturbation block for performance. The minimum value of μ found is 0.6363, and the corresponding (unscaled) controller $c_1(s) = 2.31 \cdot 10^5 \cdot \frac{0.962s+1}{0.962s}$.

Step 2: Loop 2. $\hat{H}_2$ is updated, using the controller found for loop 1. W_{12} consists of the first two rows of $(I + E_2\hat{H}_2)^{-1}[I\ G_d]$, there is one diagonal 2×2 perturbation block for the input uncertainty and a full 7×2 perturbation block for performance. The minimum value of μ found is 1.09, and the corresponding controller is $c_2(s) = 4.99 \cdot 10^{-2} \cdot \frac{0.148s+1}{0.148s}$.

Step 3: Loop 3. Now all loops are included in the design problem, consequently $\hat{G}_3 = G$ and $W_1 = [I\ G_d]$. There is a diagonal 3×3 perturbation block for the input uncertainty and a full 7×3 perturbation block for performance. The minimum value of μ found is 0.89, and the corresponding controller is $c_3(s) = -1.27 \cdot 10^5 \cdot \frac{0.131s+1}{0.131s}$.

In comparison, with the best decentralized PI controller found using simultaneous parametric optimization μ improved only marginally, from 0.89 to 0.86. The best multivariable controller found using μ-synthesis gave $\mu = 0.65$.

The fact that a lower value of μ is achieved after closing all loops that when closing loops 1 and 2 only, illustrates that the design problem may become easier as more loops are brought into service, and that the estimates of W_{1k} used may be conservative. However, it appears that the controller found using sequential design is relatively close to the optimal for a decentralized PI controller.

6.6 Discussion on Sequential Design

1. Sequential design using parameter optimization consists of much smaller optimization problems than designing all loops simultaneously by parameter optimization, and designing one loop at the time is therefore preferable. Also, parameter optimization for all loops simultaneously does not guarantee the limited degree of failure tolerance that is associated with sequential design. For the examples studied in this paper, the sequential design procedure presented in this paper achieved a quality of control that is not significantly poorer than that achieved using parameter optimization for all loops simultaneously.

2. The idea of using a simplified estimate of the effect of closing the other loops is not new. Balchen and Mummé ([1], Appendix C) derive an estimate the transfer function between input u_i and output y_i when all the other loops are closed, using an estimate of the complementary sensitivity function for the other loops. In [1] this is used to find pairings for decentralized control.

Other uses can also be considered using the estimated transfer function from u_i to y_i when the other loops are closed. For example, one may use the Ziegler-Nichols tuning rules using this estimate of the transfer function. Since the Ziegler-Nichols tuning rules are very simple, the loops can be redesigned with little effort, thus reducing the problem that the initial estimate of the complementary sensitivity function for the other loops may well be poor in the bandwidth region.

3. It is easier to estimate the complementary sensitivity function for the individual loops than to estimate the controller in the individual loops. This holds especially at low frequency, where control is almost perfect, and we know that $\tilde{h}_i \approx 1$.

4. The idea of using an estimate of the effect of closing the other loops is not confined to the H_∞ or μ framework, it may also be used with other norms, e.g. H_2.

5. Many multivariable controllers consist of simple pre- and/or post-compensator and have the main dynamics in a diagonal matrix. The compensators are often designed to counteract interactions at a given frequency (e.g. steady-state decoupling), and there will be interactions at other frequencies. Both the independent design and the sequential design procedures in this paper may be used for designing the diagonal matrix of such multivariable controllers. Note that decentralized controllers are known to be relatively robust (but the performance may be poor even nominally), when using non-diagonal compensators the issue of robustness is more important [18].

6.7 Conclusions on Sequential Design

Sequential design should not be based on independent design, but rather on considering one loop at the time.

We propose a sequential design procedure, starting from the individual loops. It is shown that including the appropriate rows of $(I + E_k\hat{H}_k)^{-1}[I \ G_d]$, using an estimate of $\hat{H}_k$, has the effect of including an estimate of the effect that the loops that are still open will have on the closed loops and the loop that is closed at step k in the procedure.

Acknowledgement The authors wish to tank M. P. Newlin and P. M. Young of the California Institute of Technology for access to their software for calculating μ with mixed real and complex perturbations.

References

[1] Balchen J. G. and Mummé, K. I. (1988). *Process Control. Structures and Applications*, Van Nostrand Reinhold, New York, USA.

[2] Bernstein, D. S. (1987). Sequential Design of Decentralized Dynamic Compensators Using the Optimal Projection Equations. *Int. J. Control*, **46**, pp. 1569-1577.

[3] Chiu, M.-S. and Arkun, Y. (1991). A Methodology for Sequential Design of Robust Decentralized Control Systems. Submitted to *Automatica*, preprint.

[4] Chiu, M.-S. and Arkun, Y. (1991). A Methodology for Sequential Design of Robust Decentralized Control Systems. *Proc. ACC*, Boston, Massachusetts, pp. 1836-1841.

[5] Doyle, J. C., Wall, J. E. and Stein, G. (1982). Performance and Robustness Analysis for Structured Uncertainty. *Proc. IEEE Conf. Decision Contr.*, Orlando, FL., pp. 629-636.

[6] Doyle, J. C. and Chu, C.-C. (1985). Matrix Interpolation and H_∞ Performance Bounds. *Proc. ACC*, Boston, MA, pp. 129-134.

[7] Garcia, C. and Morari, M (1985). Internal Model Control - 2. Design Procedure for Multivariable Systems. *Ind. Eng. Chem. Proc. Des. & Dev.*, **24**, pp. 472-484.

[8] Hovd, M. and Skogestad, S. (1992). Simple Frequency-Dependent Tools for Control System Analysis, Structure Selection and Design. To appear in *Automatica.*

[9] Hovd, M. and Skogestad, S. (1992). Controllability Analysis for Unstable Processes. *Proc. IFAC Workshop on Interactions Between Process Design and Process Control*, London, England, September 1992.

[10] Lee, J. H. and Morari, M. (1991). Robust Measurement Selection. *Automatica*, **27**, 3, pp. 519-527.

[11] Lie, B. (1990). Control Structures for Polymerization Processes Applied to Polypropylene Manufacturing. *Dr. Ing. Thesis*, University of Trondheim-NTH.

[12] Lie, B. and Balchen, J. G. (1992). A Comparison of Strategies for Control of a Polypropylene Reactor, *Preprints IFAC Symposium DYCORD+'92*, College Park, Maryland, Apr. 1992, 265-270.

[13] Maciejowski, J. M. (1989). *Multivariable Feedback Design.* Addison-Wesley, Wokingham, England.

[14] Mayne, D. Q. (1973). The Design of Linear Multivariable Systems. *Automatica*, **9**, pp. 201-207.

[15] Morari, M. and Zafiriou, E., (1989). *Robust Process Control*, Prentice-Hall, Englewood Cliffs, New Jersey.

[16] Nett, C. N. and Uthgenannt, (1988). An Explicit Formula and An Optimal Weight for the 2-Block Structured Singular Value Interaction Measure. *Automatica*, **24**, pp. 261-265.

[17] Niederlinski, A. (1971). A Heuristic Approach to the Design of Linear Multivariable Interacting Control Systems. *Automatica*, **7**, pp. 691-701.

[18] Skogestad, S. and Morari, M. (1987). Implications of Large RGA-Elements on Control Performance, *Ind. Eng. Chem. Res.*, **26**, 11, pp. 2323-2330.

[19] Skogestad, S. and Morari, M. (1988). Some New Properties of the Structured Singular Value. *IEEE Trans. Autom. Control*, **33**, 12, pp. 1151-1154.

[20] Skogestad, S. and Morari, M. (1989). Robust Performance of Decentralized Control Systems by Independent Design. *Automatica*, **25**, 1, pp.119-125.

[21] Skogestad, S. and Lundström, P. (1990). Mu-optimal LV-control of Distillation Columns. *Computers chem. Engng.*, **14**, 4/5, pp. 401-413.

[22] Skogestad, S. and Hovd, M., (1990) Use of Frequency-dependent RGA for Control Structure Selection. *Proc. American Control Conference*, San Diego, CA., pp. 2133-2139.

[23] Viswanadham, N. and Taylor, J. H. (1988). Sequential Design of Large-Scale Decentralized Control Systems. *Int. J. Control*, **47**, pp. 257-279.

[24] Young, P. M., Newlin, M. P. and Doyle, J. C. (1991). μ Analysis with Real Parametric Uncertainty. *Proc. 30th CDC*, Brighton, England, pp. 1251-1256.

Appendix. Description of the Process in Example 4

The monomer feed enters into a stirred tank reactor containing a slurry of monomer, catalyst, cocatalyst, polymer and some impurities. The reaction is exothermic, causing some of the slurry components to vaporize. The vapor leaving the reactor is transferred to an accumulator vessel. Heat is removed from the system by condensing parts of the vapor leaving the reactor, before it enters the accumulator. Heat removal is adjusted by adjusting a split range valve which determines what fraction of the vapor leaving the reactor is passed through the condenser. The liquid in the accumulator is returned to the reactor, and the vapor from the accumulator is compressed and bubbled through the reactor slurry. This results in a 3×3 plant model $G(s)$ with seven states. The inputs and outputs are

y_1 - reactor slurry level $(0 - 1)$
y_2 - reactor pressure (gauge pressure in atmospheres)
y_3 - accumulator liquid level $(0 - 1)$.
u_1 - monomer feed flowrate (kg/h).
u_2 - split range valve position $(0 - 1)$.
u_3 - accumulator to reactor liquid flowrate (kg/h).

Four disturbances are considered:

d_1 - monomer feed temperature $(°C)$
d_2 - cooling water flowrate through heat exchanger (kg/h)
d_3 - catalyst mass feed flowrate (kg/h)
d_4 - recycle flow of unreacted monomer (kg/h)

We scale the outputs such that a magnitude of 1 for the scaled outputs correspond to offsets: $y_1 = 0.05$, $y_2 = 1.0atm$ and $y_3 = 0.10$. Likewise, the disturbances are scaled such that a magnitude of 1 for the scaled disturbances correspond to: $d_1 = 20°C$, $d_2 = 10000kg/h$, $d_3 = 3kg/h$ and $d_4 = 1000kg/h$.

State space description of the plant in Example 4.

The description is given as $y(s) = G(s)u(s) + G_d(s)d(s) = [C(sI - A)^{-1}B + D]u(s) + [C(sI - A)^{-1}B_d + D_d]d(s)$.

$$A = \begin{bmatrix} 0 & 0 & 0 & -3.08e+03 & 0 & 3.17e+03 & 1.00e+00 \\ 4.30e-04 & -4.71e-01 & 0 & 0 & 0 & 0 & 0 \\ 3.49e-01 & 0 & -4.71e-01 & 0 & 0 & 0 & -1.00e+00 \\ 0 & 0 & 0 & -7.50e+00 & 0 & 8.49e+00 & -1.96e-02 \\ 0 & 0 & 0 & 3.08e+03 & 0 & -3.17e+03 & 0 \\ 0 & 0 & 0 & 3.57e+01 & 0 & -6.77e+01 & 0 \\ 0 & -3.81e+02 & 0 & -3.53e+02 & 0 & 0 & -1.00e+00 \end{bmatrix}$$

$$B = \begin{bmatrix} 1.00e+00 & 0 & 1.00e+00 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ -1.45e-03 & 0 & -4.79e-04 \\ 0 & 0 & -1.00e+00 \\ 0 & 2.73e+03 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$B_d = \begin{bmatrix} 0 & 0 & 0 & -1.00e+00 \\ 0 & 0 & 1.00e+00 & -9.15e-04 \\ 0 & 0 & 0 & -7.40e-01 \\ 2.99e-05 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & -2.02e-04 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 7.49e-05 & 0 & 3.29e-05 & 1.23e-02 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4.86e-01 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 5.39e-04 & 5.75e-03 & 0 \end{bmatrix}$$

$$D = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}; \quad D_d = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$