Chapter 8 Decoupling and Sparse Controls

8.1 Classical Decoupling Control

8.1.1 Motivation

The objective in this part is to present some of the principles and techniques for designing multivariable controllers, motivated by the fact that the design of effective multivariable controllers remains one of the most challenging problems faced by industrial process control practitioners. Here, we will assume that the simple approach of good loop pairing and appropriate tuning of multiple single-loop controllers has not produced a control system that meets the requirements of the process. Thus, it is necessary to design a true multivariable controller that acts on data from all the process outputs and makes adjustments to all the process inputs in response to a disturbance or set-point change. To make this more explicit, let us adopt the notation introduced in before and assume that after input/output pairing has been carried out, the input variables become u_i, and that the system has been configured with the y_1 — u_1 , y_2 — u_2,…, y_n — u_n, pairings. Under the multiple, independent, single-loop control strategy, each controller g_{ci} operates according to:

$$u_i = g_{ci}(y_{di} - y_i) = g_{ci}\varepsilon_i$$

On the other hand, a true multivariable controller must decide on u_i, not using only ε_i, but using the entire set, ε_1, ε_2,, **...,**ε_n,**;.** Thus, the controller actions are obtained from:

$$\begin{aligned}
u_1 &= f_1(\varepsilon_1, \varepsilon_2, \cdots \varepsilon_n) \\
u_2 &= f_2(\varepsilon_1, \varepsilon_2, \cdots \varepsilon_n) \\
u_3 &= f_3(\varepsilon_1, \varepsilon_2, \cdots \varepsilon_n) \\
\cdots &= \cdots \\
u_n &= f_n(\varepsilon_1, \varepsilon_2, \cdots \varepsilon_n)
\end{aligned}$$

The design problem is therefore that of determining $f_1(.)$, $f_2(.)$,..., $f_n(.)$ so that each of the output errors is driven to zero in an acceptable fashion; this often reduces to designing the controller such that the interaction effects are eliminated, or at least significantly reduced. In a very general sense, the principle behind how the functions indicated in Eq. (8.1.1) are determined is what distinguishes one multivariable control technique from another.

By using transformations to create combinations of the original input and/or output variables of a process experiencing strong interactions, it is possible to obtain an equivalent system whose variables no longer interact; for example, consider a 2×2 system whose transfer function model is given as:

$$\begin{aligned}
y_1 &= a(s)u_1 + a(s)u_2 \\
y_2 &= b(s)u_1 - b(s)u_2
\end{aligned}$$

Observe now that the process variables of this system interact with one another; but by defining two new variables μ_1 and μ_2 according to:

$$\begin{aligned}
\mu_1 &= u_1 + u_2 \\
\mu_2 &= u_1 - u_2
\end{aligned}$$

the modeling equations become

$$y_1 = a(s)\mu_1$$
$$y_2 = b(s)\mu_1$$

and μ_1 does not interact with y_2 neither does μ_2 interact with y_1. The implication now is that independent controllers can be designed for the transformed system, the control equations will now be:

$$\mu_1 = g_{c1}\varepsilon_1$$
$$\mu_2 = g_{c2}\varepsilon_2$$

Note however, that the new variables μ_1 and μ_2 are not the real process variables; we must now recover u_1 and u_2, the actual process input variables, in terms of μ_1 and μ_2. The result of the reverse transformation is:

$$u_1 = \frac{\mu_1 + \mu_2}{2}$$
$$u_2 = \frac{\mu_1 - \mu_2}{2}$$

we obtain the actual control equations which, on the basis of the indicated variable transformation, will eliminate interactions from this process:

$$u_1 = \frac{g_{c1}\varepsilon_1 + g_{c2}\varepsilon_2}{2}$$
$$u_2 = \frac{g_{c1}\varepsilon_1 - g_{c2}\varepsilon_2}{2}$$

Finding the appropriate combination of process variables that lead to total elimination of, or, at least a reduction in, the control loop interactions, tends in general to be an “art”.

From our introduction to multivariable system, we know that all the input variables of a typical multivariable process are *coupled* to all its output variables; the main the y_1 — u_1 , y_2 — u_2,…, y_n — u_n, couplings are desirable (this is what makes it possible to control the process variable in the first instance); it is the y_i — u_j, cross-couplings, by which y_i is influenced by u_j, (for all i and all j, with $i \neq j$), that are undesirable: they are responsible for the control loop interactions.

It is therefore clear that any technique that eliminates the *effect* of the undesired cross-couplings will improve control system performance. Note that the purpose is *not* to “eliminate” the cross-couplings; that is an impossibility, since to do this will require altering the physical nature of the system. Observe, for example, that it is not possible to stop the hot stream flowrate from affecting the tank temperature in the stirred mixing tank system, despite the fact that our main desire is to use this flowrate to control the level; neither can we prevent the cold stream from affecting the level, even though controlling the tank temperature is its main responsibility. We can, however, *compensate* for the noted cross-coupling effects by carefully balancing the hot and cold stream flowrates.

The main objective in decoupling is to *compensate* for the effect of interactions brought about by cross-couplings of the process variables. This is to be achieved by introducing an additional transfer function “block” (the interaction compensator) between the single-loop controllers, and the process. This interaction compensator, together with the single-loop controllers, now constitute the multivariable decoupling controller.

In the ideal case, the decoupler causes the control loops to act as if totally independent of one another, thereby reducing the controller tuning task to that of tuning several, noninteracting controllers. The main advantage here is that it will now be possible to use SISO controller design techniques.

The decoupler design problem is that of choosing the elements of the compensator to satisfy usually one of the following objectives:

1. *Dynamic Decoupling:* eliminate interactions from *all* loops, at every instant of time.
2. *Steady-State Decoupling:* eliminate only steady-state interactions from all loops; because dynamic interactions are tolerated, this objective is less ambitious than dynamic decoupling but leads to much simpler decoupler designs.
3. *Partial Decoupling:* eliminate interactions (dynamic, or only steady state) in a subset of the control loops; this objective focuses attention on only the critical loops in which the interactions are strongest, leaving those with weak interactions to act without the aid of decoupling.

Having thus illustrated the general principles and concepts of a multivariable controller, we now proceed to discuss some of the multivariable controller design techniques in greater detail. In each case we will focus attention on three things:

1. *Principles* behind the technique
2. *Results*
3. *Applications,* including practical considerations for effective utilization

8.1.2 **Simplified Decoupling**

The principles of ideal decoupling, is illustrated in Fig. 8.1.1 by using a 2×2 example. Additional transfer function blocks g_{I1} and g_{I2} are introduced between the single-loop controllers and the process, functioning as links between the otherwise independent controllers. The controller outputs are designated as v_1 and v_2, while the actual control on the process remains as u_1 and u_2. With the interaction compensator, Loop 2 is "informed" of changes in v_1 through g_{12}, so that u_2 is adjusted accordingly. The same task is performed for Loop 1 by g_{11}, which adjusts u_1 with v_2 information. Consequently, the actual control action experienced by the process will therefore now contain information from all the other controllers.

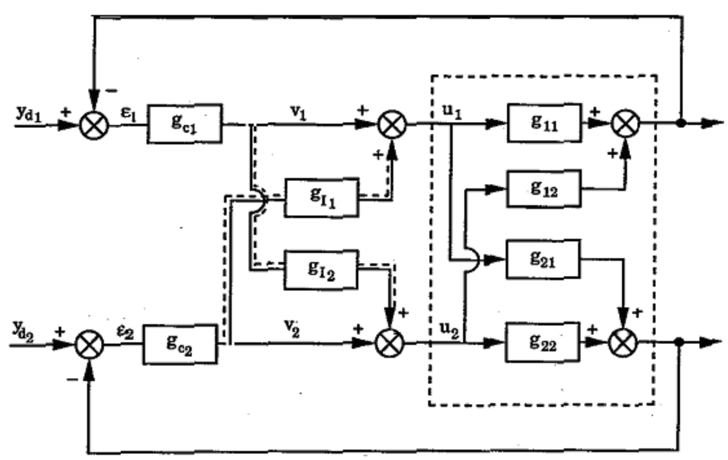


Fig. 8.1.1 Multivariable 2 x 2 control system incorporating simplified decoupling

The design procedures are as follows:

For the process transfer functions

$$\begin{aligned} y_1 &= g_{11}u_1 + g_{12}u_2 \\ y_2 &= g_{21}u_1 + g_{22}u_2 \end{aligned} \tag{8.1.1}$$

the equations governing the control action are:

$$\begin{aligned} u_1 &= v_1 + g_{I1}v_2 \\ u_2 &= v_2 + g_{I2}v_1 \end{aligned} \tag{8.1.2}$$

It gives:

$$\begin{aligned} y_1 &= (g_{11} + g_{12}g_{I2})v_1 + (g_{12} + g_{11}g_{I1})v_2 \\ y_2 &= (g_{21} + g_{22}g_{I2})v_1 + (g_{22} + g_{21}g_{I1})v_2 \end{aligned} \tag{8.1.3}$$

To have only v_1 affect y_1 and eliminate the effect of v_2

$$g_{12} + g_{11}g_{I1} = 0$$

which requires:

$$g_{I1} = -\frac{g_{12}}{g_{11}} \tag{8.1.4a}$$

To make y_2 free of v_1 requires:

$$g_{I2} = -\frac{g_{21}}{g_{22}} \tag{8.1.4a}$$

The overall system equations become:

$$\begin{aligned} y_1 &= (g_{11} - \frac{g_{12}g_{21}}{g_{22}})v_1 \\ y_2 &= (g_{22} - \frac{g_{12}g_{21}}{g_{11}})v_2 \end{aligned} \tag{8.1.5}$$

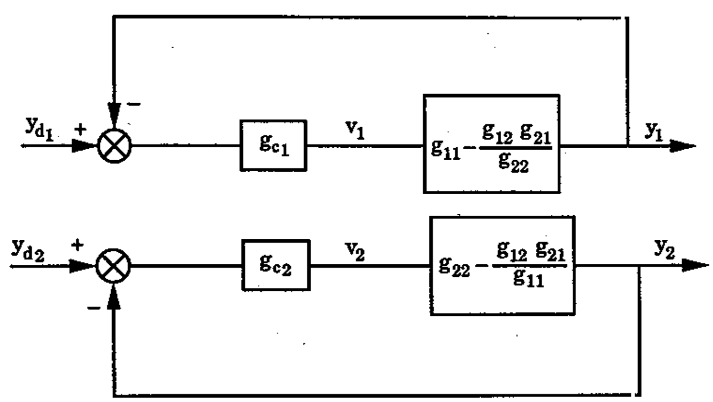


Fig 8.1.1 Decoupled TITO Process

the system is completely decoupled, with v_1 affecting only y_1, and v_2 affecting only y_2, each can be tuned individually.

8.1.2 Generalized Decoupling

The diagonal decoupling control system is depicted as in Figure 8.1.2, where $G(s)$, $G_I(s)$ and $G_c(s)$ are *n*- dimensional process, decoupler and controller transfer function matrices, respectively. The introduction of the additional transfer function block $G_I(s)$ between the single-loop controllers $G_c(s)$ and the process $G(s)$, is to act upon the process $G(s)$ such that the transfer function matrix

$$G_R(s) = G(s)G_I(s) \tag{8.1.6}$$

looking from the controller output is diagonal. In the ideal case, the decoupler causes the control loops to act as if they are totally independent of one another, thereby reducing the controller tuning task to that of tuning several, non-interacting controllers.

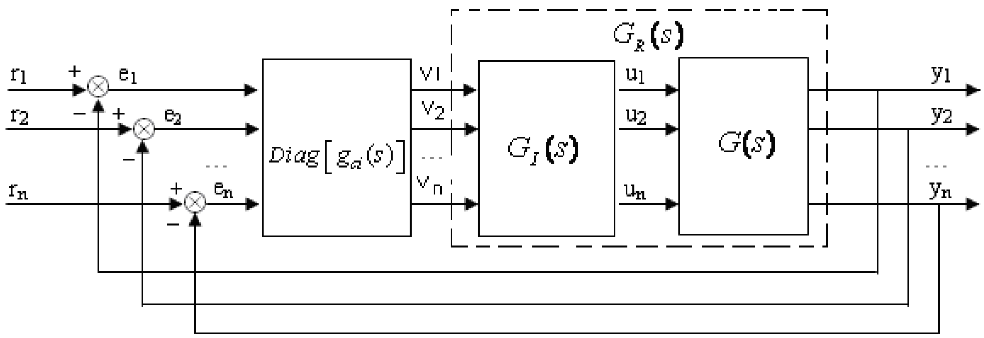


Figure 8.1.2 Block Diagram of Decoupling control system

In the ideal diagonal decoupling control scheme, the decoupler can be easily obtained by

$$G_I = G^{-1}G_R(s) \tag{8.1.7}$$

so that:

$$\begin{aligned} y &= G(s)G_I(s)v \\ &= G_R(s)v \end{aligned} \tag{8.1.8}$$

where $G_R(s)$. represents the equivalent diagonal process. The elements of G_R should be selected to provide the desired decoupled behavior with the simplest possible decoupler. A commonly employed choice is:

$$G_R(s) = Diag[G(s)] \tag{8.1.9}$$

other choices may also be used.

8.1.3 Comparison of the two Decoupling Schemes

For simplified decoupling 2 x 2 system, the compensator transfer function matrix is given by:

$$G_I = \begin{bmatrix} 1 & g_{I1} \\ g_{I2} & 1 \end{bmatrix}$$

for a 3 x 3 system, the compensator matrix G_I is:

$$G_I = \begin{bmatrix} 1 & g_{I12} & g_{I13} \\ g_{I21} & 1 & g_{I23} \\ g_{I31} & g_{I32} & 1 \end{bmatrix}$$

For the desired G_I, task is to find the g_{Iij} $(i \neq j)$ to make GG_I diagonal.

The general decoupling final diagonal form for GG_I is specified as G_R, then the G_I can be derived.

Example 8.1.1 *Distillation Column:* The transfer function model is given as:

$$G(s) = \begin{bmatrix} \dfrac{12.8e^{-s}}{16.7s+1} & \dfrac{-18.9e^{-3s}}{21.0s+1} \\ \dfrac{6.6e^{-7s}}{10.9s+1} & \dfrac{-19.4e^{-3s}}{14.4s+1} \end{bmatrix}$$

Simplified decoupling:

$$g_{I1} = -\frac{\dfrac{-18.9e^{-3s}}{21.0s+1}}{\dfrac{12.8e^{-s}}{16.7s+1}} \qquad g_{I2} = -\frac{\dfrac{6.67e^{-7s}}{10.9s+1}}{\dfrac{-19.4e^{-3s}}{14.4s+1}}$$

which simplify to:

$$g_{I1} = 1.48\frac{(16.7s+1)e^{-2s}}{21.0s+1} \qquad g_{I2} = 0.34\frac{(14.4s+1)e^{-4s}}{10.9s+1}$$

Each has the form of a lead/lag system with a gain term and a time-delay term.

In terms of actual implementation

$$u_1 = v_1 + 1.48\frac{(16.7s+1)e^{-2s}}{21.0s+1}v_2\,, \qquad u_2 = v_2 + 0.34\frac{(14.4s+1)e^{-4s}}{10.9s+1}v_1$$

The “equivalent” open-loop decoupled system as:

$$y_1 = \left(g_{11} - \frac{g_{12}g_{21}}{g_{22}}\right)v_1 = \left(\frac{12.8e^{-s}}{(16.7s+1)} - \frac{18.9\times 6.6(14.4s+1)e^{-7s}}{19.4(21.0s+1)(10.9s+1)}\right)v_1$$

$$y_2 = \left(g_{22} - \frac{g_{12}g_{21}}{g_{11}}\right)v_2 = \left(\frac{-19.4e^{-3s}}{(14.4s+1)} - \frac{18.9\times 6.6(16.7s+1)e^{-9s}}{12.8(21.0s+1)(10.9s+1)}\right)v_2$$

Generalized decoupling:

If we choose G_R to be diagonal elements, then:

$$G_R(s) = \begin{bmatrix} \dfrac{12.8e^{-s}}{16.7s+1} & 0 \\ 0 & \dfrac{-19.4e^{-3s}}{14.4s+1} \end{bmatrix}$$

To obtain the required decoupler, evaluate the inverse of the transfer function matrix which is given by:

$$G^{-1}(s) = \frac{1}{\Delta}\begin{bmatrix} \dfrac{-19.4e^{-3s}}{14.4s+1} & \dfrac{18.9e^{-3s}}{21.0s+1} \\ \dfrac{-6.67e^{-7s}}{10.9s+1} & \dfrac{12.8e^{-s}}{16.7s+1} \end{bmatrix}$$

where Δ is the determinant of the matrix:

$$\Delta = \frac{-248.32(21.0s+1)(10.9s+1)e^{-4s} + 124.74(16.7s+1)(14.4s+1)e^{-10s}}{(21.0s+1)(10.9s+1)(16.7s+1)(14.4s+1)}$$

By multiplying the inverse matrix and G_R, we have:

$$G_I = \begin{bmatrix} g_{I11} & g_{I12} \\ g_{I21} & g_{I22} \end{bmatrix}$$

with each element given as follows:

$$g_{I11} = \frac{-248.32(21.0s+1)(10.9s+1)}{124.74(16.7s+1)(14.4s+1)e^{-6s} - 248.32(21.0s+1)(10.9s+1)}$$

$$g_{I12} = \frac{-366.66(16.7s+1)(10.9s+1)e^{-2s}}{124.74(16.7s+1)(14.4s+1)e^{-6s} - 248.32(21.0s+1)(10.9s+1)}$$

$$g_{I21} = \frac{84.48(21.0s+1)(14.4s+1)}{124.74(16.7s+1)(14.4s+1)e^{-6s} - 248.32(21.0s+1)(10.9s+1)}$$

and

$$g_{I22} = g_{I11}$$

The actual implementation:

$$u_1 = g_{I11}v_1 + g_{I12}v_2$$

$$u_2 = g_{I21}v_1 + g_{I22}v_2$$

Advantages and disadvantages of each method:

1. For simplified decoupling, the “equivalent” open-loop decoupled system is much more complicated than the G_R matrix specified in the *Generalized* decoupling design. Therefore, the tuning and closed-loop performance for the *generalized* decoupling is much better than for *Simplified* decoupling.

2. For generalized decoupling, the tuning and closed-loop performance for the *generalized* decoupling is much simpler than for *Simplified* decoupling. However, the price for this improvement is the complicated decoupler.

The ideal decouplers by the simplified decoupling approach are similar to feedforward controllers. Therefore, ideal dynamic decouplers are subject to the realization problems, particularly when time delays are involved in the transfer function elements. While generalized decouplers are based on model inverses. As such, they can only be implemented if such inverses are both *causal* and *stable*.

8.1.4 Conditions for decouplers

General conditions for implementing simplified dynamic decoupling for $N \times N$ systems:

1. Causality: Consider the 2 x 2 compensators, the transfer functions g_{I1} and g_{I2} must be causal (no $e^{+\alpha s}$ terms) and stable. To satisfy causality for the 2 x 2 system, any time delays in g_{11} must be smaller than the time delays in g_{12}, and a similar condition must hold for g_{22} and g_{21}. In $G(s)$, the smallest delay in each row must occur on the diagonal. For simplified decoupling, this is an absolute requirement. If the original process G does not comply, we can add delays to the inputs $u_1, u_2, ..., u_n$, by define:

$$G_m = GD \tag{8.1.10}$$

   where D is a diagonal matrix of time delays:

$$D(s) = \begin{bmatrix} e^{-d_{11}s} & & & 0 \\ & e^{-d_{22}s} & & \\ & & \ddots & \\ 0 & & & e^{-d_{nn}s} \end{bmatrix} \tag{8.1.11}$$

   The simplified decoupler is then designed by using the elements of G_m rather than G, and the matrix D must be inserted into the control loop. For generalized decoupling, one may use modified process G_m as, or adjust the time delays in the diagonal matrix, G_R, so that the elements of $G_I=(GD)^{-1}G_R$ are causal. This is to requiring that $G_R^{-1}(GD)$ have the smallest delay in each row on the diagonal.

2. Stability: In order to ensure the stability of the compensator transfer functions, there be no RHP zeros of the process $G(s)$. For simplified decoupling, g_{11} and g_{22} must not have any RHP zeros and also g_{12} and g_{21} must not have any RHP poles. and that the off-diagonal elements of G be stable. For generalized decoupling this may be relaxed by adjusting the dynamics of G_R in order that the elements of $G_I=G^{-1}G_R$ be stable.

Example 8.1.2 *Distillation Column:* Consider a slight modification of the Distillation Column process where a process change has added a time delay of 3 minutes to the input u_1:

$$G(s) = \begin{bmatrix} \dfrac{12.8e^{-4s}}{16.7s+1} & \dfrac{-18.9e^{-3s}}{21.0s+1} \\ \dfrac{6.67e^{-10s}}{10.9s+1} & \dfrac{-19.4e^{-3s}}{14.4s+1} \end{bmatrix}$$

Note that the smallest delay in each row is not on the diagonal, the simplified decoupling compensator becomes:

$$g_{I1} = 1.48\frac{(16.7s+1)e^{s}}{21.0s+1}$$

Design $D(s)$ to add a time delay of 1 minute to the input u_2, i.e.:

$$D(s) = \begin{bmatrix} 1 & 0 \\ 0 & e^{-s} \end{bmatrix}$$

so that the modified process is:

$$G_m = GD = \begin{bmatrix} \dfrac{12.8e^{-4s}}{16.7s+1} & \dfrac{-18.9e^{-4s}}{21.0s+1} \\ \dfrac{6.67e^{-10s}}{10.9s+1} & \dfrac{-19.4e^{-4s}}{14.4s+1} \end{bmatrix}$$

the simplified decoupling compensators is:

$$g_{I1} = 1.48\frac{(16.7s+1)}{21.0s+1}; \qquad g_{I2} = 0.34\frac{(14.4s+1)e^{-6s}}{10.9s+1}$$

They are implemented together with the delay matrix, *D(s)*. It comes at a cost of an additional delay of one minute in Loop 2.

If we use generalized decoupling for the process, all the compensators are causal except for:

$$g_{I12} = \frac{-366.66(16.7s+1)(10.9s+1)e^{s}}{124.74(16.7s+1)(14.4s+1)e^{-6s} - 248.32(21.0s+1)(10.9s+1)}$$

which cannot be implemented. But we use the *D(s)* and let:

$$G_I = (GD)^{-1}G_R$$

Alternatively, we could let *D(s)* = *I* and increase the time delay in the (2, 2) element of to 4 minutes and have the same generalized compensator, but

$$g_{I21} = \frac{84.48(21.0s+1)(14.4s+1)}{124.74(16.7s+1)(14.4s+1)e^{-6s} - 248.32(21.0s+1)(10.9s+1)}$$

$$g_{I22} = \frac{-248.32(21.0s+1)(10.9s+1)e^{-2s}}{124.74(16.7s+1)(14.4s+1)e^{-6s} - 248.32(21.0s+1)(10.9s+1)}$$

Because of complexities in implementation or the lack of a high-quality dynamic model, it is often necessary to make use of simplifications of the ideal decoupler.

8.1.5. Inverted Decoupling

The general design procedure for inverted decoupling scheme is as follows:

Assign the process input as

$$\begin{aligned} u_1 &= v_1 + D_{12}u_2 \\ u_2 &= v_2 + D_{21}u_1 \end{aligned} \tag{8.1.12}$$

substituting equation (8.1.12) into equation (8.1.1), and let

$$D_{12} = -\frac{g_{12}}{g_{11}}, \qquad D_{21} = -\frac{g_{21}}{g_{22}} \tag{8.1.13}$$

Equation (8.1.13) is the decoupler with the required compensation elements are the same for the simplified conventional decoupler but the apparent process is the same as was achieved with generalized decoupling, that is:

$$y_1 = g_{11}v_1, \qquad y_2 = g_{22}v_2 \tag{8.1.14}$$

The inverted decoupling scheme is shown as in Fig. 8.1.3. One advantage of inverted decoupling is that the apparent process seen by each controller, when decoupling is implemented, is the same as if there were no decoupling and the alternate controller were in the manual mode.

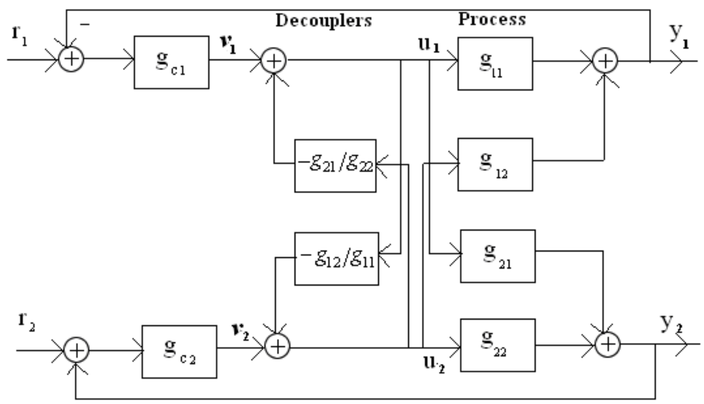


Fig 8.1.3 inverted decoupling.

For TITO processes with multiple time delays, $e^{-\theta_{ij}s}$ (*i, j =1, 2*), if both g_{12}/g_{11} and g_{21}/g_{22} are causal and realizable, the decoupling configuration described in Fig.8.1.4 meets the requirement. If there is one non realizable element in { g_{12}/g_{11} , g_{21}/g_{22} }, an additional time delay term can be inserted before the process input to force the non realizable element into being a realizable one. For example, suppose $\theta_{11} > \theta_{12}, \theta_{22} < \theta_{21}$ and $\theta_{11} - \theta_{12} < \theta_{21} - \theta_{22}$. The decoupling structure in Fig.8.1.5 can be used with a time delay $N_x = e^{\theta_{12}-\theta_{11}}$ is inserted at the process input u_2.

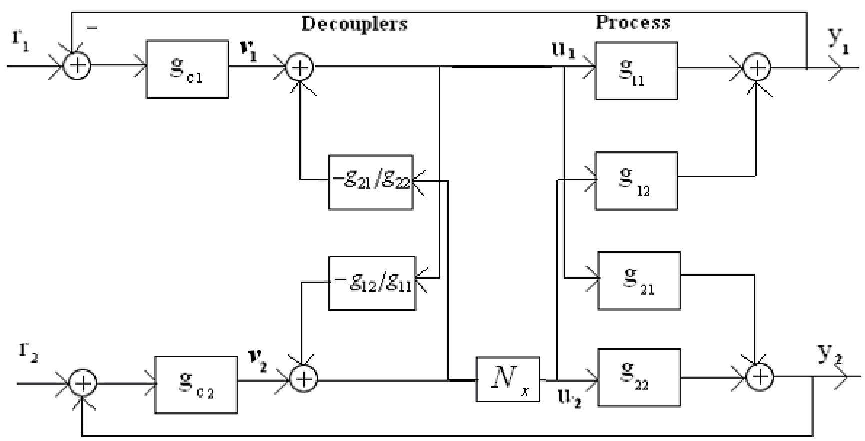


Fig. 8.1.5 Inverted decoupling with artificial time delay added to force realizability.

The term N_x increases the time delay of the process elements g_{12} and g_{22} effectively. Thus, the decoupler pairs are changed into

$$D_{12} = -\frac{g'_{12}}{g_{11}} = -\frac{g_{12}N_x}{g_{11}},$$
$$D_{21} = -\frac{g_{21}}{g'_{22}} = -\frac{g_{21}}{g_{22}N_x} \tag{8.1.15}$$

Due to the supposition that $\theta_{11} - \theta_{12} < \theta_{21} - \theta_{22}$, the extra time delay term N_x does not make D_{21} non causal. At this moment, the transfer functions from y_i to c_i are

$$y_1 = g_{11}v_1, \qquad y_2 = g_{22}N_x v_2 \tag{8.1.16}$$

Note that realizability is achieved at the expense of inserting an extra time delay in one of the control loops. The controller element should be designed for $g_{22}N_x$.

Example 8.1.3 The TITO process is defined as follows:

$$G(s) = \begin{bmatrix} \dfrac{3}{15s+1} & \dfrac{4}{10s+1} \\ \dfrac{4}{10s+1} & \dfrac{3}{15s+1} \end{bmatrix}$$

The transfer functions of the inverted decoupler are:

$$-\frac{g_{12}(s)}{g_{11}(s)} = -\frac{g_{21}(s)}{g_{22}(s)} = \frac{4(15s+1)}{3(10s+1)}$$

The resulting transfer matrix T(s) is therefore:

$$T(s) = \begin{bmatrix} \dfrac{3}{15s+1} & 0 \\ 0 & \dfrac{3}{15s+1} \end{bmatrix}$$

To reach the nominal performance defined by a first order dynamics with a time constant of five time units, the controller is:

$$G_c(s) = \begin{bmatrix} \dfrac{1+15s}{15s} & 0 \\ 0 & \dfrac{1+15s}{15s} \end{bmatrix}$$

8.1.6 Summary

In summary, the characteristics of the three decoupling schemes are listed in Table 8.1.1, one of these choices is the best balance between complexity and closed-loop controller performance.

Table: 8.1.1.Advantages and disadvantages of each decoupling methods

DECOUPLING METHODS CHARACTERISTICS	Simplified decoupling	Ideal decoupling	Inverted decoupling
When one loop is in manual mode, dynamics of the remaining loop is unchanged	NO	Yes	Yes
Decoupling elements do not contain a sum of transfer functions	Yes	No	Yes
Transfer matrix of the decoupler in series with the process does not contain sums of transfer functions	No	Yes	Yes
When loops are switched from manual to automatic mode, decoupling system initialization is simple	No	No	Yes
Saturation of the manipulated variables is easily taken into account with a PID function block having an anti reset feature and a feed forward input	No	No	Yes

Implementation with lead-lag and delay function blocks may not decrease performance	Yes	Yes	No
A feed forward input to the PID function block facilitates the decoupling system implementation	No	No	Yes
The anti reset windup of the PID function block can be used Without inverting a transfer function	Yes	No	Yes

8.2. Normalized Decoupling Control

8.2.1 Design Procedure

Using ETF matrix to replace the inverse of the process transfer matrix, and substituting Equation ETF into Equation (8.1.6), the design of normalized decoupler starts from the obtained $\hat{G}^T(s)$, determines the diagonal forward transfer function matrix $G_R(s)$, such that the decoupler $G_I(s)$

$$G_I(s) = \hat{G}^T(s) G_R(s)\,, \tag{8.2.1}$$

satisfies certain conditions for implementation.

To see how the problem definition and design procedure of Normalized decoupling control system is different from the existing methods, let each element of the process transfer function matrix, ETF and the forward transfer function element be of the form

$$g_{ij}(s) = k_{ij} \times \frac{1}{\hat{\tau}_{ij} s + 1} e^{-\theta_{ij} s} \quad \text{for } i, j = 1, 2 \ldots n \tag{8.2.2a}$$

$$\hat{g}_{ij}(s) = \hat{k}_{ij} \times \frac{1}{\hat{\tau}_{ij} s + 1} e^{-\hat{\theta}_{ij} s} \quad \text{for } i, j = 1, 2 \ldots n \tag{8.2.2b}$$

$$g_{R,ii}(s) = \frac{k_{R,ii}}{\tau_{R,ii} s + 1} e^{-\theta_{R,ii} s} \quad \text{for } i = 1, 2 \ldots n \tag{8.2.2c}$$

From Equation (8.2.2), we have

$$G_I(s) = \hat{G}^T(s) G_R(s) \Rightarrow$$

$$\begin{bmatrix} g_{I,11}(s) & g_{I,12}(s) & \ldots & g_{I,1n}(s) \\ g_{I,21}(s) & g_{I,22}(s) & \ldots & g_{I,2n}(s) \\ \ldots & \ldots & \ldots & \ldots \\ g_{I,n1}(s) & g_{I,n2}(s) & \ldots & g_{I,nn}(s) \end{bmatrix}$$

$$= \begin{bmatrix} 1/\hat{g}_{11}(s) & 1/\hat{g}_{21}(s) & \ldots & 1/\hat{g}_{n1}(s) \\ 1/\hat{g}_{12}(s) & 1/\hat{g}_{22}(s) & \ldots & 1/\hat{g}_{n2}(s) \\ \ldots & \ldots & \ldots & \ldots \\ 1/\hat{g}_{1n}(s) & 1/\hat{g}_{2n}(s) & \ldots & 1/\hat{g}_{nn}(s) \end{bmatrix} \times \begin{bmatrix} g_{R,11}(s) & 0 & \ldots & 0 \\ 0 & g_{R,22}(s) & \ldots & 0 \\ \ldots & \ldots & \ddots & \ldots \\ 0 & 0 & \ldots & g_{R,nn}(s) \end{bmatrix},$$

which result

$$\begin{bmatrix} g_{I,11}(s) & g_{I,12}(s) & \dots & g_{I,1n}(s) \\ g_{I,21}(s) & g_{I,22}(s) & \dots & g_{I,2n}(s) \\ \dots & \dots & \dots & \dots \\ g_{I,n1}(s) & g_{I,n2}(s) & \dots & g_{I,nn}(s) \end{bmatrix} = \begin{bmatrix} \dfrac{g_{R,11}(s)}{\hat{g}_{11}(s)} & \dfrac{g_{R,22}(s)}{\hat{g}_{21}(s)} & \dots & \dfrac{g_{R,nn}(s)}{\hat{g}_{n1}(s)} \\ \dfrac{g_{R,11}(s)}{\hat{g}_{12}(s)} & \dfrac{g_{R,22}(s)}{\hat{g}_{22}(s)} & \dots & \dfrac{g_{R,nn}(s)}{\hat{g}_{n2}(s)} \\ \dots & \dots & \dots & \dots \\ \dfrac{g_{R,11}(s)}{\hat{g}_{1n}(s)} & \dfrac{g_{R,22}(s)}{\hat{g}_{2n}(s)} & \dots & \dfrac{g_{R,nn}(s)}{\hat{g}_{nn}(s)} \end{bmatrix}. \tag{8.2.3}$$

and the decoupler design is to select $g_{R,ii}(s)$, such that $G_I(s)$ has the simplest form and is implementable.

Substitute Equations (8.2.2) and (8.2.1) into Equation (8.2.3), the element of the resultant decoupler is

$$g_{I,ij}(s) = \frac{k_{R,ii}}{\hat{k}_{ji}} \times \frac{\hat{\tau}_{1,ji}s+1}{\tau_{R,ii}s+1} e^{-(\theta_{R,ii}-\hat{\theta}_{ji})s} \qquad i, j = 1, 2, \dots n \tag{8.2.4}$$

The design of decoupler is now to specify the parameters in $g_{R,ii}(s)$ such that equation (8.2.4) to be physically realizable, therefore, the parameters of $g_{R,ii}(s)$ must be selected to satisfy the following conditions:

Stability: The decoupler must generate bounded responses to bounded inputs; therefore all poles of $g_{R,ii}(s)$ must lie in the open Left-Half Plane. Both dynamic properties and stability can be easily satisfied by selecting the corresponding time constant $T_{R,ii}$.

Properness: In order to avoid pure differentiation of signals, we must require that $g_{I,ij}(s)$ be proper or semi-proper, that is, the quantity of

$$\lim_{|s|\to\infty} g_{I,ij}(s) \qquad i, j = 1, 2, \dots n \tag{8.2.5a}$$

must be finite

$$0 \le \lim_{|s|\to\infty} |g_{I,ij}(s)| < \infty \qquad i, j = 1, 2, \dots n. \tag{8.2.5b}$$

This condition requires that the order of $g_{R,ii}(s)$ be higher than or equal to the orders of all $g_{I,ji}(s)$. For the given original system model, the specification of $g_{R,ii}(s)$ in Equation (8.2.2c) already satisfies this condition.

Causality: $g_{I,ij}(s)$ must be causal, which means that the decoupler must not require prediction, i.e., it must rely only on the current and previous measurements. This requires that

$$(\theta_{R,ii} - \hat{\theta}_{ij}) \ge 0, \quad \text{for } j = 1, 2, \dots n \tag{8.2.6}$$

be satisfied by letting $\theta_{R,ii} = \underset{j=1,2,\dots n}{\text{Max}} \hat{\theta}_{ij}$.

Once $G_I(s)$ is determined, the controller $G_c(s)$ can be designed by the elements of $G_R(s)$ such that $g_{c,ii}(s)g_{R,ii}(s)$ for $i = 1, 2, \dots n$, meet the control system specifications.

8.2.2 Examples

Example 8.2.1 For the VL column system with its transfer function matrix

$$G(s)=\begin{bmatrix}\dfrac{-2.2e^{-s}}{7s+1} & \dfrac{1.3e^{-0.3s}}{7s+1}\\ \dfrac{-2.8e^{-1.8s}}{9.5s+1} & \dfrac{4.3e^{-0.35s}}{9.2s+1}\end{bmatrix}$$

By using normalized decoupling control system design rules 1-3, the decoupled forward transfer function is selected as

$$G_R(s)=\begin{bmatrix}\dfrac{2.0785}{6.6910\ s+1}e^{-0.9558s} & 0\\ 0 & \dfrac{4.4769}{8.7939\ s+1}e^{-1.5935s}\end{bmatrix}$$

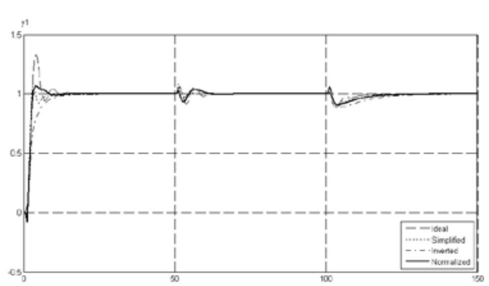
which gives a stable, causal and proper Decoupler

$$\hat{G}_I(s)=\hat{G}^T(s)\ G_R(s)=\begin{bmatrix}-1.5357 & \dfrac{8.4103\ s+1}{8.7939\ s+1}\\ (-1)\dfrac{6.1970\ s+1}{6.6910\ s+1}e^{-0.6903s} & 1.6923e^{-1.2590s}\end{bmatrix}$$

Using gain and phase margins method to $G_R(s)$, the diagonal controller is obtained as

$$G_c(s)=\begin{bmatrix}1.7633+\dfrac{0.2635}{s} & 0\\ 0 & 0.6454+\dfrac{0.0734}{s}\end{bmatrix}$$

The resultant matrices and PI controllers of the proposed design method and other three decoupling control design methods are listed in Table 8.2.1 and the output responses are shown in Figure 8.2.1, respectively, where the unit set-points change in r_1 at t=0 and r_2 at t=50. To assess their disturbance rejection capabilities, a step output disturbances $d=0.5$ at t=100 is inserted in both loops. It is noted that a delay element is required in the three existing decoupling methods to make the decouplers causal, while in is embedded into the design process in the normalized decoupling method. The Integrated Absolute Error (IAE) for each case is calculated and listed in Table 8.2.2.



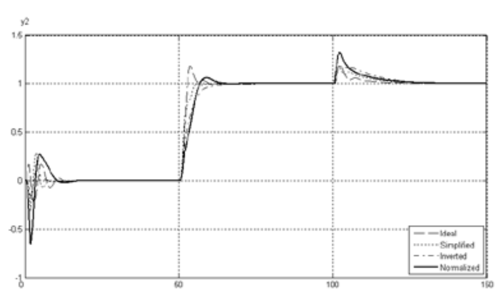


Figure 8.2.1 Output responses to step inputs

Table 8.2.1 Difference Decoupling Schemes for Example 8.2.1

Decoupling method\Matrix	G_R	$G_I(s)$	$G_c(s)$ $A_{m,i}=3db$ $\Phi_{m,i}=\pi/3rad$
Ideal $D(s)=\begin{bmatrix}1 & 0\\ 0 & e^{-0.7s}\end{bmatrix}$	$g_{R,11}=\frac{-2.2e^{-s}}{7s+1}$ $g_{R,22}=\frac{4.3e^{-1.05s}}{9.2s+1}$	$g_{I11}=g_{I22}=\frac{89.87s+9.46}{25.116s^2+59.112s+5.82}$ $g_{I12}=\frac{53.105s+5.59}{25.116s^2+59.112s+5.82}$ $g_{I21}=\frac{-42.504s^2+52.052s+6.16}{25.116s^2+59.112s+5.82}$	$g_{c1}(s)=-1.6660-\frac{0.2380}{s}$ $g_{c2}(s)=1.0669+\frac{0.1160}{s}$
Simplified $D(s)=\begin{bmatrix}1 & 0\\ 0 & e^{-0.7s}\end{bmatrix}$	$g_{R11}(s)\approx\frac{-1.3535}{7.31s+1}e^{-1.01s}$ $g_{R22}(s)\approx\frac{2.6455}{9.52s+1}e^{-1.06s}$	$g_{I1}=0.5909$ $g_{I2}=\frac{5.9907s+0.6512}{9.5s+1}e^{-0.75s}$	$g_{c1}(s)=-2.7999-\frac{0.3830}{s}$ $g_{c2}(s)=1.7775+\frac{0.1867}{s}$
Inverted $N_x=e^{\theta_{12}-\theta_{11}}=e^{-0.7s}$	$g_{R11}(s)\approx\frac{-1.3535}{7.31s+1}e^{-1.01s}$ $g_{R22}(s)\approx\frac{2.6455}{9.52s+1}e^{-1.06s}$	$g_{I1}=0.5909$ $g_{I2}=\frac{5.9907s+0.6512}{9.5s+1}e^{-0.75s}$	$g_{c1}(s)=-1.6660-\frac{0.2380}{s}$ $g_{c2}(s)=1.0669+\frac{0.1160}{s}$
Normalized	$g_{R,11}=\frac{2.0785}{6.6910\,s+1}e^{-0.9558s}$ $g_{R,22}=\frac{4.4769}{8.7939\,s+1}e^{-1.5935s}$	$g_{I,11}=-1.5357$ $g_{I,12}=\frac{8.4103\,s+1}{8.7939\,s+1}$ $g_{I,21}=-\frac{6.1970\,s+1}{6.6910\,s+1}e^{-0.6903s}$ $g_{I,22}=1.6923e^{-1.2590s}$	$g_{c1}=1.7633+\frac{0.2635}{s}$ $g_{c2}=0.6454+\frac{0.0734}{s}$

Table 8.2.2 IAE values for each decoupling scheme

METHOD\IAE	IDEAL		SIMPLIFIED		INVERTED		NORMALIZED	
	Loop 1	Loop 2	Loop 1	Loop 2	Loop 1	Loop 2	Loop 1	Loop 2
Tracking	2.770	0.971	2.802	0.811	3.153	0.440	2.273	2.068
Interaction	0.364	2.253	0.179	2.076	0.091	3.308	0.313	3.567
Disturbance	0.546	0.926	0.883	1.510	1.411	2.432	0.917	2.276

In order to compare robustness of different decoupling control schemes, mismatch the process model by increasing all 4 steady-state gains, 4 time constants and 4 time delays by a factor of 1.5, separately, with all decouplers and controller parameters kept the same as before. The closed-loop responses are shown in Figures 8.2.2-8.2.4, respectively. It shows that under such model mismatches, the response of normalized decoupling control system exhibits better robustness than that of other design methods.

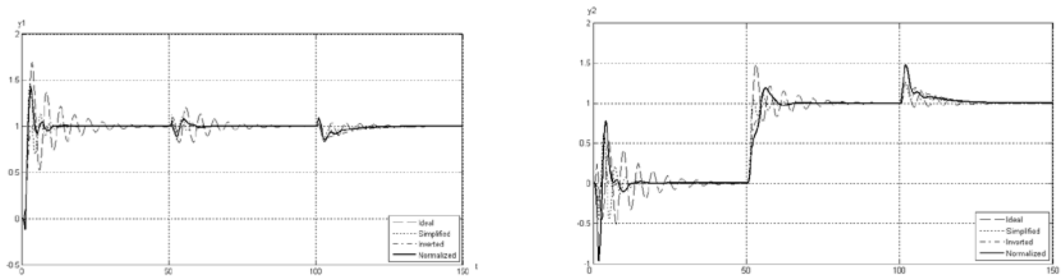


Figure 8.2.2 Robustness to steady state gain variations, $k_{ij} = 1.5 \times k_{ij}^0$ for $i, j = 1, 2$

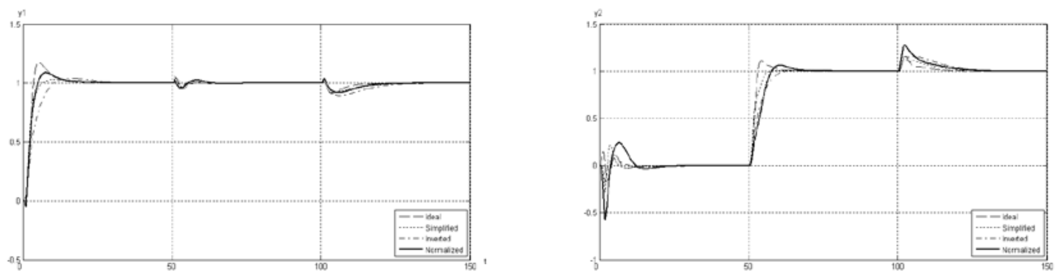


Figure 8.2.3 Robustness to time constant variations, $\tau_{ij} = 1.5 \times \tau_{ij}^0$ for $i, j = 1, 2$

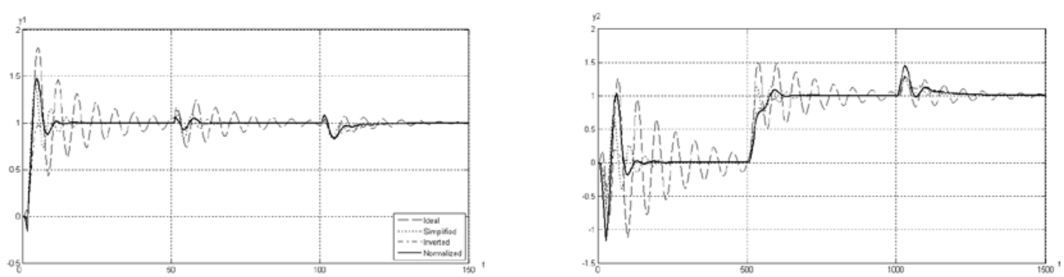


Figure 8.2.4 Robustness to time-delay variations, $\theta_{ij} = 1.5 \times \theta_{ij}^0$ for $i, j = 1, 2$

The above simulations show that the normalized decoupling control scheme is comparable to other decoupling schemes in both nominal performance and robustness, but its design procedure is much simpler and implementable decouplers can be directly designed.

8.3 Decoupling Control Implementation

8.3.1 Steady-State Decoupling

The difference between dynamic decoupling and steady-state decoupling is the former uses the complete, dynamic transfer function element in obtaining the decoupler, the latter uses only the steady-state gain portion of these transfer function elements.

If each transfer function element $g_{ij}(s)$, has a steady-state gain term K_{ij} and the matrix of steady-state gains is represented by K, then the steady-state decoupling are now summarized below:

Simplified steady-state decoupling for a 2 x 2 system

$$
\begin{aligned}
g_{I1} &= -\frac{K_{12}}{K_{11}} \\
g_{I2} &= -\frac{K_{21}}{K_{22}}
\end{aligned}
\tag{8.3.1}
$$

will take simple, constant numerical values, and will always be realizable, as well as implementable. For the overall closed-loop system, if the steady-state gain for each transfer function element is K_{ij}, it becomes:

$$y_1 = K_{11}\left(1-\frac{K_{12}K_{21}}{K_{11}K_{22}}\right)v_1$$
$$y_2 = K_{22}\left(1-\frac{K_{12}K_{21}}{K_{11}K_{22}}\right)v_2$$

Using the relative gain parameter λ, we obtain:

$$y_1 = \frac{K_{11}}{\lambda}v_1, \qquad y_2 = \frac{K_{22}}{\lambda}v_2$$

the effective closed-loop steady-state gain in each loop is the ratio of the open-loop gain and the relative gain parameter.

In generalized steady state decoupling, the decoupler matrix will be given by:

$$G_I = K^{-1}K_R \tag{8.3.2}$$

where K_R is the steady-state version of $G_R(s)$. Since the inversion concerns only a matrix of numbers, it will always be realizable and easily implemented.

The main advantages of steady-state decoupling are that its design involves simple numerical computations, and the resulting decouplers are always realizable.

Example 8.3.2 *Simplified and generalized Steady-state Decouplers for the Distillation Column*

The simplified steady-state decouplers are easily obtained by retaining only the steady-state gains of the transfer function elements; the results are:

$$g_{I1} = -\frac{-18.9}{12.8} = 1.48, \qquad g_{I2} = -\frac{6.6}{-19.4} = 0.34$$

and these will be implemented on the real process as:

$$u_1 = v_1 + 1.48v_2$$
$$u_2 = v_2 + 0.34v_1$$

For the generalized decoupler, we have:

$$K = \begin{bmatrix} 12.8 & -18.9 \\ 6.6 & -19.4 \end{bmatrix}$$

and multiplying by:

$$K_R = \begin{bmatrix} 12.8 & 0 \\ 0 & -19.4 \end{bmatrix}$$

we obtain:

$$G_I = \begin{bmatrix} 2.01 & 2.97 \\ 0.68 & 2.01 \end{bmatrix}$$

and these will be implemented as:

$$u_1 = 2.01v_1 + 2.97v_2$$
$$u_2 = 0.68v_1 + 2.01v_2$$

Note that these same results could have been obtained by setting $s=0$ in the dynamic decoupler.

As steady-state decouplers are very easy to design and implement, it is recommended that steady-state decoupling is the first technique to try; only if the dynamic interactions prove to be persistent will dynamic considerations be entertained.

If the dynamic aspects of the transfer function elements in each row of the transfer function matrix are similar, the dynamic decoupler will be very close to its steady-state version. As an exercise shows that for the stirred mixing tank system whether we use the simplified or the generalized approach, the dynamic decoupler obtained in each case is identical to the corresponding steady-state decoupler.

Because steady-state decoupling often leads to great improvements in control system performance with very little work or cost, the technique is most often applied in practice.

8.3.2 Effect of Controller Constraints

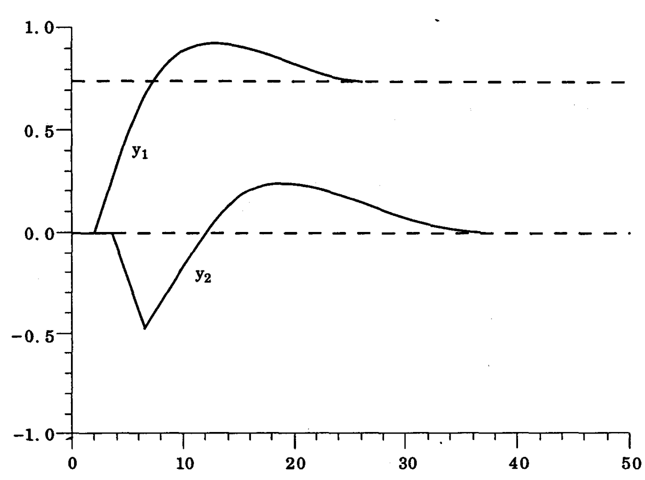
In process control, there are always constraints on the process input variables; valves cannot go beyond full open or full shut, heaters cannot go beyond full power or zero power, etc. If the controller gains are not reached their constraints, decoupling control is not affected. However, if even one of the inputs reaches a constraint then the control system can no longer send the combination of control actions required to achieve decoupling, and extremely poor (or even unstable) responses could result.

Example 8.3.1 *Steady-state Decoupler with Input Constraints*

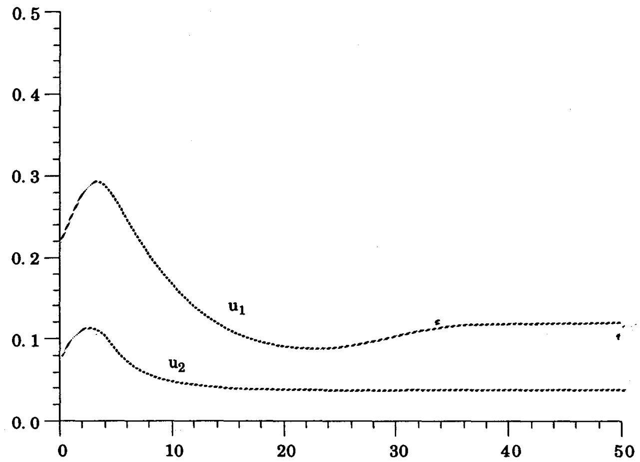
Consider the implementation of the simplified steady-state decoupler for the Wood and Berry Distillation Column.

$$g_{I1} = -\frac{-18.9}{12.8} = 1.48, \qquad g_{I2} = -\frac{6.6}{-19.4} = 0.34$$

when it is implemented together with diagonal PI controllers with tuning parameters $K_{c1} = 0.30$, $1/\tau_{I1}$ = 0.307, K_{c2} = - 0.05, $1/\tau_{I2}$ = 0.107, the closed-loop response is better than what was obtained with multiple single-loop designs. Thus a steady-state decoupler seems very attractive.

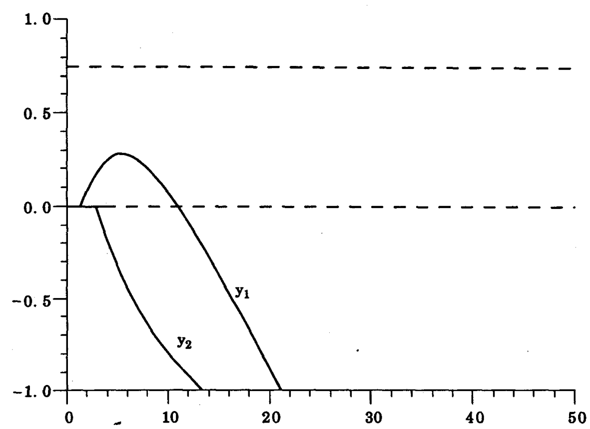


Closed-loop response of Y_1 and Y_2.

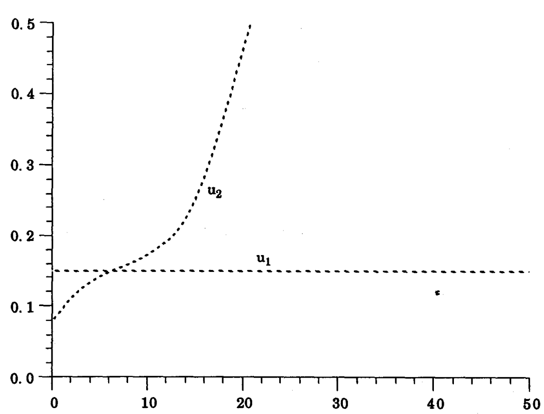


Unconstrained manipulated variables, u_1, u_2,

Suppose that u_1 has limited maximum flowrate, $0 \leq u_1 \leq 0.15$. When the same decoupling control scheme is applied, the closed-loop response is very poor once the reflux valve is full open and the system becomes unstable. This is because u_1 is pegged at its maximum value trying to increase y_1 and Loop 2 is trying to compensate imagined control actions u_1, which are computed, but not implemented. Thus steady-state decoupling cannot be applied in this case; however, more sophisticated, constraint following controllers could be implemented.



Response of Y_1 and Y_2 with constrained u_1, $0 \leq u_1 \leq 0.l5$.



Manipulated variables u_1, u_2 when $0 \leq u_1 \leq 0.15$.

In most commercial DCS, the PID function block can prevent reset windup through the use of a reset feedback input. The signal sent to the reset feedback input should be equal to the PID output, unless there is saturation. In this case, the reset feedback is the saturated PID output. The anti- reset windup feature of the PID, combined to its feed forward input, can be used to directly take into account the saturation of the manipulated variables.

8.3.3 Robustness

Since the robustness study deals with process variations and not with controller or decoupler variations, it is independent of the decoupler proximity to instability. In practice, however, it is recommended to make sure that some variations to the controller and decoupler parameters do not lead to instability. In fact, slight differences between calculated and implemented parameters may appear due to the number of significant digits. Fine-tuning of the controller and decoupler must also be possible without destabilizing the system. Finally, an unwanted delay can be created in one of the decoupling elements, when implementing decoupling using lead-lag and delay function blocks.

When the determinant of a matrix is close to zero, the matrix is said to be nearly *singular;* and one of the manifestations of singularity in a matrix is linear dependence between the columns and/or the rows, or *degeneracy* of the matrix. Now, consider the system model is given by:

$$y(s) = G(s)u(s)$$

what we are trying to do with decoupling is exactly the same as solving the system of linear equations:

$$G(s)u(s) = G_R(s)v(s)$$

to determine the decoupled process input vector *u;* therefore, if *G(s)* is degenerate, it will be extremely difficult to achieve decoupling.

The processes whose structures make them poor candidates for decoupling control are called "ill-conditioned" processes. If decoupling is applied on such a system as

$$u(s) = G_I(s)v(s)$$

where G_I is the decoupler matrix. The generalized decoupler is given by:

$$G_I(s) = G^{-1}(s)G_R(s)$$

where $G_R(s)$ is the diagonal gain matrix. If the model is perfect, the relationship between *v* and the process output will be:

$$y(s) = G(s)G^{-1}(s)G_R(s)v(s) = G_R(s)v(s)$$

and since $G_R(s)$ is diagonal, $G_I(s)$ will have achieved decoupling.

If there is an error $\Delta G(s)$ in the estimate of transfer function matrix, the true model becomes:

$$y(s) = \left(G(s) + \Delta G(s)\right)u(s) \tag{8.3.3}$$

The decoupler designed on the basis of $G(s)$, will now provide the following relationship between *v* and *y*:

$$\begin{aligned} y(s) &= \left(G(s) + \Delta G(s)\right)G^{-1}(s)G_R(s)v \\ &= G_R(s)v(s) + \Delta G(s)G^{-1}(s)G_R(s)v(s) \end{aligned} \tag{8.3.4a}$$

And

$$\Delta y(s) = \Delta G(s) G^{-1}(s) G_R(s) v(s) \tag{8.3.4b}$$

represents the amount of error introduced into y by $\Delta G(s)$.

From the definition of the inverse of a matrix, it becomes:

$$\Delta y(s) = \frac{\Delta G(s)\left[AdjG(s)\right]G_R(s)}{|G(s)|} v(s) \tag{8.3.5}$$

If the determinant of the gain matrix, $|G(s)|$, is very small, its reciprocal will be very large, and:

1. Small modeling errors will be magnified into very large errors in y.
2. Small changes in controller output v will also result in large errors in y.

Under steady state decoupling schemes, it becomes

$$\Delta y(s) = \frac{\Delta K\left[AdjK\right]K_R}{|K|} v(s) \tag{8.3.5}$$

The first condition under which decoupling will prove difficult:

> *Whenever the determinant of the process gain matrix is very small, the system will be extremely sensitive to modeling errors and decoupling will be difficult to achieve; in the limit as the determinant completely vanishes, decoupling will be entirely impossible.*

The results can also be deduced from the elements of the RGA as the elements of the RGA are related to the determinant of the process gain matrix. From the matrix method for calculating RGA elements, if C_{ij} represents the cofactor of K_{ij} (the i,jth element of the gain matrix K),

$$\left[AdjK\right]_{ij} = C_{ij}$$

then the RGA elements given by:

$$\lambda_{ij} = \frac{K_{ij}C_{ij}}{|K|} \tag{8.3.6}$$

Therefore $|K|$ goes to zero — will be manifested as inordinately large RGA values.

RGA Provides a measure for both control loop interactions and system's sensitivity to modeling errors.

> *Decoupling will be difficult to achieve for a system whose input/output variables are paired on very large RGA values; such a system will also be very sensitive to modeling errors.*

Example 8.3.2 Partial Steady-state Decoupling of the Heavy Oil Fractionator

Consider the partial decoupling of a 2×2 subsystem involving the top end point and the intermediate reflux temperature as the two outputs, with top draw rate and the intermediate reflux duty as the two inputs. The transfer function model is:

$$G(s) = \begin{bmatrix} \dfrac{4.05e^{-27s}}{50s+1} & \dfrac{1.20e^{-27s}}{45s+1} \\ \dfrac{4.06e^{-8s}}{13s+1} & \dfrac{1.19}{19s+1} \end{bmatrix}$$

The steady-state gains are dimensionless, and the time delays and time constants are in minutes. The steady-state gain matrix for this subsystem is:

$$K = \begin{bmatrix} 4.05 & 1.20 \\ 4.06 & 1.19 \end{bmatrix}$$

with determinant:

$$|K| = -0.0525$$

which is very close to zero, indicating that decoupling will be very difficult. The RGA for this system is:

$$\Lambda = \begin{bmatrix} -91.8 & 92.8 \\ 92.8 & -91.8 \end{bmatrix}$$

Thus, because of the small value of the determinant of the gain matrix, and the large values of the RGA elements, decoupling will be extremely difficult for this system. #

Example 8.3.3 Degeneracy and Decoupling Control of the Stirred Mixing Tank System

Consider the stirred mixing tank expanded with an additional heater. If we fix the heater at a constant power and only consider the hot and cold stream flowrates as inputs, the linearized model is given by

$$G(s) = \begin{bmatrix} \dfrac{0.261}{43.5s+1} & \dfrac{0.261}{43.5s+1} \\ \dfrac{0.01235(T_H - T_s)}{21.7s+1} & \dfrac{0.01235(T_C - T_s)}{21.7s+1} \end{bmatrix}$$

Investigate the possible degeneracy for this system under two conditions:

- *Condition I*

  Hot Stream Temperature: $T_H = 53^0$C

  Cold Stream Temperature: $T_c = 13^0$C

  Operating steady-state tank temperature: $T_s = 33^0$C

- *Condition 2*

  Hot Stream Temperature: $T_H = 28.5^0$C

  Cold Stream Temperature: $T_c = 28^0$C

  Operating steady-state tank temperature: $T_s = 33^0$C

Solution: Under Condition 1, the steady-state gain matrix is given by:

$$K = \begin{bmatrix} 0.261 & 0.261 \\ 0.247 & -0.247 \end{bmatrix}$$

whose determinant is - 0.129; the RGA for this system is obtained as:

$$\Lambda = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}$$

even though the indication is that there will be significant interactions, decoupling is very feasible in this case. Because of the -0.247 element, the rows and the columns are *not* linearly dependent even though they may appear to be so.

The gain matrix under Condition 2 is:

$$K = \begin{bmatrix} 0.261 & 0.261 \\ -0.0556 & -0.0618 \end{bmatrix}$$

and if we rounded all the elements off to only the second decimal place, *Column* 1 will be identical to Column 2.

The determinant of this matrix is -0.00162, and the RGA is:

$$\Lambda = \begin{bmatrix} 10 & -9 \\ -9 & 10 \end{bmatrix}$$

and it is now clear that the system will be easier to decouple under Condition 1 than under Condition 2.

There is a physical reason for the increased degeneracy under Condition 2: the difference between the hot and cold stream temperatures is only half of a degree. The implication is that we no longer have two input variables, we essentially have only one and we have lost one degree of freedom in controlling the process. #

8.3.4 Condition Number

There are however some situations in which the determinant is not too small, neither are the RGA elements too large, and yet decoupling is not feasible.

Consider for example the process whose gain matrix is:

$$K = \begin{bmatrix} -60 & 0.05 \\ -40 & -0.05 \end{bmatrix}$$

The determinant for this matrix is 5; and the RGA is:

$$\Lambda = \begin{bmatrix} 0.6 & 0.4 \\ 0.4 & 0.6 \end{bmatrix}$$

neither of which is unusual. Nevertheless, observe that if u_1 and u_2 are of the same order of magnitude, this matrix indicates that u_1 is the only effective control variable; the effect of u_2 on the process output variables is negligible in comparison.

This situation somewhat similar to that illustrated with the stirred mixing tank, under Condition 2; the system essentially has only one input variable; decoupling cannot be achieved in two output variables using only one input variable.

Compare this now with the process whose gain matrix is:

$$K = \begin{bmatrix} -3 & 1 \\ -2 & -1 \end{bmatrix}$$

The two processes have identical determinants and identical RGAs, but it is entirely feasible here.

It is obvious from this example that neither the determinant nor the RGA is a reliable indicator of ill-conditioning; in this case; the difference between the two processes are indicated by the eigenvalues of the two matrices.

The eigenvalues of the matrix in first case are:

$$\lambda_1 = 59.965$$
$$\lambda_2 = 0.0835$$

while those for the other process are:

$$\lambda_1 = -2 + j$$
$$\lambda_2 = -2 - j$$

We note that while the product of each set of eigenvalues is the same, for the difficult-to-decouple system, one eigenvalue is about 720 times larger that the other; for the easy-to-decouple system, the two eigenvalues have identical magnitudes.

Example 8.3.4 The following process having a gain matrix:

$$K = \begin{bmatrix} 1 & 0.001 \\ 100 & 1 \end{bmatrix}$$

The determinant for this matrix is 0.9; and the RGA is:

$$\Lambda = \begin{bmatrix} 1.11 & -0.11 \\ -0.11 & 1.11 \end{bmatrix}$$

both of which give no indication of the serious ill-conditioning problems afflicting the process.

The eigenvalues for this matrix are:

$$\lambda_1 = 1.316, \qquad \lambda_2 = 0.684$$

and the larger is less that twice the value of the smaller, again giving no indication of any problems with decoupling.

However, upon critical examination, we see that the problem with this process is that u_1 exerts 1000 times more influence on y_1 than does u_2 and 100 times more influence on y_2. Thus, we have a situation in which one input variable is significantly more influential than the other: another case of ill-conditioning. #

The most reliable indicators of ill-conditioning in a matrix are its *singular values* defined as the square root of the eigenvalues of the matrix K^TK. (Since we are now dealing with constant matrices, we do not need to use the more general definition involving the transpose of the complex conjugate, see Part I.)

In above example, the singular values are:

$$\sigma_1 = 100.01, \quad \sigma_2 = 0.009$$

The ratio of the largest to the smallest singular value of a matrix is called the *condition number,* and it is the single most reliable indicator of the conditioning of a matrix: the larger the condition number, the poorer the conditioning of the matrix.

The condition number for the process is:

$$\kappa = 1.113 \times 10^4$$ #

Example 8.3.5 The Prett and Garcia fractionator

The gain matrix is:

$$K = \begin{bmatrix} 4.05 & 1.20 \\ 4.06 & 1.19 \end{bmatrix}$$

which has singular values σ_1=5.978, σ_2 = 0.00878 and a condition number:

$$\kappa = 680.778$$

This clearly indicates serious ill-conditioning. #

It is impossible to assess the conditioning of a process *accurately*, and *reliably*, without knowing the singular values of the gain matrix *K*;

8.4 Sparse Control

8.4.1 Sparse Control Design Procedure

Since

$$\left|G_C(s) - s^{-1}\hat{G}^T(s)\right|_{ij} \approx \begin{cases} \left|g_{c,ij} - s^{-1}\hat{g}_{ji}\right| & g_{c,ij}(s) \neq 0 \\ \left|-s^{-1}\hat{g}_{ji}\right| & g_{c,ij}(s) = 0 \end{cases}, \tag{8.4.1}$$

and minimization of *J* requires that $g_{c,ij}(s)$ is determined by

$$g_{c,ij}(s) - s^{-1}\hat{g}_{ij}^{-1}(s) = 0 \Rightarrow g_{c,ij}(s)\hat{g}_{ji}(s) = \frac{1}{s}. \tag{8.4.2}$$

Image that $g_{c,ij}(s)\hat{g}_{ji}(s)$ is the forward transfer function of an artificial closed control system as shown in Fig. 8.4.1, $g_{c,ij}(s)\hat{g}_{ji}(s) = \dfrac{1}{s}$ is the ideal control performance target for the loop and the controller design is totally independent from other loops.

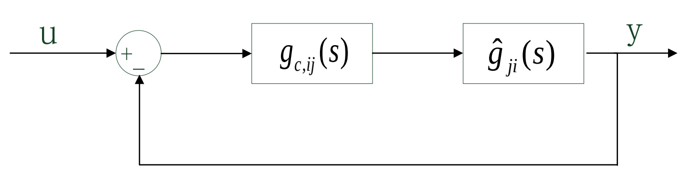


Fig. 8.4.1. Block diagram of independent design method

In designing controllers for multivariable control systems, the closed-loop integrity is very important to guarantee the overall system remains stable regardless taking out or putting in control loops. The integrity requires that each individual loop controller is no more aggressive than original single loop controller without interaction. For different combinations of $\hat{\lambda}_{ij}$ and $\hat{\gamma}_{ij}$, the general rules to preserve the integrity of sparse control for controller design are updated as:

$$\hat{k}_{ij} = \begin{cases} k_{ij} / \lambda_{ij}, & \left|\lambda_{ij}\right| < 1 \\ k_{ij}, & \left|\lambda_{ij}\right| \geq 1 \end{cases} \tag{8.4.7}$$

$$\hat{\tau}_{ar,ij} = \begin{cases} \gamma_{ij}\tau_{ar,ij} = \gamma_{ij}\tau_{ij} + \gamma_{ij}\theta_{ij}, & \gamma_{ij} > 1 \\ \tau_{ar,ij}, & 0 < \gamma_{ij} \le 1 \end{cases} \tag{8.4.8}$$

And the ETFs together with PID controller parameters for FOPDT and SOPDT are summarized for different combination of λ_{ij} and γ_{ij}, are listed in Tables 1 and 2, respectively.

Table 8.4.1 ETFs and PI parameters for FOPDT process

Mode	$\hat{g}_{ij}(s)$	$k_{P,ij}$	$k_{I,ij}$
$\lvert\lambda_{ij}\rvert < 1$, $0 < \gamma_{ij} \le 1$	$\frac{k_{ij}}{\lambda_{ij}} \times \frac{1}{\tau_{ij}s+1} e^{-\theta_{ij}s}$	$\frac{\pi\lambda_{ij}T_{ij}}{2A_m\theta_{ij}k_{ij}}$	$\frac{\pi\lambda_{ij}}{2A_m\theta_{ij}k_{ij}}$
$\lvert\lambda_{ij}\rvert < 1$, $\gamma_{ij} > 1$	$\frac{k_{ij}}{\lambda_{ij}} \times \frac{1}{\gamma_{ij}T_{ij}s+1} e^{-\gamma_{ij}\theta_{ij}s}$	$\frac{\pi\lambda_{ij}T_{ij}}{2A_m\theta_{ij}k_{ij}}$	$\frac{\pi\lambda_{ij}}{2A_m\gamma_{ij}\theta_{ij}k_{ij}}$
$\lvert\lambda_{ij}\rvert \ge 1$, $\gamma_{ij} > 1$	$\frac{sign(\lambda_{ij})\bullet k_{ij}}{\gamma_{ij}T_{ij}s+1} e^{-\gamma_{ij}\theta_{ij}s}$	$\frac{\pi T_{ij}}{2A_m\theta_{ij}k_{ij}}$	$\frac{\pi}{2A_m\gamma_{ij}\theta_{ij}k_{ij}}$
$\lvert\lambda_{ij}\rvert \ge 1$, $0 < \gamma_{ij} \le 1$	$\frac{sign(\lambda_{ij})\bullet k_{ij}}{T_{ij}s+1} e^{-\theta_{ij}s}$	$\frac{\pi T_{ij}}{2A_m\theta_{ij}k_{ij}}$	$\frac{\pi}{2A_m\theta_{ij}k_{ij}}$

Table 8.4.2 ETFs and PID parameters for SOPDT process

Mode	$\hat{g}_{ij}(s)$	$k_{P,ij}$	$k_{I,ij}$	$k_{D,ij}$
$\lvert\lambda_{ij}\rvert < 1$, $\gamma_{ij} \le 1$	$\frac{k_{ij}}{\lambda_{ij}} \times \frac{\omega_{n,ij}^2}{s^2 + 2\xi_{ij}\omega_{n,ij}s + \omega_{n,ij}^2} e^{-\theta_{ij}s}$	$\frac{\pi\lambda_{ij}\xi_{ij}}{A_m\omega_{n,ij}\theta_{ij}k_{ij}}$	$\frac{\pi\lambda_{ij}}{2A_m\theta_{ij}k_{ij}}$	$\frac{\pi\lambda_{ij}}{2A_m\omega_{n,ij}^2\theta_{ij}k_{ij}}$
$\lvert\lambda_{ij}\rvert < 1$, $\gamma_{ij} > 1$	$\frac{k_{ij}}{\lambda_{ij}} \times \frac{\omega_{n,ij}^2}{s^2 + 2\frac{\xi_{ij}}{\gamma_{ij}}\omega_{n,ij}s + \omega_{n,ij}^2} e^{-\gamma_{ij}\theta_{ij}s}$	$\frac{\pi\lambda_{ij}\xi_{ij}}{A_m\omega_{n,ij}\gamma_{ij}^2\theta_{ij}k_{ij}}$	$\frac{\pi\lambda_{ij}}{2A_m\gamma_{ij}\theta_{ij}k_{ij}}$	$\frac{\pi\lambda_{ij}}{2A_m\omega_{n,ij}^2\gamma_{ij}\theta_{ij}k_{ij}}$
$\lvert\lambda_{ij}\rvert \ge 1$, $\gamma_{ij} > 1$	$sign(\lambda_{ij})k_{ij} \times \frac{\omega_{n,ij}^2}{s^2 + 2\frac{\xi_{ij}}{\gamma_{ij}}\omega_{n,ij}s + \omega_{n,ij}^2} e^{-\gamma_{ij}\theta_{ij}s}$	$\frac{sign(\lambda_{ij})\cdot\pi\xi_{ij}}{A_m\omega_{n,ij}\gamma_{ij}^2\theta_{ij}k_{ij}}$	$\frac{sign(\lambda_{ij})\cdot\pi}{2A_m\gamma_{ij}\theta_{ij}k_{ij}}$	$\frac{sign(\lambda_{ij})\cdot\pi}{2A_m\omega_{n,ij}^2\gamma_{ij}\theta_{ij}k_{ij}}$
$\lvert\lambda_{ij}\rvert \ge 1$, $\gamma_{ij} \le 1$	$sign(\lambda_{ij})k_{ii} \times \frac{\omega_{n,ij}^2}{s^2 + 2\xi_{ij}\omega_{n,ij}s + \omega_{n,ij}^2} e^{-\theta_{ij}s}$	$\frac{sign(\lambda_{ij})\cdot\pi\xi_{ij}}{A_m\omega_{n,ij}\theta_{ij}k_{ij}}$	$\frac{sign(\lambda_{ij})\cdot\pi}{2A_m\theta_{ij}k_{ij}}$	$\frac{sign(\lambda_{ij})\cdot\pi}{2A_m\omega_{n,ij}^2\theta_{ij}k_{ij}}$

8.4.2 Working Examples

Consider a 4×4 process with

$$G(s)=\begin{bmatrix} \dfrac{4.09e^{-1.3s}}{(33s+1)(8.3s+1)} & \dfrac{-6.36e^{-0.2s}}{(31.6s+1)(20s+1)} & \dfrac{-0.25e^{-0.4s}}{21s+1} & \dfrac{-0.49e^{-5s}}{(22s+1)^2} \\ \dfrac{-4.17e^{-4s}}{45s+1} & \dfrac{6.93e^{-1.01s}}{44.6s+1} & \dfrac{-0.05e^{-5s}}{(34.5s+1)^2} & \dfrac{1.53e^{-2.8s}}{48s+1} \\ \dfrac{-1.73e^{-17s}}{(13s+1)^2} & \dfrac{5.11e^{-11s}}{(13.3s+1)^2} & \dfrac{4.61e^{-1.02s}}{18.5s+1} & \dfrac{-5.48e^{-0.5s}}{15s+1} \\ \dfrac{-11.18e^{-2.6s}}{(43s+1)(6.5s+1)} & \dfrac{14.04e^{-0.02s}}{(45s+1)(10s+1)} & \dfrac{-0.1e^{-0.05s}}{(31.6s+1)(5s+1)} & \dfrac{4.49e^{-0.6s}}{(48s+1)(6.3s+1)} \end{bmatrix}$$

The RGA (Λ), normalized gain matrix (K_N), RNGA (Λ_N), RARTA (Γ) and can be calculated respectively:

$$\Lambda=\begin{bmatrix} 3.1058 & -0.9007 & -0.4749 & -0.7302 \\ -5.0308 & 4.6742 & -0.0395 & 1.3961 \\ -0.0838 & 0.0543 & 1.5492 & -0.5197 \\ 3.0088 & -2.8278 & -0.0348 & 0.8538 \end{bmatrix},$$

$$\mathrm{K}_N=\begin{bmatrix} 0.2808 & -0.2575 & -0.0117 & -0.0181 \\ -0.0851 & 0.1519 & -0.0013 & 0.0301 \\ -0.0577 & 0.2103 & 0.2362 & -0.3535 \\ -0.8047 & 0.8570 & -0.0115 & 0.3825 \end{bmatrix},$$

$$\Lambda_N=\begin{bmatrix} 1.8033 & -0.4423 & -0.1774 & -0.1836 \\ -1.0320 & 2.2582 & 0.0291 & -0.2553 \\ -0.0212 & 0.0246 & 1.2330 & -0.2364 \\ 0.2498 & -0.8405 & -0.0847 & 1.6754 \end{bmatrix},$$

$$\Gamma=\begin{bmatrix} 0.5806 & 0.4910 & 0.3736 & 0.2515 \\ 0.2051 & 0.4831 & -0.7361 & -0.1829 \\ 0.2530 & 0.4523 & 0.7959 & 0.4549 \\ 0.0830 & 0.2972 & 2.4314 & 1.9622 \end{bmatrix}.$$

According to the pairing rules, it is a diagonal pairing, i.e., the loop should be paired as: 1-1, 2-2, 3-3 and 4-4 and the index matrix $B=[\beta_{ij}]_{n\times n}$ is calculated as:

$$B=\begin{bmatrix} 1.0000 & 0.2452 & 0.0984 & 0.1018 \\ 0.4570 & 1.0000 & 0.0129 & 0.1131 \\ 0.0172 & 0.0199 & 1.0000 & 0.1917 \\ 0.1491 & 0.5017 & 0.0505 & 1.0000 \end{bmatrix}.$$

Decentralized control: According to the adjustment rules, the ETFs of the diagonal transfer functions are calculated as

$$\hat{g}_{11}(s)=\frac{4.09e^{-1.3s}}{273.9s^2+41.3s+1},\ \hat{g}_{22}(s)=\frac{6.93e^{-1.01s}}{44.6s+1},$$

$$\hat{g}_{33}(s)=\frac{4.61e^{-1.02s}}{18.5s+1},\ \hat{g}_{44}(s)=\frac{5.2588}{302.4s^2+27.6730s+1}e^{-1.1773s}$$

The decentralized controller is designed used the gain and phase method ($A_{m,i}=4db$) as

$$G_{c\_decentralized}(s)=\begin{bmatrix} 3.0503+\frac{0.0739}{s}+20.2295s & 0 & 0 & 0 \\ 0 & 2.502+\frac{0.05611}{s} & 0 & 0 \\ 0 & 0 & 1.545+\frac{0.08351}{s} & 0 \\ 0 & 0 & 0 & 1.7530+\frac{0.0634}{s}+19.1809 \end{bmatrix}$$

Sparse control: Since $\beta_{12}>0.15,\ \beta_{21}>0.15\ \ \beta_{42}>0.15$ and $\beta_{34}>0.15$, the sparse control structure should include $g_{c,21}, g_{c,12},\ g_{c,24}$ and $g_{c,43}$ loops. The ETFs for these transfer functions are obtained, respectively, as:

$$\hat{g}_{12}=\frac{-6.36/-0.9007e^{-0.2s}}{(31.6s+1)(20s+1)}=\frac{7.0612e^{-0.2s}}{(31.6s+1)(20s+1)},$$

$$\hat{g}_{21}=\frac{-4.17/-1e^{-4s}}{45s+1}=\frac{4.17e^{-4s}}{45s+1},$$

$$\hat{g}_{42}=\frac{14.04/-1e^{-0.02s}}{(45s+1)(10s+1)}=\frac{-14.04e^{-0.02s}}{(45s+1)(10s+1)},$$

$$\hat{g}_{34}=\frac{-5.48/-0.5197e^{-0.5s}}{15s+1}=\frac{10.5445e^{-0.5s}}{15s+1}.$$

Considering the high D/τ ratios of the above four ETFs, the off-diagonal controllers are designed by IMC-Maclaurin method, the sparse controllers are then obtained as:

$$G_{c\_sparse}(s)=\begin{bmatrix} 3.0503+\frac{0.0739}{s}+20.2295s & 1.2227(1+\frac{0.02176}{s}+\frac{0.8628s}{8.631s+1}) & 0 & 0 \\ 0.6824(1+\frac{0.02037}{s}+\frac{10.4103s}{10.14s+1}) & 2.502+\frac{0.05611}{s} & 0 & -0.3732(1+\frac{0.0190}{s}+\frac{8.1913s}{64.95s+1}) \\ 0 & 0 & 1.545+\frac{0.08351}{s} & 0 \\ 0 & 0 & 0.4074(1+\frac{0.0665}{s}+\frac{0.0353s}{3.532s+1}) & 1.7530+\frac{0.0634}{s}+19.1809 \end{bmatrix}$$

The results of closed-loop step response and IAE values are given in Fig. 5.

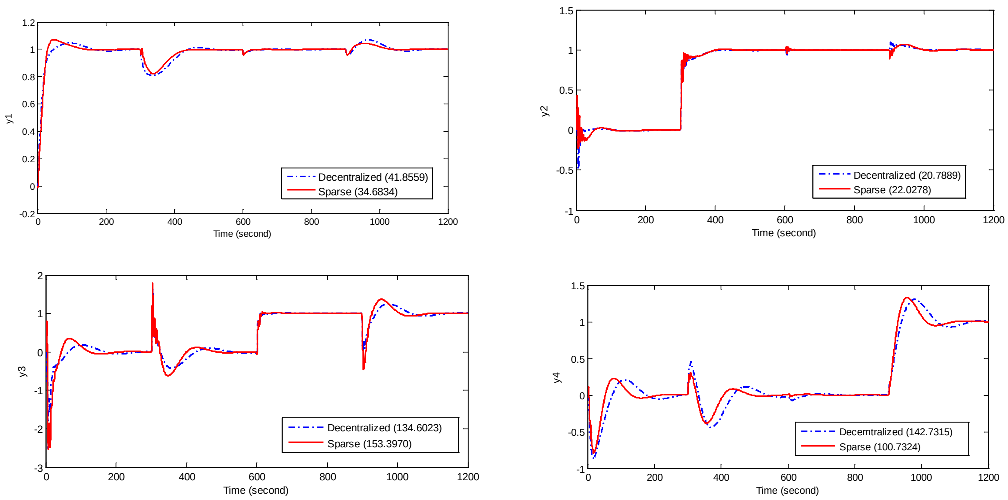


Fig. 8.4.2 Response and IAE values of A2 process

It can be seen that the overall performance of sparse control is superior.