PID control of TITO systems

Nordfeldt, Pontus

2005

Document Version:
Publisher's PDF, also known as Version of record

Department of Automatic Control
Lund Institute of Technology
Box 118
SE-221 00 Lund Sweden

Document name
LICENTIATE THESIS

Date of issue
December 2005

Document Number
ISRN LUTFD2/TFRT--3238--SE

Author(s)
Pontus Nordfeldt

Supervisor
Tore Hägglund

Sponsoring organisation
Swedish Research Council (VR)

Title and subtitle
PID Control of TITO Systems

Abstract

This thesis treats controller design and tuning for systems with two input signals and two output signals in the process industry. Two design methods that can be combined to form a core in an algorithm for automatic design and tuning for the considered systems are presented.

The proposed controller consists of a decoupler and a diagonal PID controller. This implies that the two main problems to solve are those of how to design the decoupler and how to design the diagonal controller.

The decoupling problem is treated in a general way in this thesis and a decoupler design method is proposed.

A PID controller design method is also proposed. The method is based on exhaustive search, and a simple version of software for this is presented.

The methods are combined and tested in both simulations and on a real process in an industrial environment.

Key words
PID Control, Decoupling, Linear Systems, Process Control, Multi-variable Control

Classification system and/or index terms (if any)

Supplementary bibliographical information

ISSN and key title
0280–5316

ISBN

Language
English

Number of pages
98

Recipient's notes

Security classification

PID Control of TITO Systems

Pontus Nordfeldt

Department of Automatic Control
Lund University
Lund, December 2005

Department of Automatic Control
Lund University
Box 118
SE-221 00 LUND
Sweden

ISSN 0280–5316
ISRN LUTFD2/TFRT--3238--SE

Printed in Sweden,
Lund University, Lund 2005

Contents

Acknowledgements

There are a number of persons who helped me during the work with this thesis.

First I would like to thank professor Tore Hägglund for his able supervision. He always had time for me.

Tests of the developed methods were performed at Stora Enso Publication Paper, Hylte Mill in Sweden. I would like to thank Stefan Snygg, Anna-Karin Sundquist and the others at Stora Enso who helped me with that. Mikael Petersson at ABB Automation Technology Products also helped me during that part of the work.

I would also like to thank all the people at the department who made it a good working environment. Special thanks to Anders Robertsson, Olof Garpinger, Ather Gattami, Martin Kjaer, Toivo Henningsson, Ola Slätteke and Andreas Wernrud who read and commented material from this thesis. Special thanks also to Peter Alriksson, Ather Gattami, Ola Slätteke and Toivo Henningsson for discussions about my work. Thanks to Leif Andersson and Peter Alriksson for help with Linux and LaTeX.

Margot Lundquist at Comerco AB made a language review of this thesis and I would like to thank her for that.

I would like to thank my parents and my brothers and sisters for the support they gave me.

Finally I would like to thank my Sara and my wonderful children Noah and Malkolm for support and inspiration.

1

Introduction

1.1 Background

As soon as we get up in the morning we start using products produced in large factories. The sugar in our coffee was refined in a sugar mill. The paper that the morning paper was printed on was manufactured in a paper mill. The electricity for the toaster was produced in a power plant. And it goes on like that all day long. The factories that produce the products that we are all dependent on are to a large extent automated. It increases their effectiveness and capacity, and great effort is put into making them even more automated. Over the years a lot of different control structures have been developed both in academia and in the industry. Some of them are very complicated and others are quite simple. But in spite of all the advances the old PID controller is still the most widely used controller, at least in the process industry. The PID controller has a fixed structure with only three parameters and is thus easy to tune manually without detailed knowledge of the process to be controlled. Nevertheless, many of the PID controllers that are used to control processes in the industries are poorly tuned [Bialkowski, 1993][1]. This results in a loss of performance in the processes and thus a loss of economical benefit for the industry. Consequently, methods for automatic tuning of PID controllers are valuable. Automatic tun-

[1]This reference is rather old but the author's experience is that the point of it is still valid.

ing involves both identification and controller design. There are also some practical aspects to be treated, but the algorithm must always contain an automatic design method. There has been a lot of research in the area of automatic PID design for several years and during very recent years some good results have been obtained for SISO (Single Input Single Output) systems [Skogestad, 2001], [Åström and Hägglund, 2005]. Attempts have also been made to find results for multi-variable systems [Åström *et al.*, 2002], [Wang *et al.*, 2000], [Wang and Yang, 2002], [Wang *et al.*, 2002], [Wang *et al.*, 2003]. None of these attempts have been successful which means that finding such results is an open problem. This thesis is part of an attempt to solve that problem for systems with two input signals and two output signals.

1.2 The Problem

Automation in the process industry are in many cases performed by hundreds of different controllers. Each controller tries to make a process property follow a certain reference value, and it is often independently tuned. The fact that control circuits may affect each other presents a problem. Suppose, for example, that temperature and pressure in a vessel are controlled by different controllers. Then the control circuits will, of course, affect each other. When separate controllers are used in this way the cross couplings in the process may result in poor performance. A solution to this problem is to design a multi-variable controller that takes the interactions in the process into account. There are different theoretical solutions of that kind available in the literature. Some of them use a decoupler that deals with the cross couplings, and SISO controllers to control each decoupled loop (for example [Åström *et al.*, 2002], [Wang *et al.*, 2000]). Some of them handle the process in a more direct way [Zhou and Doyle, 1998], [Åström and Wittenmark, 1990].

Systems with two input signals and two output signals are important kind of systems that may have cross couplings. They are often called TITO systems (Two Input Two Output). A lot of the TITO systems found in the process industry have an additional property apart from being TITO systems. They are close to being linear square stable non-singular systems as defined in Definition 1.1.

DEFINITION 1.1—LINEAR SQUARE STABLE NON-SINGULAR SYSTEMS
A linear square stable non-singular system is stable and has the same number of input signals and output signals. It is linear and can be represented by a linear square stable non-singular transfer-function matrix. That the transfer-function matrix is non-singular means here that it is not singular for any positive finite frequency on the real axis. □

The work presented in this thesis is focused on TITO systems. When they are mentioned in the text it is assumed that they have the properties stated in Assumption 1.1.

ASSUMPTION 1.1—TITO SYSTEMS
It is assumed that the treated TITO systems are linear square stable non-singular systems with two input signals and two output signals. □

Systems that are non-linear have to be linearized.

The work presented in this thesis is part of a project that aims to find an algorithm for automatic design and tuning of PID controllers for TITO systems. The objective of the project is stated in Objective 1.1. A clear understanding of the project's objective is crucial to an understanding of the theory presented in this thesis.

OBJECTIVE 1.1—THE OBJECTIVE OF THE PROJECT
The objective of the project is to find a suitable controller structure and an automatic design and tuning procedure for the controller. □

The specific objective of the work presented in this thesis is to provide methods that can make up the core of a solution that satisfies Objective 1.1.

The first step of this work was to find a suitable control structure. Today, TITO systems in the process industry are often controlled by two PID controllers and with no special structure for treatment of cross couplings. This control structure is depicted in the block diagram of Figure 1.1. If oscillations occur due to cross couplings it is often dealt with by slowing down one of the loops. This is an effective way of reducing the oscillations, but it may result in slow control.

In the control structure depicted in Figure 1.1 there is no part specially designed to deal with cross couplings. Each input signal is

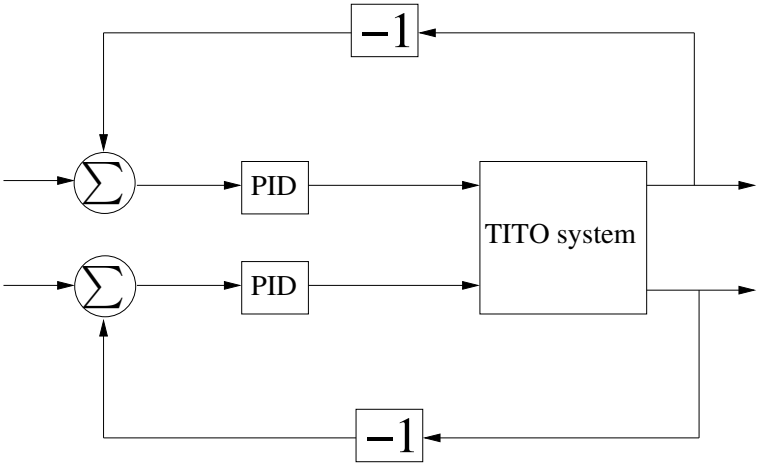

Figure 1.1 Block diagram of a common control structure for TITO systems in the process industry

paired with one of the output signals and the controllers are tuned (often manually) as well as possible. This is not satisfactory. The cross couplings may cause oscillations or other unwanted behavior in the closed loops since they are not properly treated. Here, the use of another control structure is proposed. Keeping the PID controllers in the structure is motivated because the PID controller is common in the process industry and recognized by the operators.

The structure depicted in Figure 1.2 is sometimes used for control of TITO systems but no satisfying decoupler design method, and no satisfying PID design method for decoupled loops, have been found before (see Chapter 3 and Chapter 4).

The use of decoupling is not a new idea. It has been used several times before (see Chapter 3). However, it is generally not used in the process industry. The reason is that the existing decoupler design methods are hard for an operator to use, hard to automate, or do not give a decoupler with good performance. There are some special requirements of an automatic method that are not fulfilled by these methods (see Chapter 2 and Chapter 3). A new decoupler design method that

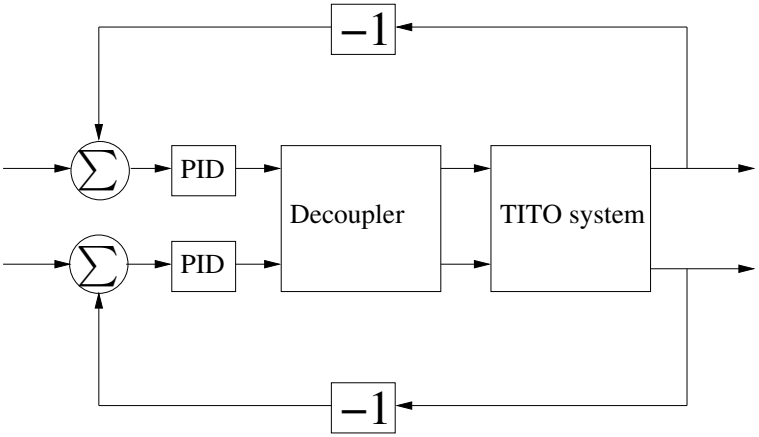

Figure 1.2 Block diagram of the proposed control structure for TITO systems in the process industry

fulfills these requirements and gives good performance is proposed in Chapter 3.

There are a lot of PID design methods that were developed for the process industry. But it turns out that none of them works satisfactorily for decoupled systems (see Chapter 4). This is due to the fact that they rely on simple models. It is true that simple process models can often be used in the process industry. But this simplicity is destroyed by decoupling. Thus a new PID design method is proposed in this thesis (see Chapter 4).

The work is aimed at TITO systems. The reason is that TITO systems are the next step in complexity above SISO systems and that they are quite common in the process industry.

With the structure depicted in Figure 1.2 the control problem is separated into two parts, one part that concerns decoupling and a second part that concerns control of decoupled loops. It turns out that it is possible to find automatic design methods for these two parts. The main contribution of this thesis is that it provides such methods.

1.3 Outline

Chapter 2 contains a discussion of specifications in design methods and special requirements of automatic tuning methods. Chapter 3 contains a description of a decoupling method. The results of Chapter 3 are of both general theoretical interest and practical interest when it comes to automatic design and PID controller tuning for TITO systems. Chapter 5 contains a description of how knowledge of the decoupler structure can be used to tune the decoupler and to improve the models to be used in the PID design procedure. Chapter 4 contains a description of a PID design method that was developed to work for decoupled systems. Chapter 6 contains a description of how the methods described in Chapter 3 and Chapter 4 can be combined to form a core in an algorithm for automatic design and PID controller tuning for TITO systems. It also contains some simulated examples. Chapter 7 contains a description of a project where the developed methods were tested on a process at Stora Enso Publication Paper, Hylte Mill in Sweden.

2

General Requirements

In Chapter 1 it was established that the main objective of the work presented in this thesis is to find methods that can be used in an algorithm for automatic design and PID controller tuning for TITO systems. In this chapter specifications and some general requirements of such methods are described.

2.1 Specifications

One of the first steps of controller design is to find out what specifications the closed loop system should satisfy. In some applications the specifications are given and easy to understand. They could, for example, be specifications of the maximum deviation from a reference trajectory. In other applications the specifications can be vague. For example it could be that a step load disturbance should be attenuated fast and that the system should be robust, and not overly sensitive to measurement disturbances. It can then be hard to tell what fast means and how the trade off between fastness and robustness (a trade off like that is common) should be done. In the case of automatic controller design and tuning it is important to find out if it is possible to find general specifications that are reasonable for the class of system for which the controller is to be used.

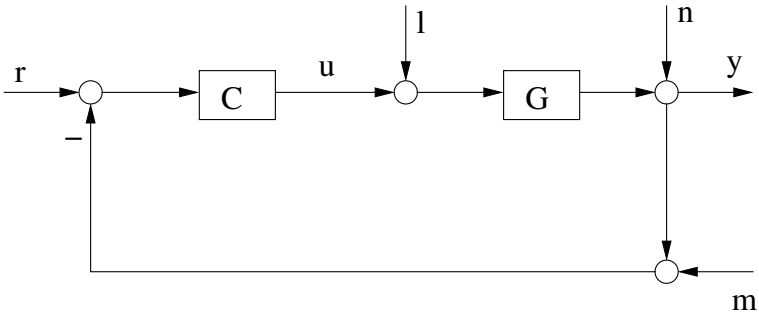

Figure 2.1 Block diagram of the closed loop system with relevant disturbances

General Disturbances in the Process Industry

In the process industry (for which this work is aimed) robustness and disturbance attenuation are often the primary objective of control, and this should be reflected in the specifications on which the design method is based. Below is a general description of what disturbances we should expect on a process in the process industry, and a short discussion.

The feedback system with input signal, output signal and disturbances is depicted in Figure 2.1. A similar figure and a discussion of specifications are present in [Zhou and Doyle, 1998].

The load disturbance (or process-input disturbance) l is generally a low-frequency disturbance and process output disturbances n and measurement disturbances m are often high-frequency disturbances.

It is undesirable to have high-frequency disturbances amplified in the control signal u since it may damage the actuator and/or give a high energy consumption. Thus, the transfer functions from n and m to u

$$G_{nu} = -C\left(I + GC\right)^{-1}$$

$$G_{mu} = -C\left(I + GC\right)^{-1}$$

should have low high-frequency gain. Since these two transfer functions are identical they can be treated as one.

Ideally all disturbances in the output signal y should be attenuated. Thus, the transfer functions from n, m, and l to y

$$G_{ny} = (I + GC)^{-1}$$

$$G_{my} = -GC\,(I + GC)^{-1}$$

$$G_{ly} = (I + GC)^{-1}\,G$$

should be small or at least bounded.

The transfer function G_{ny} is often called the output sensitivity function S and the transfer function G_{my} is often called the output complementary sensitivity function T [Zhou and Doyle, 1998].

Stability and a Stability Margin

Stability of the closed loop system is, of course, necessary but it is also necessary that the system be robust enough not to be pushed over to instability by small modeling errors or small nonlinearities. It is thus necessary to have a measure of the system stability margin, a measure of how far it is from instability. Theorem 2.1 concerns SISO systems and is called the Nyquist criterion. It is well known and documented in the literature (see for example [Goodwin *et al.*, 2001]).

THEOREM 2.1—THE NYQUIST CRITERION
If the system is open-loop stable, then, for the closed loop to be internally stable, it is necessary and sufficient that no unstable cancellations occur and that the Nyquist plot of GC not encircle the point $(-1, 0)$. □

In the case where the prerequisites of Theorem 2.1 are fulfilled and the closed loop system is stable, the shortest distance of the Nyquist curve to the critical point $(-1, 0)$ is a good measure of the system stability margin. Stability and the need for good stability margins should be reflected in the specifications.

Special Demands on Automatic Methods

Different controller design methods have different ways of optimizing more or less relevant criteria. The specification that the method should

work as an automatic method gives rise to some special demands. If a design method or a decoupling method is to be used for automatic tuning it must be simple. In this context this means that the method should not require qualitative choices to be made by the user. It is, for example, not good if approximations that depend on the process model structure have to be done. It is not a problem if the method requires complicated computations, as long as these can be performed by a computer in a numerically stable way without qualitative choices and in a limited amount of time. If the method requires parameter tuning by an operator, there must be default values that always work fairly well.

In this thesis, methods that can be used for automatic decoupling and tuning of PID controllers for TITO systems are developed. During the work with these methods the special requirements for automatic methods were taken into account. These requirements are summarized in Summary 2.1.

SUMMARY 2.1—SPECIAL REQUIREMENTS FOR AN AUTOMATIC METHOD
The method should not require that qualitative choices be made by the user. If the method requires parameter tuning by the user, there must be default values that always work fairly well. □

2.2 Optimization Criteria

An automatic design method has to perform some kind of optimization in the search for the controller structure and/or controller parameters. It is very important that the specifications are reflected in the chosen optimization criteria. If the specifications don't capture all important properties of the closed loop system, or if the optimization criteria are not carefully chosen, the designed controller might of course be bad, even though it is optimal with respect to the specifications and the optimization criteria.

2.3 Conclusion

This chapter contains a discussion of specifications and general requirements of design methods. It serves as a background for the following chapters where such methods are proposed.

3

Decoupling

3.1 Introduction

A decoupler design method for linear square stable non-singular systems is presented in this chapter. The method is used in decoupler design for TITO systems in later chapters (see Chapter 6 and Chapter 7).

Structure

In Figure 1.2 a suitable control structure for a TITO system is shown. In this chapter the theory of decoupling for more general multi-variable systems is explained. It is assumed that the systems are linear square stable non-singular systems as defined in Definition 1.1. Systems that are non-linear have to be linearized.

As explained in Chapter 1 TITO systems belong to the class of linear square stable non-singular systems, which means that the method proposed here is also valid for TITO systems. Figure 3.1 shows the general structure of the closed loop. G is the linear square stable non-singular transfer-function matrix of the process. D is the linear square stable non-singular transfer-function matrix of the decoupler. C is the diagonal transfer-function matrix with SISO PID controller transfer-functions on the diagonal. Structures similar to the one depicted in Figure 3.1 have been used in the context of decoupling before (see the references in Section 3.1 and Section 3.4).

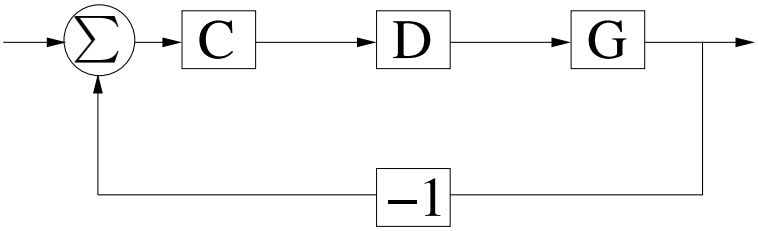

Figure 3.1 Block diagram of the closed loop system

Why Decouple?

It is important to ask what the purpose of the decoupler is before the theory of decoupler design is developed. That question is answered with the statement of a decoupler design objective in Objective 3.1.

Objective 3.1—The Decoupler Objective
The decoupler objective is to diagonalize the considered system, adding a minimal amount of dynamics and a minimal amount of time delay to the system. It is also important that the decoupler is not of high-pass character. Furthermore, the decoupler design method should fulfill the special requirements of an automatic design method (see Summary 2.1). In this context diagonalizing the considered system means choosing the transfer function D in such a way that the product GD is a diagonal transfer-function matrix. □

Below is a motivation for the decoupler objective.

The decoupler should diagonalize the system because it is then possible to use SISO controllers and SISO controller design methods for the decoupled loops. Furthermore, it is easy to make the output signals follow independent setpoint changes.

It is of course obvious that it is advantageous to add as little time delay as possible to the system by the decoupler.

The decoupler is used together with the controller and they are implemented in the same system. PID controllers are often low-pass filtered by a second order filter. This is done because it gives the controller low-pass character, which is necessary to give good high-frequency noise attenuation. The product of the decoupler and the

controller DC should be of low-pass character for the same reason. Thus, it is reasonable to require that the decoupler not be of high-pass character.

Objective 3.1 is not the only possible decoupler objective. In the work presented in this thesis PID controllers are used to control the decoupled system but it is also perfectly possible to add extra dynamics to the decoupler using the decoupler for both decoupling and loop shaping. This is obvious from the structure of the decoupler described in Section 3.2. The reason why loop shaping is not included in Objective 3.1 is that it would be hard to automate it and the work presented here aims for automatic design and tuning. Since loop shaping is not included in the design objective, it is reasonable to state that the decoupler should contain as little dynamics as possible.

The last sentence of Objective 3.1 states that the decoupler design method should fulfill the special requirements of an automatic method. The first and most obvious reason for that is that the method can then be used for automatic design. Another reason is that a method that fulfills these requirements is easy to start with even if the method does not have to be automatic.

Objective 3.1 is reasonable in many cases, but even in cases where a different decoupler objective is stated the theory developed in this chapter may be of interest.

Brief History and Contributions

Many textbooks and papers have treated decoupling in the past, including those by [Maciejowski, 1989], [Goodwin *et al.*, 2001], [Wang *et al.*, 2000], [Wang and Yang, 2002] , [Wang *et al.*, 2003] , [Wang *et al.*, 2002]. The sources that have been of special importance for the work presented here are treated more carefully in Section 3.4.

The existing decoupler design methods did not satisfy the decoupler objective, Objective 3.1. A new method for decoupler design is presented in this chapter. The contributions are that a clear expression for the whole space of possible decouplers for a linear square stable non-singular system is provided, and that very simple rules of how to choose a decoupler among those are developed. The proposed decoupler satisfies Objective 3.1.

3.2 The Decoupler Design Method

The structure of the closed loop system is depicted in Figure 3.1. A design method for the decoupler is described in this section. The decoupler D should satisfy Objective 3.1. This means that the product GD should be diagonal, that the decoupler should not be of high-pass character, that it should contain as little dynamics and time delay as possible, and that the decoupler design method should fulfill the special requirements of automatic methods. Below follows a matrix theory description of the problem and an answer to the question of how the decoupler should be chosen.

Definition 3.1 and Proposition 3.1 are well known and documented in the literature (see for example [Lancaster, 1969]).

DEFINITION 3.1
The adjoint of a matrix A, denoted $adj(A)$, is the transposed matrix of cofactors of A

$$(adj(A))_{ij} = A_{ji},$$

where A_{ji} are the cofactors of A. □

PROPOSITION 3.1

$$A \cdot adj(A) = adj(A) \cdot A = det(A) \cdot I.$$

□

The first step in the search for a decoupler design method is to find an expression for the whole space of possible decouplers, from which a suitable one could be chosen. Proposition 3.2 gives an expression for this space.

PROPOSITION 3.2
All matrices D that make the system GD diagonal can be factorized as the adjoint of G times a diagonal matrix K. D is then given by

$$D = adj(G) \cdot K.$$

Proof The proposition follows directly from Proposition 3.1. □

Proposition 3.2 may seem trivial but it is important because it shows the whole space of possible decouplers.

In Objective 3.1 the decoupler objective is stated. The question is how K should be designed to satisfy the requirements stated there.

All real processes attenuate sufficiently high frequencies and should thus be represented by models of low-pass character. This means that every component of the transfer function matrix of the process is of low-pass character. This, in turn implies that all cofactors of the transfer function matrix and thus the adjoint of the process transfer function matrix is of low-pass character. Thus it is not of high-pass character, which is one of the properties that Objective 3.1 states that the decoupler should have. Below $K = I$ and thus $D = adj(G)$ is taken as a starting point in the search for a D that fulfills the requirements of Objective 3.1.

The decoupler objective states that the decoupler should contain as little time delay as possible. This can be achieved by a modification of K. It is obvious from Proposition 3.2 that a common factor of the elements of a decoupler column can be canceled out by putting its inverse as a factor in the corresponding diagonal element of K. Since it is $D = adj(G) \cdot K$ that eventually is implemented it is not a problem if K contains non-implementable elements (like inverted time-delays) as long as D does not. This means that a time delay corresponding to the shortest time delay among the column elements can be removed from each element by multiplying the corresponding diagonal element of K with the inverse of this time delay.

In the same way as time delay is removed from the decoupler above, poles and zeroes can be removed from the decoupler columns. When poles are removed from the decoupler it might also be necessary to put extra low-pass filters into K. Otherwise D might get a high-pass character.

The decoupler design method is summarized in Method 3.1

METHOD 3.1—THE DECOUPLER DESIGN METHOD

1. Start with $K = I$, then $D = adj(G)$.
2. Remove the largest common time delay of each decoupler column by multiplying the corresponding diagonal elements of K by in-

verted time delays.

3. Remove common poles and zeros of the decoupler columns by multiplying the corresponding diagonal elements of K by the inverse of the poles and zeros, possibly also multiplying diagonal elements of K by low-pass filters to avoid giving the decoupler high-pass character.

□

All the steps of Method 3.1 are easy to automate which makes the method very suitable in an algorithm for automatic tuning.

In an implementation it may be advantageous to normalize the columns of the decoupler, but it is not done here.

The proposed decoupler design method is illustrated in Example 3.1

EXAMPLE 3.1—DECOUPLER DESIGN

The decoupler design method described above is illustrated in this example.

Consider the process

$$G = \begin{pmatrix} \dfrac{s+4}{s^2+11s+10}e^{-2.6s} & \dfrac{-(s+4)}{s^2+6s+5}e^{-2.8s} \\ \dfrac{-(s+10)}{s^2+7s+10}e^{-1.3s} & \dfrac{s+10}{s^2+17s+30}e^{-1.3s} \end{pmatrix}.$$

If K is chosen as the identity matrix I, the decoupler D becomes the adjoint $adj(G)$ of the process transfer function. It is then described by

$$D = \begin{pmatrix} \dfrac{s+10}{s^2+17s+30}e^{-1.3s} & \dfrac{s+4}{s^2+6s+5}e^{-2.8s} \\ \dfrac{s+10}{s^2+7s+10}e^{-1.3s} & \dfrac{s+4}{s^2+11s+10}e^{-2.6} \end{pmatrix}.$$

If K then is modified to be

$$K = \begin{pmatrix} e^{1.3s} & 0 \\ 0 & e^{2.6s} \end{pmatrix},$$

the decoupler D becomes

$$D = \begin{pmatrix} \dfrac{s+10}{s^2+17s+30} & \dfrac{s+4}{s^2+6s+5}e^{-0.2s} \\ \dfrac{s+10}{s^2+7s+10} & \dfrac{s+4}{s^2+11s+10} \end{pmatrix},$$

which obviously is an improvement since it then contains less time delay.

Further, if K is modified to be

$$K = \begin{pmatrix} \dfrac{s+2}{s+10}e^{1.3s} & 0 \\ 0 & \dfrac{s+1}{s+4}e^{2.6s} \end{pmatrix},$$

the decoupler D becomes

$$D = \begin{pmatrix} \dfrac{1}{s+15} & \dfrac{1}{s+5}e^{-0.2s} \\ \dfrac{1}{s+5} & \dfrac{1}{s+10} \end{pmatrix},$$

which obviously contains less dynamics.

The decoupled loop then becomes

$$GD = \begin{pmatrix} gd_{11} & 0 \\ 0 & gd_{22} \end{pmatrix},$$

where

$$\begin{aligned} gd_{11} &= \frac{s+4}{s^3+26s^2+175s+150}e^{-2.6s} - \frac{s+4}{s^3+11s^2+35s+25}e^{-2.8s} \\ gd_{22} &= \frac{-(s+10)}{s^3+12s^2+45s+50}e^{-1.5s} + \frac{1}{s^2+17s+30}e^{-1.3s}. \end{aligned}$$

The dynamics of the diagonal elements of the transfer-function matrix are quite complicated. Hence, PID controller design methods that rely on simple dynamics can not be used. A new PID tuning method that was developed to work in this situation is presented in Chapter 4. □

3.3 Approximate Decoupler Design Method

In this section it is assumed that the transfer function matrix elements can be approximated with first order plus dead-time models.

$$G(s) = \begin{pmatrix} \frac{k_{11}}{T_{11}s+1}e^{-sL_{11}} & \frac{k_{12}}{T_{12}s+1}e^{-sL_{12}} \\ \frac{k_{21}}{T_{21}s+1}e^{-sL_{21}} & \frac{k_{22}}{T_{22}s+1}e^{-sL_{22}} \end{pmatrix}$$

It is motivated to look at this case because the use of first order plus dead-time models is very common in the process industry.

The adjoint of G is

$$adj(G(s)) = \begin{pmatrix} \frac{k_{22}}{T_{22}s+1}e^{-sL_{22}} & -\frac{k_{12}}{T_{12}s+1}e^{-sL_{12}} \\ -\frac{k_{21}}{T_{21}s+1}e^{-sL_{21}} & \frac{k_{11}}{T_{11}s+1}e^{-sL_{11}} \end{pmatrix}.$$

Common time delays of the model adjoint columns are removed during decoupler design, according to the method proposed above. Common poles are also removed according to the proposed method. The number of common column poles can be increased if a certain approximation is used. Each decoupler column element has a pole with a time constant T_{ij}. The shortest of these in each column is called T_s and the longest is called T_l. The elements with the long time constants are approximated with second order transfer functions using the approximation

$$\frac{1}{T_l s+1} \approx \frac{1}{(T_s s+1)((T_l - T_s)s+1)}.$$

Then the pole

$$\frac{1}{(T_s s+1)}$$

can be removed from the column.

The approximate design method is summarized in Method 3.2.

METHOD 3.2—APPROXIMATE DECOUPLER DESIGN METHOD

1. Create a first order plus dead-time model of the process.
2. Start with $K = I$, then $D = adj(G)$.
3. Remove the largest common time delay of each decoupler column by multiplying the corresponding diagonal element of K by the inverse of that time delay.
4. Use the approximation

$$\frac{1}{T_l s + 1} \approx \frac{1}{(T_s s + 1)((T_l - T_s)s + 1)}$$

   on the element with the longest time constant in each column.
5. Remove the common pole of each decoupler column by multiplying the corresponding diagonal element of K by the inverse of that pole.

□

The method is demonstrated in Example 3.2.

EXAMPLE 3.2—APPROXIMATE DECOUPLER DESIGN

Consider the process

$$G = \begin{pmatrix} \frac{3}{9s+1}e^{-3s} & \frac{2}{6s+1}e^{-2s} \\ \frac{1}{5s+1}e^{-4s} & \frac{2}{7s+1}e^{-4s} \end{pmatrix}.$$

If K is chosen as the identity matrix I, the decoupler D becomes the adjoint $adj(G)$ of the process transfer function. It is then described by

$$D = \begin{pmatrix} \frac{2}{7s+1}e^{-4s} & \frac{-2}{6s+1}e^{-2s} \\ \frac{-1}{5s+1}e^{-4s} & \frac{3}{9s+1}e^{-3s} \end{pmatrix}.$$

If K then is modified to be

$$K = \begin{pmatrix} e^{4s} & 0 \\ 0 & e^{2s} \end{pmatrix},$$

the decoupler D becomes

$$D = \begin{pmatrix} \dfrac{2}{7s+1} & \dfrac{-2}{6s+1} \\ \dfrac{-1}{5s+1} & \dfrac{3}{9s+1}e^{-s} \end{pmatrix},$$

which obviously is an improvement since it then contains less time delay.

Further, if D is approximated to be

$$D = \begin{pmatrix} \dfrac{2}{(5s+1)(2s+1)} & \dfrac{-2}{6s+1} \\ \dfrac{-1}{5s+1} & \dfrac{3}{(6s+1)(3s+1)}e^{-s} \end{pmatrix},$$

and K is modified to be

$$K = \begin{pmatrix} (5s+1)e^{4s} & 0 \\ 0 & (6s+1)e^{2s} \end{pmatrix},$$

the decoupler D becomes

$$D = \begin{pmatrix} \dfrac{2}{2s+1} & -2 \\ -1 & \dfrac{3}{3s+1}e^{-s} \end{pmatrix},$$

which obviously contains less dynamics.

Since the decoupler design method used in this example is approximate the system is not completely decoupled by the decoupler. Figure 3.2 shows the step response of the open loop system GD.

The Figure shows that the cross couplings are reduced to an acceptable level.

□

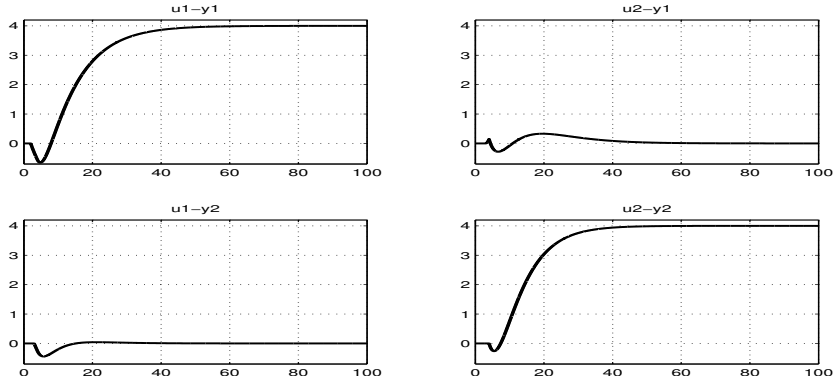

Figure 3.2 The step response of the system GD. $u_i - y_j$ is the step response of the j:th output signal to the i:th input signal.

3.4 Previous Results

Over the years decoupling has been addressed several times. Some approaches have been static [Åström *et al.*, 2002], and others dynamic [Wang *et al.*, 2000], [Wang and Yang, 2002], [Wang *et al.*, 2003], [Wang *et al.*, 2002]. Static decoupling (see for example [Åström *et al.*, 2002]) has some drawbacks. A static decoupler guarantees complete decoupling only for low frequencies. This might not be enough to achieve good performance. Further, cross couplings at other frequencies must be handled in some way during controller design. This is hard to automate. Thus, the static decoupler does not fulfill the decoupler objective.

The decoupling method described in the last chapter was developed as a continuation and generalization of previous work done by others. A decoupler D has the following property. If D diagonalizes GD then the same is true for DK, where K is a diagonal matrix. This was used in the theory of decoupling described above, and has been used before [Wang and Yang, 2002].

Different structures of D have been proposed before. Some of them are interpreted here as different choices of K (see Section 3.2 for a description of K).

Some methods have special importance for the work done here and

are briefly described below.

In [Wang *et al.*, 2000] G is assumed to be on the form:

$$G(s) = \begin{pmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{pmatrix}$$

where $g_{ij} = p_{ij}e^{-l_{ij}s}$ and p_{ij} are rational transfer functions. K is chosen for a two times two system as

$$K = \begin{pmatrix} -1/p_{22} & 0 \\ 0 & -1/p_{11} \end{pmatrix}$$

In addition to this, common time delays of the column elements of D are removed as described in the previous section. This choice of K has an obvious drawback compared to the one proposed in this chapter. There is no guarantee that the decoupler does not get high-pass character. Thus, this decoupler does not fulfill the decoupler objective (see Objective 3.1).

In [Wang *et al.*, 2003] a decoupling controller that corresponds to a choice of K as

$$k_{ji} = \frac{q_{rii}}{|G(s)|}$$

is proposed. The decoupler D is then approximated with a low order transfer-function matrix with elements that are rational transfer functions plus possible time delay. When this controller is possible to use without approximations it gives the ith open-loop transfer functions q_{rii}. However, the fact that there is a determinant of G in the denominator makes it likely that approximations have to be done. This is not desirable because approximations may result in a non-diagonalizing decoupler. Further, approximations of this kind are hard to automate, which means that the decoupler method does not fulfill the special requirements of an automatic method stated in Summary 2.1. Thus, the decoupler does not fulfill the decoupler objective (see Objective 3.1). A similar approach to decoupling is also taken in [Wang *et al.*, 2002].

3.5 Conclusion

A decoupler objective (Objective 3.1) was stated in the beginning of this chapter. Later in the chapter a decoupler design method that fulfills the objective was presented. An approximate method to be used on processes with simple models was also proposed.

4

PID Design Method

In chapter 3 a decoupling method was proposed. It fulfills Objective 3.1 and is suitable for an automatic design algorithm. In this chapter a PID controller design method is proposed. The method is suited for controller design for both systems with simple dynamics and systems with more complex dynamics. The decoupling method and the PID design method can be combined to form a core in an algorithm for automatic PID tuning and design for TITO systems.

4.1 The Problem

There are many PID design methods [Skogestad, 2001], [Åström and Hägglund, 2005]. These methods are normally based on the idea of first approximating the process dynamics with a simple model, and then basing the design on this model. This approach works well on SISO systems in the process industry, since these systems are often well described by simple models. An example of that is methods that use step responses for tuning [Ziegler and Nichols, 1942], [Hägglund and Åström, 2002], [Hägglund and Åström, 2004].

When PID controllers are to be used for multi-variable control of processes with strong cross couplings the situation is different. In many cases the system has to be decoupled. Even if the elements of the system have simple dynamics, decoupling may result in complicated diagonal elements consisting of parallel coupled processes that might have different signs and different time delays. An example of such a

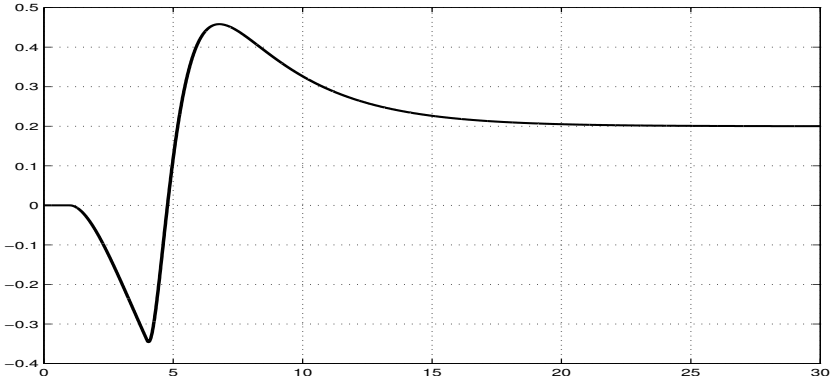

Figure 4.1 Step response of the process (4.1).

diagonal element is

$$G = \frac{1.2}{(0.5s+1)(0.7s+1)}e^{-4s} - \frac{1}{(3s+1)(2s+1)}e^{-1s}. \tag{4.1}$$

The step response of this process is shown in Figure 4.1.

If PID controllers are used to control a system with diagonal elements like this, methods that rely on simple process dynamics, like step response methods, are not appropriate. Because of that, a PID design method that does not rely on simple process dynamics was developed and is presented in this chapter.

4.2 The Design Procedure

The Controller

The PID controller is described by

$$C = K\left(1 + \frac{1}{T_i s} + T_d s\right), \tag{4.2}$$

where K is the proportional gain, T_i is the integral time and T_d is the derivative time. A pure PID controller would have infinite high-frequency gain. It is both undesirable and impossible to realize such a controller. Therefore a low-pass filter would be required. A second order low-pass filter

$$F = \frac{1}{(sT_f + 1)^2}$$

is used here.

The Optimization Criteria

The chosen optimization criteria is to minimize the integrated absolute error

$$IAE = \int_0^\infty |e(t)|dt, \tag{4.3}$$

where $e(t)$ is the control error at step load disturbances, subject to bounds on the sensitivity function and the complementary sensitivity function. The bounds on the sensitivity functions can be interpreted as two circles in the complex plane that the Nyquist curve of the open loop system has to stay outside. A larger circle that encircles these two are constructed and called the M-circle [Åström and Hägglund, 2005]. The bounds on the sensitivity functions then means that the Nyquist curve should stay outside the M-circle.

The optimization criteria was motivated and used before [Åström and Hägglund, 2005] but a short motivation is in place anyway. The general requirements of a design method for the process industry were described in Chapter 2, and below is a description of how these requirements are reflected in the optimization criteria.

Figure 3.1 shows the general disturbances l, n and m. The bound on the sensitivity functions bounds the transfer functions G_{ny} and G_{my} from n and m to the output signal y.

The second order low-pass filter gives the controller low high-frequency gain. Together with the bounds on the sensitivity functions this gives the high-frequency region of the transfer functions G_{nu} and G_{mu} from n and m to the control signal u low gain. This is important since n and m are high-frequency disturbances. Furthermore the transfer function G_{ly} from the low-frequency disturbance l to the output signal y is small since the effect of this disturbance is minimized.

Good stability margin is given by the fact that the Nyquist curve is kept at a distance from the critical point -1 by the M-circle.

The design objective is to minimize the integrated absolute error, IAE, at step load disturbances subject to the bounds on the sensitivity functions. Previously the integrated error

$$IE = \int_0^\infty e(t)dt$$

together with bounds on the sensitivity functions and other constraints has been used to approximate the IAE [Åström and Hägglund, 2005]. Furthermore, it has previously been shown that the IE of a step load disturbance is directly proportional to the inverse of the integral gain of the controller, which makes minimization of the IE easier than direct minimization of the IAE. However, if the control error $e(t)$ shifts signs, it is not good enough to calculate the IE.

The Design Method

An upper bound on the sensitivity functions is specified. The space of possible controllers is discretized in the parameters T_i, T_d, and K. For each combination of T_i and T_d, a K is found, that puts the Nyquist curve of the open loop system on the edge of the M-circle in such a way that the Nyquist curve does not encircle the -1 point, if possible. For each controller a step load disturbance is simulated and the integrated absolute error, IAE, is calculated. The controller that gives the smallest IAE is chosen.

The Sign

Since the algorithm should be able to handle processes with different signs, a sign is added to the PID controller. The output of the process after a step change of the control signal is simulated. If the output goes to a positive value or towards plus infinity the sign is chosen positive. If the output goes to a negative value or towards minus infinity the sign is chosen negative. In either case the controller is connected to the process using negative feedback.

Parameters T_d and T_i

The controller has one pole in the origin, two filter poles and two zeros. The zeros are located in:

$$z = -\frac{1}{2T_d} \pm \sqrt{\frac{1}{4T_d^2} - \frac{1}{T_i T_d}} \tag{4.4}$$

If T_i is less than $4T_d$ the zeros are complex conjugated with a real part $a = -1/2T_d$. The imaginary part will increase with decreasing T_i. If T_i is greater than $4T_d$, the zeros will be real and centered around $a = -1/2T_d$.

$1/2T_d$ is swept over the frequency region of interest. This region could, for example, be 0.001Hz to 1000Hz with the grid points spread in a logarithmic fashion. In this way many processes can be covered.

For each value of T_d, T_i is swept over a reasonable region. In most cases it is not interesting to get a controller with zeros that have very large imaginary parts or a controller with zeros at frequencies far below or above the non-integrator poles and the zeros of the process, so this region is limited.

Parameter K

For every pair of T_d and T_i, a K that gives the system the prespecified maximum values of the sensitivity functions, without making the system unstable has to be found. For stable processes this corresponds to finding a K that puts the Nyquist curve on the edge of the M-circle without making it encircle the point -1. An algorithm that checks if the -1 point is encircled has to be used.

A large K, K_{max} is chosen as a starting value. K is decreased until the point -1 is not encircled and the Nyquist curve is outside the M_s circle. If K is lowered under a certain bound K_{min}, without making the system satisfy these specifications, the conclusion is drawn that no stable closed loop system exists for the present combination of T_i and T_d. Subsequently K is gently increased until the Nyquist plot is close to the edge of the M circle. K_{max} and K_{min} works as upper and lower bounds on K. This is necessary for the method algorithm to work but it is also a natural thing to have some bounds on the controller gain.

IAE

The integrated absolute error IAE (see 4.3) is calculated by integration of a simulation of a step load disturbance response. The controller

with the smallest IAE is then chosen. To improve the accuracy the algorithm can be repeated with the intervals of T_d and T_i centered around the T_d and T_i values of the first controller and with a narrower grid.

4.3 Algorithm Complexity

The optimization is performed by means of exhaustive search. The most obvious risk of exhaustive search is that the complexity of the optimization algorithm becomes too high, so that it takes an unreasonably long time to do the optimization.

The design time was about a minute in tested examples (see Section 4.4), which is acceptable since the optimization is performed offline. Other applications may, however, have hard time constraints or require other implementations with greater values of some of the algorithm variables. Then the possibility of shortening the design time may have to be considered, and a natural starting step in that procedure is to find a time-complexity function T_c for the algorithm. It is probably also possible to reduce the computation time by using a more time-efficient programming environment than Matlab/Simulink.

It is interesting to look at a time-complexity function even if the design time is acceptable since it may give an extra understanding of the algorithm. The first step in finding this function is to look at the algorithm. Appendix A contains a Matlab implementation of the algorithm and Summary 4.1 contains a summary of the algorithm. It is recommended that the reader look at both the code in the appendix and the summary for better understanding.

SUMMARY 4.1—THE ALGORITHM

1. Do some necessary one-time operations like loading the process model, calculating center and radius of the M-circle, initializing constants, calculating the complex frequency-function vector, and determining the sign of the process.

   The number of frequency-function vector points is called n_w. A simulation time t is also set in this part of the code. These are mentioned since they affect the complexity of the method algorithm.

2. The outer loop.

   In the inner loops T_i and T_d are gridded and K is found. This is repeated twice by the outer loop. It is first done in a sparse grid to find the interesting region and then in a narrower grid in that region to find the final values for the parameters. It is of course possible to repeat the outer loop more than twice to improve the accuracy but it has been found to be unnecessary in examples.

   The number of times that the outer loop is repeated is called n_o. The complexity of the method algorithm depends linearly on this number.

3. It is reasonable to discretize the two parameters T_d and T_i in the same number of points. This number of points is called n. T_d is gridded in an outer loop and T_i in an inner loop. The complexity of the method algorithm depends quadratically on n.

4. A value of K that satisfies the optimization demands is found. The demand is that it should place the Nyquist plot of the process transfer function on the edge of the M-circle in such a way that it does not encircle the -1 critical point. This is done by starting out with a large K and decreasing it exponentially until the Nyquist curve does not encircle the critical point and the Nyquist plot is outside the M-circle. Then K is gently increased exponentially (for example by one percent in each step) until it puts the Nyquist plot on the edge of the M-circle (in practice, until the distance is smaller than an error bound).

   The search for K depends in a logarithmic fashion on K_{max}/K_{min} and of K_{opt}/K_{min}, where K_{opt} is the K that puts the Nyquist curve on the edge of the M-circle. Furthermore, for each K some operations that depend linearly on n_w have to be performed (like stability test and determination of the shortest distance between the Nyquist curve and the M-circle).

5. The closed loop system is simulated and IAE calculated. This step depends linearly on the number of time steps in the simulation. In the worst case this is t/dt, where dt is the minimum time step size of the simulation.

6. If the simulation gives the lowest IAE so far, the parameters of the controller and the grid point are saved.

□

It is easy to determine the approximate complexity of the algorithm after a look at Summary 4.1. The time-complexity function is

$$T_c \approx n_o n^2 \left(c_1 n_w \left(c_2 \log \frac{K_{max}}{K_{min}} + c_3 \log \frac{K_{opt}}{K_{min}} \right) + c_4 \frac{t}{dt} \right),$$

where c_i is a real constant.

The time-complexity function gives a good idea of which parameters are expensive to increase. Clearly it would be most expensive to increase the parameter n since the time-complexity function depends quadratically on it. The parameters n_o, n_w, K_{max}, K_{min}, t, dt and the constants are somewhat less expensive to increase. In an application where the time must be shortened a careful analysis of the constants and required values of the parameters may help.

4.4 Examples

In this section the proposed design method is illustrated in three examples.

Example 4.1—Process with Simple Dynamics
Another algorithm that tries to minimize the load disturbance step response was presented in [Panagopoulos *et al.*, 2002], [Hägglund and Åström, 2004]. That algorithm (called MIGO tuning) works well on a large class of processes but fails when it comes to more complicated processes like two parallel coupled processes with different time delays and different signs.

The proposed algorithm was compared in an example with the MIGO tuning algorithm. A simple process that both algorithms could handle was used

$$G = \frac{1}{(s+1)^4}. \tag{4.5}$$

The grid used in the proposed design method was the following: $1/2T_d$ was first divided into 12 grid points between 0.001 and 10000

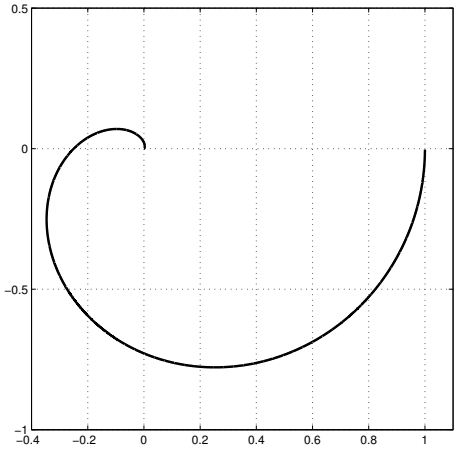

Figure 4.2 Nyquist diagram of the process (4.5).

Hz in a logarithmic fashion. The best controller for these values was calculated as described above. Subsequently $1/2T_d$ was again divided into 12 grid points with the best value of the first round in the middle. The calculations were done in Matlab (R14) and Simulink on a 2.66 GHz Pentium 4. It took 78s to find the controller.

The design method proposed in this chapter can be illustrated in the Nyquist plot of the open loop system. Figure 4.2 shows a Nyquist plot of the process (4.5). The controller tries to find a way to bend the Nyquist plot to the edge of the M-circle and at the same time to minimize the impact of load disturbances on the closed-loop system.

The controllers

The MIGO tuning algorithm tries to minimize the integrated area error by minimization of the integrated error IE subject to a constraint on the sensitivity function and some additional constraints. The MIGO tuning with an M_s value of 1.4 and an M_p value of 1.4 gave the parameters $K = 1.19$, $T_i = 2.22$ and $T_d = 1.20$ [Hägglund and Åström, 2004]. The controller was filtered by a low-pass filter with poles in $p = -20.8$. This moved the Nyquist curve slightly to the edge of an M-circle with

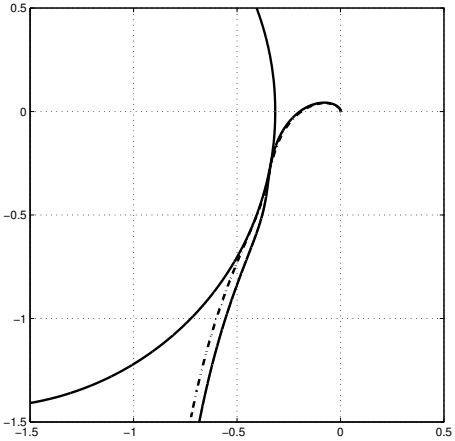

Figure 4.3 Nyquist diagram of the loop transfer functions in Example 4.1. The controllers were designed by MIGO (dot-dashed line) and the design method proposed in this chapter (solid line).

M_s and M_p values of 1.46.

The design method proposed in this chapter with M_s and M_p values of 1.46 gave the parameters $K = 1.22$, $T_i = 2.30$ and $T_d = 1.25$. The low-pass filter used had two poles in $p = -20.0$. Figure 4.3 shows the Nyquist curves of the loop transfer functions together with the M-circle.

The step load disturbance responses y and the control signals u are shown in Figure 4.4.

The integrated area error of the MIGO controlled system was calculated to:

$$IAE = 2.40$$

The integrated area error of the system controlled by the controller that the proposed design method resulted in was calculated to:

$$IAE = 2.35$$

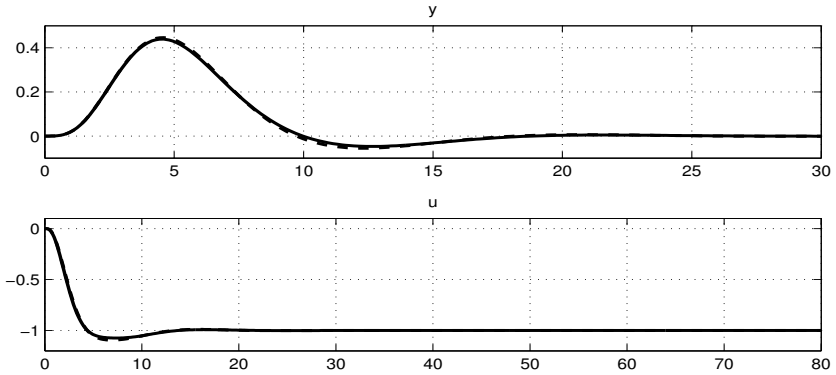

Figure 4.4 Step load disturbance response of the closed loop systems in Example 4.1. The controllers were designed by MIGO (dot-dashed line) and the design method proposed in this chapter (solid line).

Conclusions from the Example

The proposed method was developed to be used in automatic design for systems that may have either complicated or simple dynamics. Thus, it must work for both kinds. In this example it was compared with another method for a process with simple dynamics, and the designed controllers worked equally well. □

EXAMPLE 4.2—PROCESS WITH COMPLEX DYNAMICS

The proposed algorithm was used to determine a controller for an example of two parallel coupled processes with different dead-time and different signs, see (4.1). This is a kind of process that is expected to appear at diagonal elements of a decoupled two times two system in the process industry. A Nyquist plot of the process is shown in Figure 4.5.

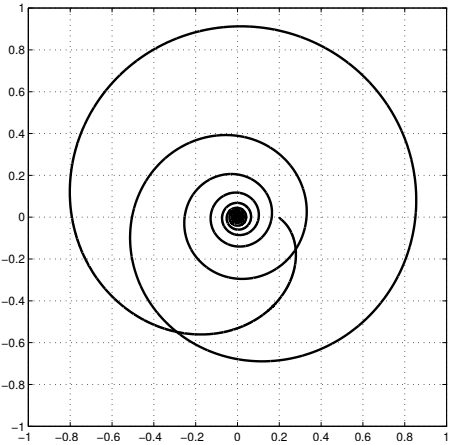

Figure 4.5 Nyquist diagram of the process (4.1).

The M_s and M_p values were 1.4. The same grid as in Example 1 was used. The calculations were done in Matlab (R14) and Simulink on a 2.66 GHz Pentium 4. It took 95s to find the controller. The proposed design method gave the controller parameters $K = 0.172$, $T_i = 0.888$, $T_d = 2.01$. The controller low-pass filter had its poles in $p = -12.5$. Figure 4.6 shows a Nyquist plot of the loop transfer function together with the M-circle. It is easy to see that the specification that the Nyquist plot should touch the edge of the M-circle holds. Further, we know that the algorithm has compared a lot of different PID controllers that fulfill this specification and chosen the one that gives the smallest IAE of a step load disturbance response.

Figure 4.7 shows a step load disturbance response of the process y controlled by the controller and the control signal u. The sign of the step response changes, indicating that a minimum of the IE would not be a good approximation of the IAE in this case.

The integrated area error of the controlled system was calculated to:

$$IAE = 6.25$$

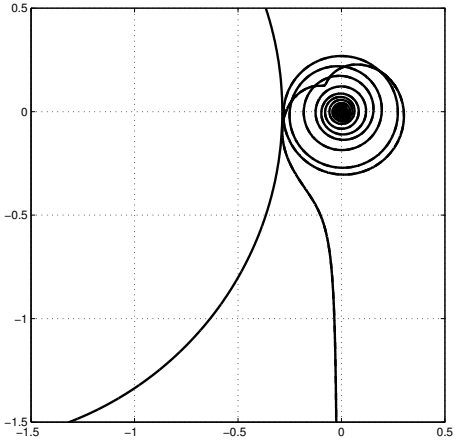

Figure 4.6 Nyquist diagram of the loop transfer function in Example 4.2.

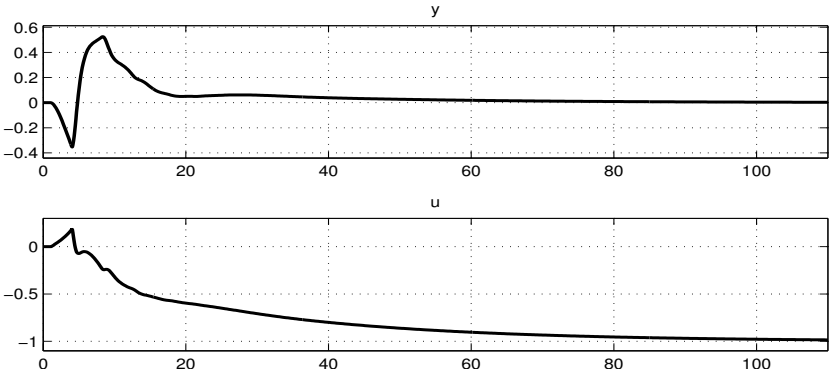

Figure 4.7 Step load disturbance response of the closed loop system in Example 4.2.

Conclusions from the Example

This example shows that the proposed design method is able to handle processes that are quite complicated, in this case a process that is expected to appear among the cases that the method has to cope with. □

EXAMPLE 4.3—HIGHER ORDER PROCESS

In Example 5 of [Skogestad, 2001] several controllers for the process (4.6) were tuned with different methods. The best one was a SIMC-PID controller. This controller was given on cascade form [Skogestad, 2001] but was converted to the form in (4.2).

$$G = \frac{1}{(s+1)(0.2s+1)(0.04s+1)(0.008s+1)} \tag{4.6}$$

A controller determined with the design method proposed in this chapter was compared with the SIMC-PID controller. The tuning was done under the same prerequisites as in Example 1 and Example 2. The tuning time was 79s.

Figure 4.8 shows a Nyquist plot of the process (4.6).

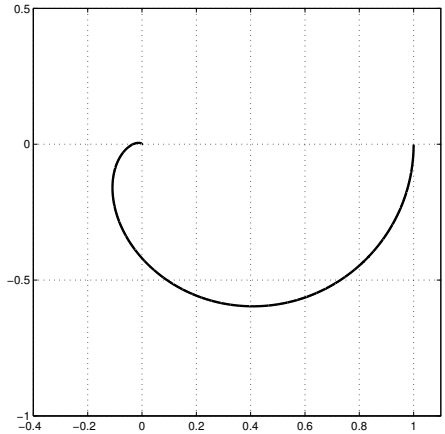

Figure 4.8 Nyquist plot of the process (4.6).

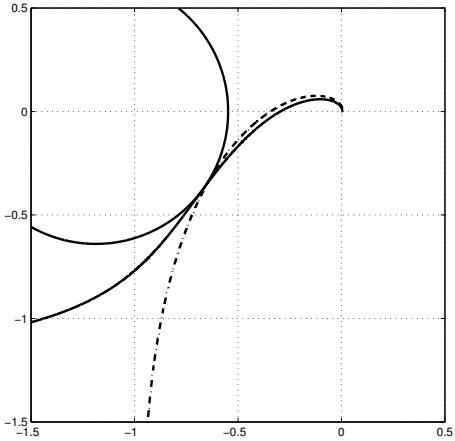

Figure 4.9 Nyquist diagram of the loop transfer functions in Example 4.3. The controllers were SIMC-PID (dot-dashed line) and a controller produced by the design method proposed in this chapter (solid line).

The Controllers

The SIMC-PID controller had the recommended parameters $K = 21.8$, $T_i = 1.22$ and $T_d = 0.180$ [Skogestad, 2001]. It was filtered by a low-pass filter. The poles of the low-pass filter were put in $p = -153$. The design method proposed in this chapter with $M_s = 2.21$ and $M_p = 2.21$ gave the parameters $K = 27.3$, $T_i = 0.200$ and $T_d = 0.127$. The low-pass filter used had two poles in $p = -197$. Figure 4.9 shows the Nyquist curve of the loop transfer functions together with the M-circle.

The step load responses y and the control signals u are shown in Figure 4.10.

The integrated absolute error of the SIMC-PID controlled system was calculated to:

$$IAE = 0.0559$$

The integrated area error of the system controlled by the controller that the proposed design method resulted in was calculated to:

$$IAE = 0.0107$$

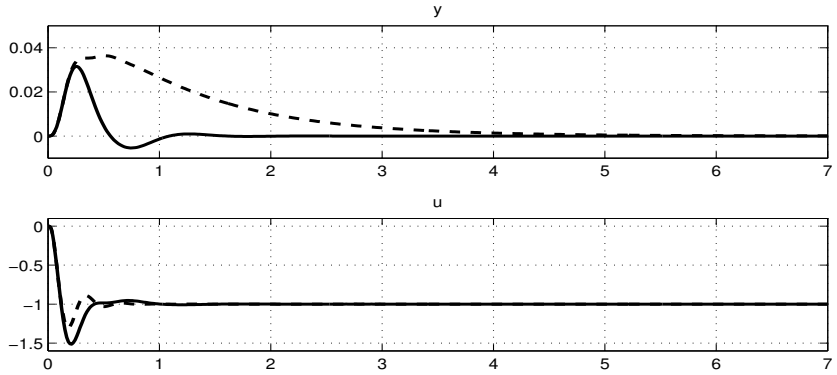

Figure 4.10 Step load disturbance response of the closed loop systems in Example 4.3. The controllers were SIMC-PID (dot-dashed line) and a controller determined by the design method proposed in this chapter (solid line).

Conclusions from the Example

The tuning method proposed in this chapter resulted in a controller that has considerably better load disturbance attenuation properties than the controller proposed in [Skogestad, 2001]. □

4.5 Conclusion

A design method for PID controllers has been proposed. The aim is to find controllers that minimize the impact of load disturbances subject to bounds on the sensitivity functions. The method has been shown in an example to work as well as another method with the same design objectives for a process with simple dynamics. Further, the method has been shown to work in an example in the difficult case of two parallel-coupled processes with different time delays and signs. It has also been compared in an example for a higher order process with a controller proposed in [Skogestad, 2001].

The proposed method requires more computations to be done than the average PID controller design method, but it works as well as or

better then previous methods and it can handle processes with complicated dynamics.

5

Decoupler Tuning

5.1 Introduction

In this chapter two methods that can be used to reduce the effects of model mismatch are presented.

In previous chapters methods for automatic decoupling and PID controller tuning for multi-variable systems were proposed. It was assumed that a model of the process was known. Furthermore it was assumed that the model didn't suffer from model mismatch. In practice it is, of course, unrealistic to assume that a perfect model is at hand. Thus it is important to find ways of coping with model mismatch. In this chapter an approach for reducing the effects of model mismatch is presented. The decoupler will have a special structure, which is due to the fact that it is designed to decouple the process. In the presence of model mismatch the decoupler will not decouple the process completely, but the structure can be used to find ways of tuning the decoupler.

When the process is satisfactorily decoupled by the decoupler PID controllers are tuned for the decoupled process. In the tuning procedure it is important to have good models of the decoupled loops. It is advantageous if these models can be identified after the decoupling by experiments performed on the decoupled loops themselves. This is advantageous since there will be no steps between the identification and the tuning in which errors can enter or be amplified. In this chap-

ter a method of simplifying the identification of the decoupled loops is presented.

5.2 Background

Consider the system

$$G(s) = \begin{pmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{pmatrix},$$

where $g_{ij} = p_{ij}e^{-l_{ij}s}$ and p_{ij} are rational transfer functions. Let the decoupler be

$$D = \begin{pmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \end{pmatrix}.$$

The open loop decoupled system GD is described by

$$GD = \begin{pmatrix} g_{11}d_{11} + g_{12}d_{21} & g_{11}d_{12} + g_{12}d_{22} \\ g_{21}d_{11} + g_{22}d_{21} & g_{21}d_{12} + g_{22}d_{22} \end{pmatrix}.$$

The open loop system GD is decoupled if the following equations are satisfied [Wang *et al.*, 2000]

$$p_{11}e^{-l_{11}s} \times d_{12} + p_{12}e^{-l_{12}s} \times d_{22} = 0 \quad (5.1)$$
$$p_{21}e^{-l_{21}s} \times d_{11} + p_{22}e^{-l_{22}s} \times d_{21} = 0. \quad (5.2)$$

In Chapter 3 it was concluded that a good solution to the decoupling problem is the adjoint of the process. Some modifications are done to avoid introducing unnecessary dynamics and time delay. The adjoint of $G(s)$ is

$$adj(G) = \begin{pmatrix} p_{22}e^{-l_{22}s} & -p_{12}e^{-l_{12}s} \\ -p_{21}e^{-l_{21}s} & p_{11}e^{-l_{11}s} \end{pmatrix}.$$

It is important to recognize that the purpose of the decoupler is to remove the effects of process cross couplings. It is no problem to alter decoupler elements to make the decoupler elements less disturbing if

a way of doing that is found. It is also important to note that if the decoupled system is controlled by a diagonal controller

$$C(s) = \begin{pmatrix} c_{11} & 0 \\ 0 & c_{22} \end{pmatrix},$$

then the open loop system becomes

$$GDC = \begin{pmatrix} (g_{11}d_{11} + g_{12}d_{21})\, c_{11} & (g_{11}d_{12} + g_{12}d_{22})\, c_{22} \\ (g_{21}d_{11} + g_{22}d_{21})\, c_{11} & (g_{21}d_{12} + g_{22}d_{22})\, c_{22} \end{pmatrix}. \tag{5.3}$$

5.3 Tuning the Decoupler

In Chapter 3 the decoupling problem is solved when the model describes the process perfectly. The method proposed here is useful when there is model mismatch. The decoupler is determined in several steps. The algorithm is summarized at the end of this section.

The first step of the procedure is to get a starting decoupler D_1. This is done with the automatic decoupling method (see Chapter 3). When model mismatch is present it is likely that the decoupler D_1 does not decouple the process completely. The second step is to compute step responses. The step responses of the left-hand side of (5.1) and (5.2) is zero when the system is decoupled. They both consist of a sum of two step responses. Ideally (when the system is completely decoupled) these two step responses are equal but with a different sign. In the presence of model mismatch they are not equal. In the proposed method the cross couplings of the approximate (approximate because of model mismatch) decoupled system are treated in the same way. Consequently it is enough to describe the procedure for one of them. One of them is described by the left-hand side of (5.1).

The starting decoupler is determined as if the model did not suffer from model mismatch. It is assumed that the signs of each element of the model are correct. The left-hand side of the (5.1) consists of two transfer functions $T_1 = p_{11}e^{-l_{11}s} \times d_{12}$ and $T_2 = p_{12}e^{-l_{12}s} \times d_{22}$. The step responses of T_1 and T_2 are recorded and called y_1 and y_2. This can be done by first setting all decoupler elements to zero except for d_{12} and recording the response of the first output signal to a step in the second

input signal. Then all decoupler elements are set to zero except for d_{22} and the response of the first output signal is again recorded to a step in the second input signal. It is easy to set the decoupler elements to zero in practice because they are implemented in the software. Each step response has a sign s_i. If the step ends at a positive value s_i is positive; otherwise it is negative. It is easy to see that one of the signs is positive and the other is negative. The time delays l_1 and l_2, the static gains k_1 and k_2 and the rise times rt_1 and rt_2 of T_1 and T_2 are determined from y_1 and y_2. This can be done in different ways. It is beyond of the scope of this chapter to choose between different methods of system identification. Either d_{12} or d_{22} is changed so that T_1 and T_2 has the same static gain. It is easy to see that this is possible.

The time delay of either d_{12} or d_{22} is changed so that T_1 and T_2 have the same time delays without making any of them negative. Then the smallest time delay of d_{12} and d_{22} is set to zero and the largest delay is decreased the same amount as was needed to achieve this. In that way no necessary delay is added to the decoupled process through the decoupler.

The rise times rt_1 and rt_2 are made equal by moving one of the poles of the d_{i2} that corresponds to the fastest rise time.

The tuning algorithm can be summarized as follows:

1. Determine a starting decoupler D_1. This is done with the automatic decoupling method (see Chapter 3). The following steps will then be carried out for each cross coupling, though it is described here for the left-hand side of (5.1) only.

2. Record step responses y_1 and y_2 of T_1 and T_2.

3. Determine the signs s_1 and s_2, the time delays l_1 and l_2, the static gains k_1 and k_2 and the rise times rt_1 and rt_2 of T_1 and T_2 from y_1 and y_2.

4. Change the static gain of either d_{12} or d_{22} so that the static gain of T_1 and T_2 becomes equal.

5. Change the time delay of either d_{12} or d_{22} so that T_1 and T_2 get the same time delays. Reduce both time delays of d_{12} or d_{22} so that the smallest of them becomes zero.

6. Move one pole of either d_{12} or d_{22} so that rt_1 and rt_2 become equal.

Comments on the Tuning Algorithm

In step five of the algorithm the static gain of either d_{12} or d_{22} is changed. The purpose of this is to give T_1 and T_2 the same static gain. Changing the static gains of both d_{12} and d_{22} with a common factor will have no effect on the open loop because the inverse of that gain will be a factor of c_{22} if a reasonable PID tuning algorithm is used, for example the one presented in Chapter 4. This is due to the fact that the common factor of T_1 and T_2 will also be a factor of the second diagonal element of (5.3). The property described above implies that it does not matter if it is the gain of d_{12} or the gain of d_{22} that is changed as long as the involved gains do not get so big or small that they give rise to numerical problems. The same is true for the other cross coupling.

In step six of the algorithm one pole of either d_{12} or d_{22} is moved. The d_{i2} that corresponds to the fastest T_i is made slower. If that d_{i2} contains more than one pole it isn't obvious which pole should be moved. It would also be possible to move more than one pole in that case. In the example below, the slowest pole is the one that is moved.

5.4 Identification of Diagonal Elements

In the presence of modeling errors no true model of the diagonal elements of the decoupled system is given, even though the decoupler is known. Consequently it might be good to find a model of the diagonal elements of the decoupled system. The diagonal elements consist of sums of two processes with, perhaps, different time delays. It is possible to identify a high order discrete time model that describes such an element well, but it is hard to automate a high order identification procedure in a safe way. One possible solution to the identification problem is to record step responses of the terms of each sum in the same way that it was done for the cross couplings of the decoupled system above and identify models for them. These models can then be combined to form models of the diagonal elements. The diagonal elements are described by the following expressions

$$p_{11}e^{-l_{11}s} \times d_{11} + p_{12}e^{-l_{12}s} \times d_{21}$$
$$p_{21}e^{-l_{21}s} \times d_{12} + p_{22}e^{-l_{22}s} \times d_{22}.$$

Each expression consists of a sum of two transfer functions. It is easy to see that step responses of each transfer functions of each expression can be recorded in four experiments. Then low order models of each transfer function are identified and combined to models of the diagonal elements. Low order identification can be performed in many ways and no specific method is recommended here. Here the dead-time, the rise time, and the static gain are determined and a second order model is produced. The model consists of a double pole, a time delay, and static gain. A second order model is preferred because it is unlikely that the transfer functions are of lower order. This is due to the fact that they consists of a product of a decoupler element and a process element that are both generally of at least order one.

Comments on the Identification Procedure

In the algorithm above four step responses are recorded and some decoupler elements may be altered. Only one of the output signals is needed for each step, but if the other output signal is also recorded it may be used in the identification algorithm. The step responses corresponding to decoupler elements that are not altered can be used, and the number of experiments that have to be performed during identification is reduced.

5.5 Example

The methods were implemented and tested on a TITO system. The system elements all consist of transfer functions with two damped poles and time delay. Uncertainty is introduced as parameter uncertainty. The process was

$$G_{true}(s) = \begin{pmatrix} \dfrac{0.8e^{-s}}{(1.7s+1)(2.7s+1)} & \dfrac{-1.9e^{-9s}}{(2s+1)(1.3s+1)} \\ \dfrac{2.6e^{-7s}}{(1.9s+1)(0.9s+1)} & \dfrac{-1.9e^{-3s}}{(1.4s+1)(1.4s+1)} \end{pmatrix},$$

and the model of the process was

$$G(s) = \begin{pmatrix} \dfrac{1.8e^{-7s}}{(1.7s+1)(0.6s+1)} & \dfrac{-1.9e^{-4s}}{(s+1)(1.5s+1)} \\ \dfrac{1.6e^{-4s}}{(1.9s+1)(0.7s+1)} & \dfrac{-1.4e^{-11s}}{(4.4s+1)(2.4s+1)} \end{pmatrix}.$$

The Decoupler Tuning

It is obvious that the model is quite different from the process. A significant model mismatch has been introduced in order to make the results as visible as possible. Thus the starting decoupler D_1 is based on a model that is quite different from the process itself. The starting decoupler is calculated as in Chapter 3

$$D_1(s) = \begin{pmatrix} \dfrac{-1.4e^{-7s}}{(4.4s+1)(2.4s+1)} & \dfrac{1.9}{(s+1)(1.5s+1)} \\ \dfrac{-1.6}{(1.9s+1)(0.7s+1)} & \dfrac{1.8e^{-3s}}{(1.7s+1)(0.6s+1)} \end{pmatrix}.$$

The decoupler tuning algorithm alters the elements of the decoupler to achieve better decoupling. After the tuning has been performed the decoupler is

$$D(s) = \begin{pmatrix} \dfrac{-1.4}{(4.4s+1)(2.4s+1)} & \dfrac{1.9e^{-7.4s}}{(s+1)(1.5s+1)} \\ \dfrac{-1.92e^{-4.2s}}{(5.9s+1)(0.7s+1)} & \dfrac{0.8}{(2.5s+1)(0.6s+1)} \end{pmatrix}.$$

Figure 5.1 and Figure 5.2 show step responses of T_1 and T_2 of the first cross coupling (the left-hand side of (5.1)) after the different steps of the decoupler tuning algorithm. d_{12} has only been altered once while d_{22} has been altered in every step of the algorithm. The figures illustrate the fact that the algorithm shapes the two step responses of T_1 and T_2 to be similar to each other but with different signs. That is the key idea of the method.

In Figure 5.3 and Figure 5.4 unit step responses of the left-hand side of (5.1) and (5.2) are plotted. The whole point of the algorithm is

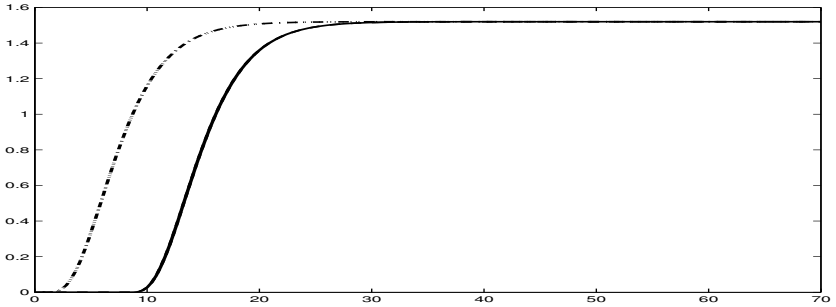

Figure 5.1 Unit step response of T_1 in (5.1) after different steps of the decoupler tuning algorithm. Starting function (dotted), after first step (dash-dotted), after second step (dashed) and final result (solid). Only two responses are seen in the figure because the decoupler element d_{12} is only altered once.

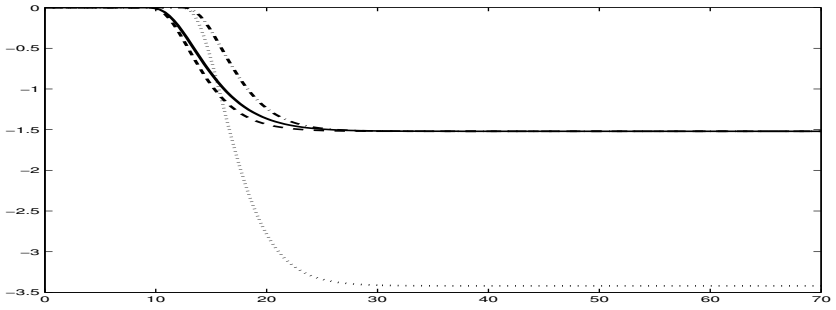

Figure 5.2 Unit step response of T_2 in (5.1) after different steps of the decoupler tuning algorithm. Starting function (dotted), after first step (dash-dotted), after second step (dashed) and final result (solid).

of course to make these cross couplings small. The figures show that this is achieved, at least for the step responses of the cross couplings. It should be pointed out that the complexity of the decoupler is not increased and the low-pass character of the decoupler (see Chapter 3) is not compromised.

It is also interesting to have a look at the cross couplings in the frequency domain. It is quite obvious that if the step response of the

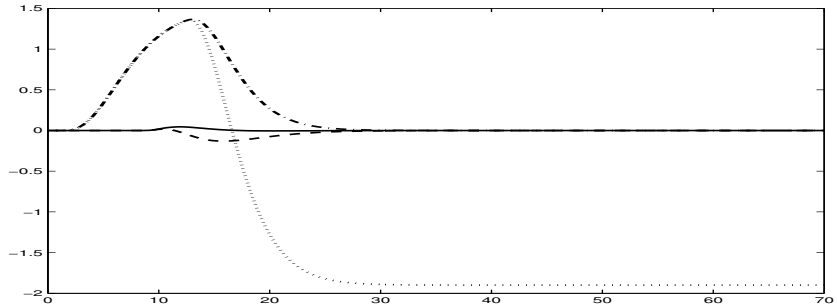

Figure 5.3 Unit step response of the left-hand side of (5.1) after different steps of the decoupler tuning algorithm. Starting function (dotted), after first step (dash-dotted), after second step (dashed) and final result (solid).

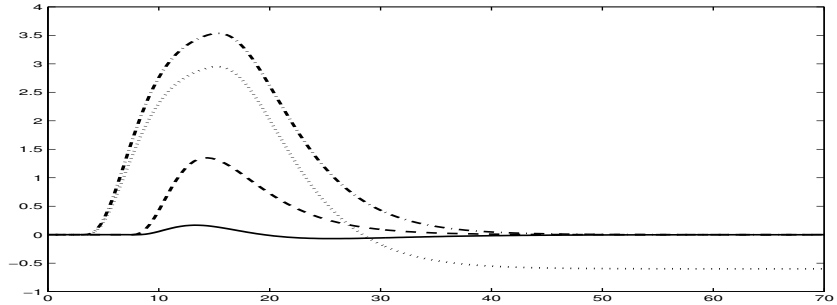

Figure 5.4 Unit step response of the left-hand side of (5.2) after different steps of the decoupler tuning algorithm. Starting function (dotted), after first step (dash-dotted), after second step (dashed) and final result (solid).

transfer function is made insignificant then the frequency response will generally be small. In Figure 5.5 and Figure 5.6, amplitude plots of the cross couplings are shown. They show that the frequency responses of the cross couplings really are made very small by the decoupler tuning.

The Identification-aid Method

The method that was proposed for the diagonal elements of the decoupled process is an identification-aid method. The idea is that knowledge

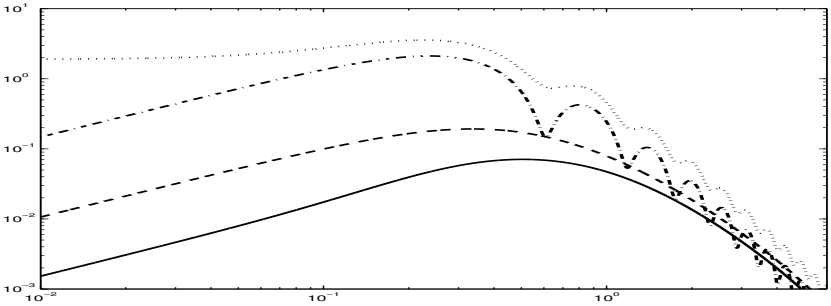

Figure 5.5 Amplitude plot of the left-hand side of (5.1) after different steps of the decoupler tuning algorithm. Starting function (dotted), after first step (dash-dotted), after second step (dashed) and final result (solid).

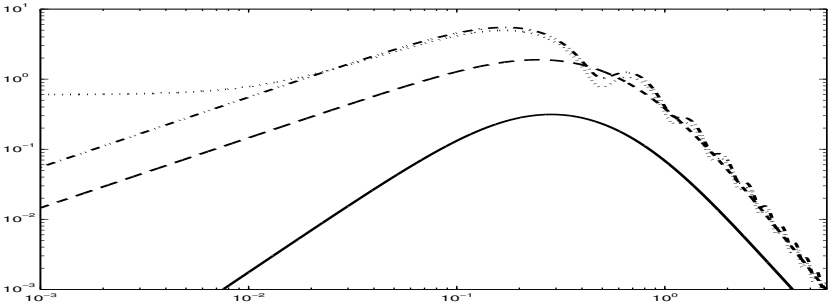

Figure 5.6 Amplitude plot of the left-hand side of (5.2) after different steps of the decoupler tuning algorithm. Starting function (dotted), after first step (dash-dotted), after second step (dashed) and final result (solid).

of the nature of the decoupler structure is used and the fact that two low order identification problems can be solved instead of one high order identification problem is recognized. There are different methods available for identification of low order models from step tests. In this example none of them was chosen because the point of the work was just to highlight the possibility of using low order identification. It is assumed that the time delay, the rise time and the static gain of the

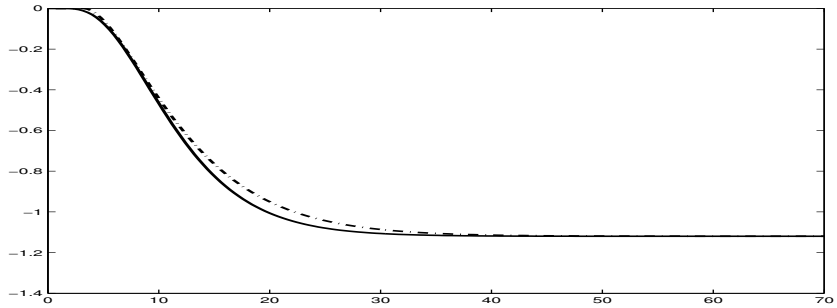

Figure 5.7 Fitted (dash-dotted) and real (solid) step response of the first transfer function in the first diagonal element of the decoupled process.

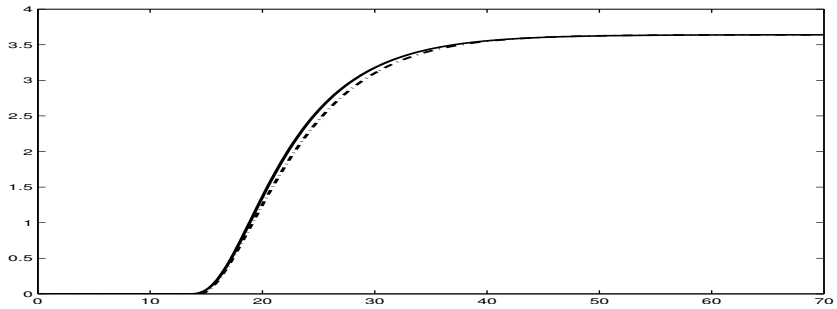

Figure 5.8 Fitted (dash-dotted) and real (solid) step response of the second transfer function in the first diagonal element of the decoupled process.

involved transfer functions can be found quite accurately.

Step responses of the diagonal elements were recorded. There were two step responses for each element. The time delay, the rise time, and the static gain of the involved transfer functions were found and second order transfer functions with time delay were fitted to the step responses. The poles of these were chosen as real double poles. Figure 5.7 and Figure 5.8 show the step responses and the fitted transfer function step responses of the first (top left) diagonal element.

Finally the identified low order models were combined to form mod-

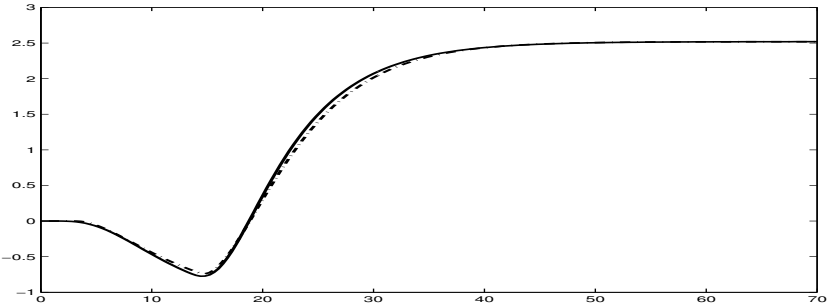

Figure 5.9 Step response of the model of the first diagonal element (dash-dotted) and the true first diagonal element (solid).

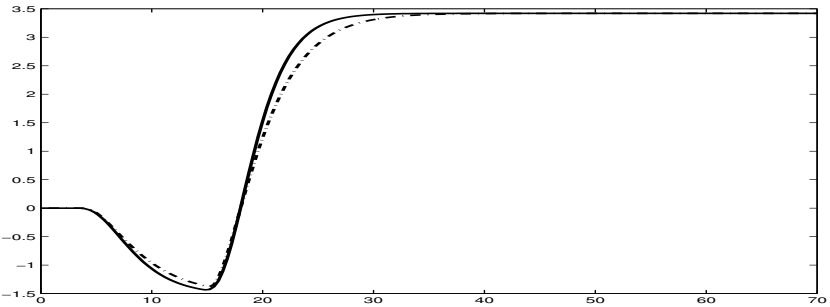

Figure 5.10 Step response of the model of the second diagonal element (dash-dotted) and the true second diagonal element (solid).

els of the diagonal elements. Figure 5.9 and Figure 5.10 show step responses of the diagonal elements. Figure 5.11 and Figure 5.12 show Nyquist plots of the diagonal elements and of the fitted models, and it can be concluded that they are very similar. The reason why it is possible to identify low-order models and model the elements like sums of them instead of using high-order models is, of course, that elements to be modeled have the structure of sums of low order models.

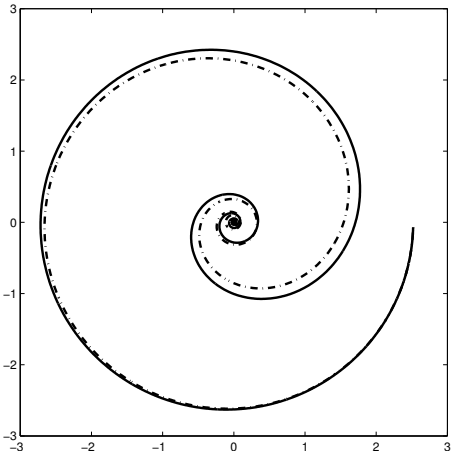

Figure 5.11 Nyquist plot of the model of the first diagonal element (dash-dotted) and the true first diagonal element (solid).

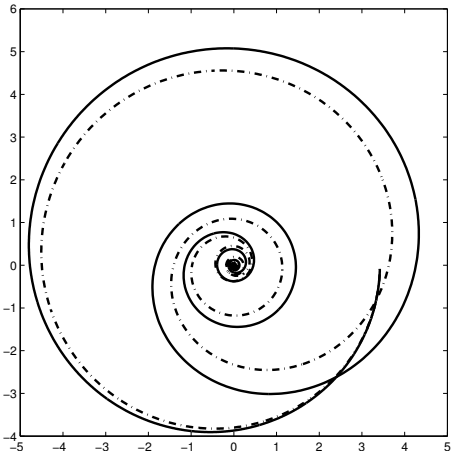

Figure 5.12 Nyquist plot of the model of the second diagonal element (dash-dotted) and the true second diagonal element (solid).

5.6 Conclusion

A method for decoupler tuning in the presence of model mismatch and an aid-method for identification of diagonal elements of the decoupled system have been presented. In both methods knowledge of the process structure and the decoupler structure is used.

6

Decoupler and Controller Design

The methods described in Chapter 3 and Chapter 4 were combined and tested in simulated examples. The methods are meant to form a core in an algorithm for automatic design and tuning of PID controllers for TITO systems that is described in this chapter. It is shown that the good disturbance attenuation properties of the proposed PID design method are inherited to a large extent by the combination of the decoupler and the PID design methods. Further, it is shown that the combination fulfills the special requirements of an automatic method.

6.1 Disturbance Attenuation

Chapter 4 contains a description of the disturbance attenuation properties of a SISO system with a controller designed by the proposed PID design method. When this method is combined with the proposed decoupler design method and applied to a TITO system, these properties are inherited to a large extent by the TITO system. This is described below.

Recall the five important transfer functions from disturbances described in Chapter 2:

$$G_{nu} = -C\,(I + GC)^{-1}$$

$$G_{mu} = -C\,(I + GC)^{-1}$$

$$G_{ny} = (I + GC)^{-1}$$

$$G_{my} = -GC\,(I + GC)^{-1}$$

$$G_{ly} = (I + GC)^{-1}\,G.$$

G_{nu} and G_{mu} should have low high-frequency gain and the others (G_{ny}, G_{my}, and G_{ly}) should be small or bounded. Here C denotes the product of the decoupler and the controller.

If the process is completely decoupled the loop transfer function is diagonal

$$GC = \begin{pmatrix} gc_{11} & 0 \\ 0 & gc_{22} \end{pmatrix}.$$

Then the output sensitivity function $S = (I + GC)^{-1} = G_{ny}$, is diagonal with the sensitivity functions of the decoupled loops on the diagonal. This implies that the H-infinity norm of the multi-variable output sensitivity function will be equal to the highest peak of the decoupled output sensitivity functions. The same is true for the output complementary sensitivity function $T = GC(I + GC)^{-1} = -G_{my}$.

A SISO design method that bounds the sensitivity function and the complementary sensitivity function is used for controller design for the decoupled loops. Thus, the H-infinity norm of the multi-variable output sensitivity function and output complementary sensitivity function of the closed loop system are bounded. This implies that the impact of the disturbances n and m on y are bounded.

The design method that is used to tune the decoupled SISO loops minimizes the integrated absolute error under a step load disturbance in each loop. In the multi-variable system this disturbance corresponds to a disturbance that enters the loop between the controller and the decoupler. Such a disturbance is equal to the same disturbance, filtered by the decoupler, entering the system at the process input. If it is assumed that the process is stable and contains no sharp resonances then the decoupler (the adjunct of the process, with some modifications) has the same properties. This is a reasonable load disturbance on which to minimize the integrated absolute error (a low frequency disturbance is good because load disturbances are expected to contain

mostly low frequencies). This means that the transfer function G_{ly} is treated in a proper way.

Since the decoupler is not of high-pass character and the PID controllers are filtered by second order low-pass filters, it is likely that the amplification of high-frequency process output disturbances n and measurement disturbances m to the control signal u are small, which implies that G_{nu} and G_{mu} also are treated in a proper way.

The discussion above shows that the TITO system closed-loop would have the required disturbance attenuation properties if the decoupler and the controllers are designed by the proposed methods.

6.2 Automatic Method

Summary 2.1 is a summary of special requirements of an automatic method. It states that the method should not require qualitative choices to be made. If the method requires parameter tuning by an operator, there must exist default values that always work fairly well. Both the proposed decoupler design method and the proposed PID design method fulfils these requirements (see Chapter 3 and Chapter 4) and since they are performed in a sequential manner in the algorithm the combination of them also does so.

6.3 Examples

The method described above was applied to two processes. The idea of the examples is to show that the method works. The examples are taken from the literature and the proposed method is compared with other methods. The comparison is made to give an idea of how well the proposed methods perform in general, not to start a competition around some specific systems.

EXAMPLE 6.1—THE WOOD-BERRY BINARY DISTILLATION COLUMN PLANT
The Wood-Berry binary distillation column plant has been studied before [Wang *et al.*, 2000], [Åström *et al.*, 2002]. The process has the transfer function

$$G = \begin{pmatrix} \dfrac{12.8e^{-s}}{16.7s+1} & \dfrac{-18.9e^{-3s}}{21.0s+1} \\ \dfrac{6.60e^{-7s}}{10.9s+1} & \dfrac{-19.4e^{-3s}}{14.4s+1} \end{pmatrix}.$$

In [Wang *et al.*, 2000] the process was decoupled and PID controllers with exact gain and phase margins were tuned for each loop in a sequential manner. The controllers were filtered here by second order low-pass filters. The Wang decoupling method is described in [Wang *et al.*, 2000] and is commented on in Section 3.4.

The decoupling method proposed in this chapter gave the decoupler

$$D = \begin{pmatrix} \dfrac{-19.4}{14.4s+1} & \dfrac{18.9e^{-2s}}{21.0s+1} \\ -\dfrac{6.60e^{-4s}}{10.9s+1} & \dfrac{12.8}{16.7s+1} \end{pmatrix}.$$

This corresponds to

$$K = \begin{pmatrix} e^{3s} & 0 \\ 0 & e^{s} \end{pmatrix}.$$

PID controllers of the form (4.2) with second order low-pass filters were tuned for each loop. The controller parameters in the first loop were $K = 0.0480$, $T_i = 10.9$, $T_d = 2.01$. The sign of the controller was negative and the low-pass filter poles were placed in $s = -3.78$. The controller parameters in the second loop were $K = 0.0281$, $T_i = 16.01$, $T_d = 2.93$. The sign of the controller was negative and the low-pass filter poles were placed in $s = -2.58$.

The proposed decoupler and controllers were compared with the previously proposed decoupler and controllers [Wang *et al.*, 2000]. The latter were filtered by low-pass filters with poles in $s = -120$ in the first loop and $s = -5.32$ in the second loop.

The performance of the closed loops under a unit step load disturbance on both channels and measurement noise on both channels were compared. The measurement noise was uniform random numbers with a sampling interval of $0.01s$, a maximum value of 0.02 and a minimum value of -0.02, as it is implemented in Matlab Simulink.

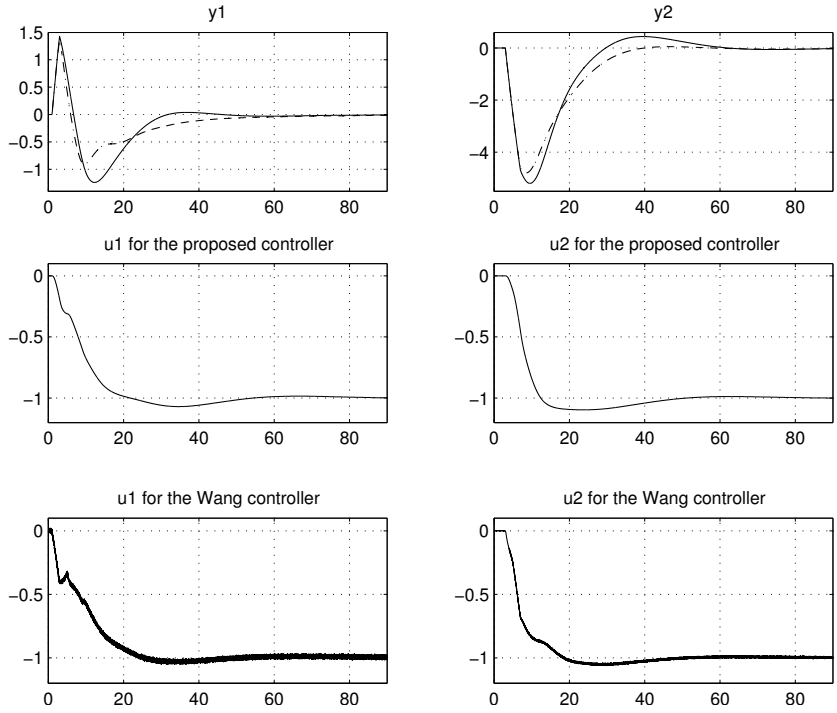

Figure 6.1 Load disturbance responses of the system in Example 6.1 with the proposed method (solid lines) and the Wang method (dot-dashed lines).

The integrated absolute error (see 4.3) under a unit step load disturbance is a good measure of the load disturbance attenuation capacity of the closed loop. The proposed method gave $IAE = 95.7$. The method proposed in [Wang *et al.*, 2000] gave $IAE = 89.1$.

The Wang decoupler and controller gave a smaller integrated absolute error but the amplification of measurement noise was not good (see Figure 6.1). The proposed decoupler and controller gave both acceptable load disturbance attenuation and acceptable sensitivity to measurement noise (see Figure 6.1). □

EXAMPLE 6.2—THE QUADRUPLE TANK

The quadruple tank has been studied before [Åström *et al.*, 2002]. Here it is studied in a non-minimum phase case [Åström *et al.*, 2002]. It has the transfer function

$$G = \begin{pmatrix} \dfrac{0.3333}{s+1} & \dfrac{0.6667}{s^2+2s+1} \\ \dfrac{0.6667}{s^2+2s+1} & \dfrac{0.3333}{s+1} \end{pmatrix}.$$

The simulations in this example were done under the same prerequisites as in Example 6.1. The noise and load disturbances were the same.

In [Åström *et al.*, 2002] static decoupling was used. The method proposed here gave the decoupler

$$D = \begin{pmatrix} 0.3333 & \dfrac{-0.6667}{s+1} \\ \dfrac{-0.6667}{s+1} & 0.3333 \end{pmatrix}.$$

This corresponds to

$$K = \begin{pmatrix} s+1 & 0 \\ 0 & s+1 \end{pmatrix}.$$

The controller parameters in the first loop were $K = 1.66$ $T_i = 2.39$, $T_d = 0.775$ The sign of the controller was negative and the low-pass filter poles were placed in $s = -6.46$. The controller parameters, the filter poles and the sign were the same in the second loop as in the first loop and they were designed with M_s and M_p values of 1.4 in both loops.

In [Åström *et al.*, 2002] static decoupling and some different PI controllers were proposed. The fastest of those were used for comparison. They were filtered with first order low-pass filters with poles in $s = -15$.

The proposed method gave $IAE = 8.69$. The method proposed in the reference gave $IAE = 13.3$. For comparison of control and output signals see Figure 6.2.

□

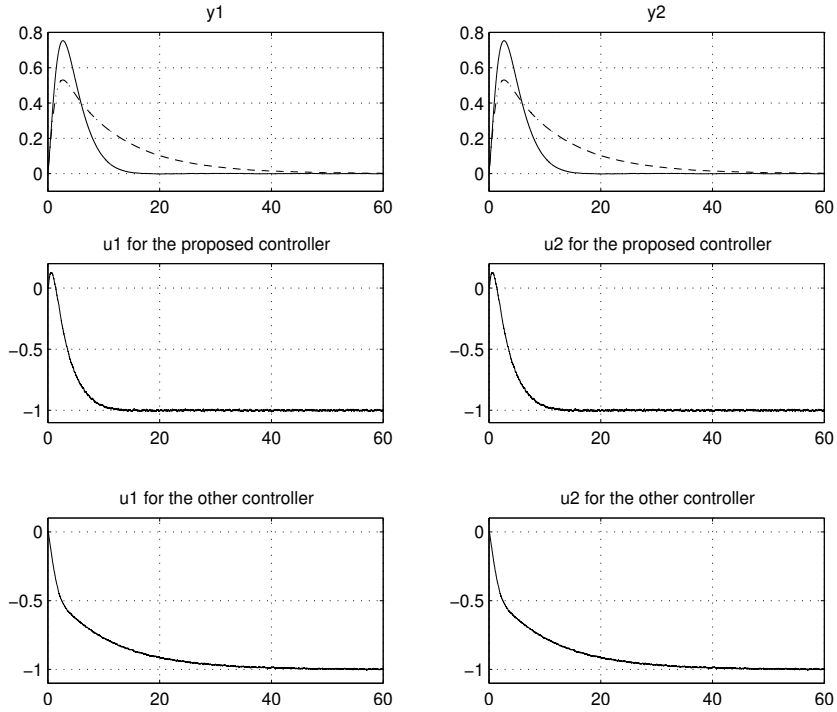

Figure 6.2 Load disturbance responses of the system in Example 6.2 with the proposed method (solid lines) and the Åström method (dot-dashed lines).

It may not be fair to compare PI and PID control, but the example shows that the proposed method works.

6.4 Conclusion

A decoupler and PID controller design method for TITO systems has been proposed. It gives the closed loop system good disturbance attenuation properties and it fulfills the special requirements of an automatic method. The method is meant to form a core in an algorithm

for automatic design and tuning of PID controllers for TITO systems. It was tested by simulation in two examples from the literature with satisfying results.

The next step in the method evaluation would be to perform tests on a real process in an industrial environment. The results of such tests are presented in the next chapter.

7

Decoupling and Tuning - Industrial Example

In Chapter 6 the proposed design methods were tested in simulations. Simulations are a good first test of a method but they should be followed by tests on a real process. In this chapter the results of such tests are presented.

7.1 The Process

At Stora Enso Publication Paper, Hylte Mill in Sweden paper is manufactured from wood and recycling paper. This involves transportation of pulp at several locations in the mill. The transportation is carried out in pipes and the flow is controlled by pumps and valves.

In this case a pipe transporting pulp from a recycling-pulp tower to a pulp container is considered. The flow and the pressure in the pipe are to be controlled. The controller objective is to hold a constant pressure in the pipe and make the flow follow a reference signal from the level controller in the container downstream. There are two actuators on the pipe, a pump and a valve. The pressure and the flow are measured a small distance upstream from the valve. A schematic picture of the process is shown in Figure 7.1.

The picture is schematic, but it contains the relevant parts of the process. The effects of other parts of the process are seen as distur-

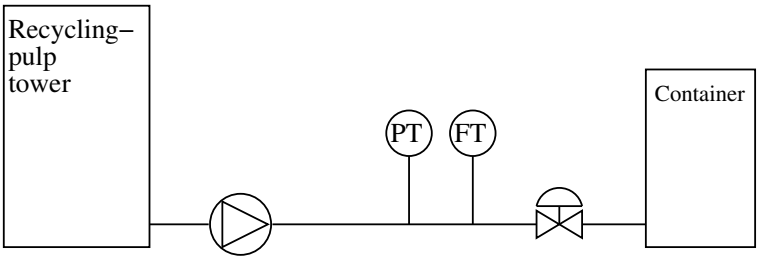

Figure 7.1 A schematic picture of the process.

bances. This process is a TITO system with the two input signals, to the pump and to the valve, and the two measured signals of the pressure and the flow. It has cross couplings since both the pump and the valve affect both the flow and the pressure in the pipe. Since it is a TITO system with cross couplings the methods proposed in previous chapters are appropriate.

A SIEMENS control system is used for control of the part of the mill where the pipe is located. The control system has PID controllers, filter blocks, and delay blocks that may be used for control. The sections below describe how the process is controlled today and how a new controller may be designed using the methods proposed in previous chapters.

The process signals were logged during the experiments as described below. In all the experiment plots the time is given in seconds, the pressure unit is bar, the flow unit is m^3/h, the signal to the pump is given in percent of capacity and the signal to the valve is given in percent of openness. Some of the experiments were performed in closed loop. In these cases the reference signals were noted by hand.

Backlash in the Valve

There is a backlash in the valve which was revealed by a simple experiment. Figure 7.2 shows the input to the valve and the flow response during the experiment. The response in the measured flow is quite slow. The reason for this is that the flow signal is filtered with a low-pass filter with the time constant 20s.

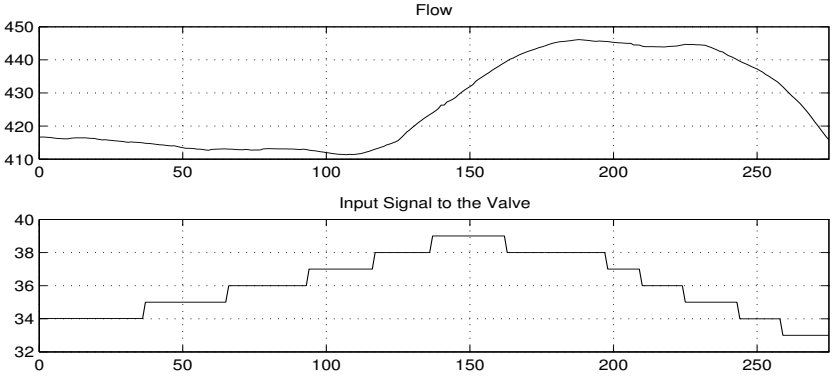

Figure 7.2 Experiment pointing out the backlash.

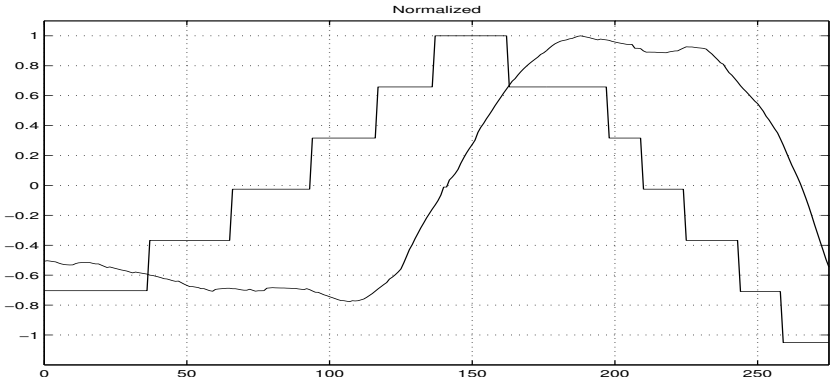

Figure 7.3 The signals of Figure 7.2 have been normalized and are displayed together in this Figure.

In Figure 7.3 the input signal and the output signal of Figure 7.2 have been normalized and put in the same figure to make it easier to draw conclusions from the experiment. The experiment shows that the backlash has a magnitude of about 3 percent.

A backlash of this magnitude limits the achievable performance of

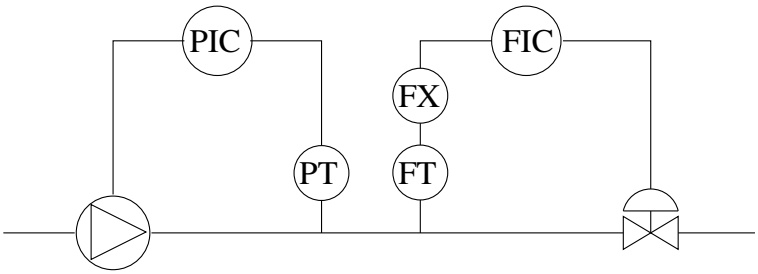

Figure 7.4 A picture of the original controller design.

the system and makes it necessary to use a robust controller. This situation is expected and the design methods proposed in previous chapters give a robust controller.

7.2 Original Design

The original control structure is depicted in Figure 7.4.

The system is controlled by a SIEMENS control system. At present two PI controllers are used to control the process. These are SIEMENS PID controllers with the D part turned off.

A first-order low-pass filter FX with the time constant $T = 20$s is placed in the flow loop before the controller. It is put there to reduce the effects of cross couplings and of the backlash.

The PI controllers are manually tuned and the parameters are $K = 0.2$ and $T_i = 2$s in the pressure controller. In the flow controller the parameters are $K = 0.6$ and $T_i = 28$s. The signals are normalized in the controllers and the controller gains K above are given for the normalized system.

The controller gains are $K = 4$ in the pressure controller and $K = 0.07$ in the flow controller if physical units are used instead of normalized. In the following sections it is assumed that physical units are used everywhere.

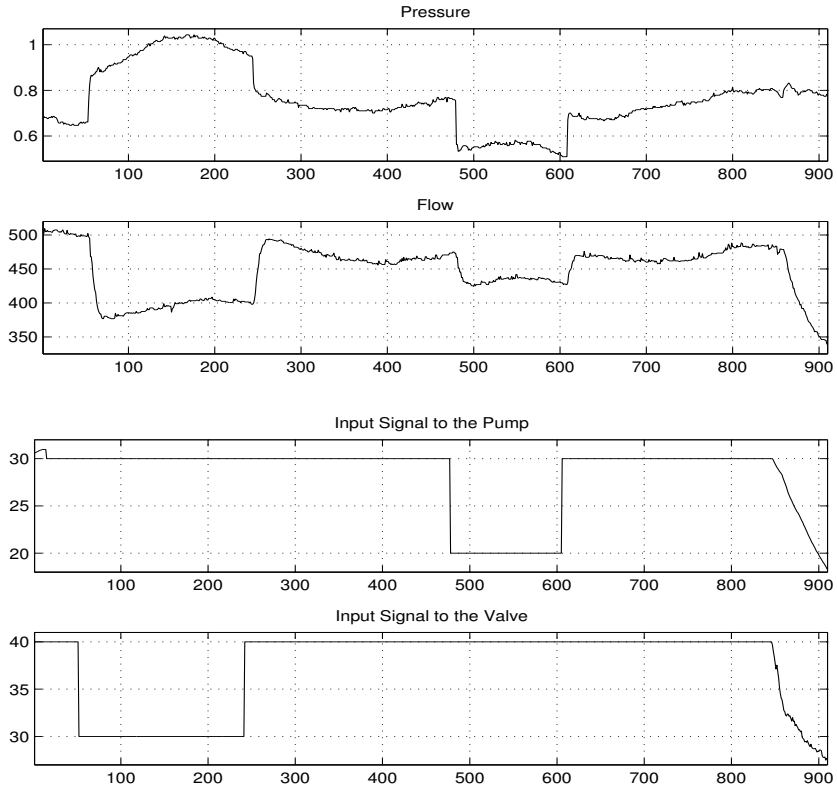

Figure 7.5 Data for system identification

7.3 System Identification

The design methods proposed in previous chapters are model-based. Thus, a model of the process had to be found. Simple experiments were carried out to be used for system identification. Figure 7.5 shows the experimental data that was used. The data was captured in open loop. Two steps, a negative step followed by a positive step in the signal to the valve were performed, and both the pressure and the flow were logged. Then the same was done with the signal to the pump. This

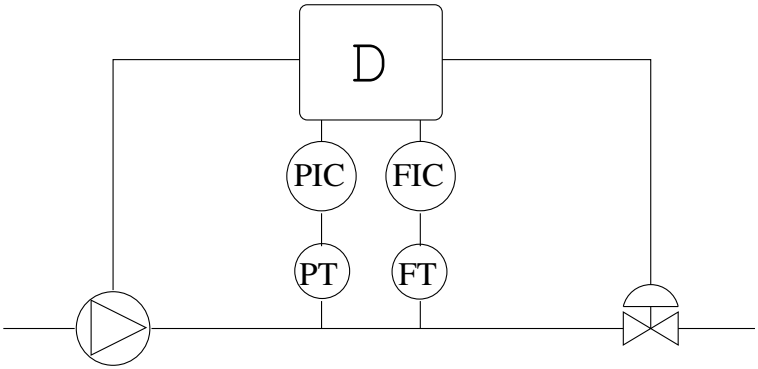

Figure 7.6 A picture of the proposed controller design.

gave the required data and a linear process transfer function matrix was identified as

$$G = \begin{pmatrix} \dfrac{0.018e^{-2s}}{0.42s+1} & \dfrac{-0.02e^{-1.4s}}{0.9s+1} \\ \dfrac{4.1e^{-2.64s}}{3s+1} & \dfrac{12e^{-4s}}{3s+1} \end{pmatrix}.$$

This was done by fitting first-order plus dead-time models to the step responses of the experimental data.

7.4 Controller Design

The proposed control structure is depicted in Figure 7.6.

A decoupler and PID controllers were designed by the methods derived in previous chapters (the columns of the decoupler were also normalized). The decoupler was built up by filter blocks and delay blocks in the SIEMENS control system. It had the structure

$$D = \begin{pmatrix} 2.00e^{-1.36s} & \dfrac{1.06}{0.9s+1} \\ -0.682 & \dfrac{0.950}{0.42s+1}e^{-0.6s} \end{pmatrix}.$$

The PID controllers in the SIEMENS control system are on the following form

$$C = K\left(1 + \frac{1}{T_i s} + T_d \frac{s}{1 + T_d s/v}\right).$$

The PID controllers were filtered by second order low-pass filters

$$F = \frac{1}{(sT_f + 1)^2}.$$

The only difference between this controller structure and the controller structure described in previous chapters is the extra filters on the D-part of the PID controllers. In the experiments the parameters v were chosen sufficiently high to make the filters negligible, such that the developed methods could be used without restrictions.

The proposed method with M_s and M_p values of 1.2 for both controllers gave the controller parameters. In the pressure loop they were $K = 3.60$, $T_i = 1.76$s, $T_d = 0.703$s and $T_f = 0.141$s, and in the flow loop they were $K = 0.0144$, $T_i = 4.49$s, $T_d = 0.703$s and $T_f = 0.0873$s. Furthermore, $v = 100$ was chosen in both loops.

The sampling time of the system was 1s. Since the design was performed in continuous time and the actual control was performed in discrete time some discretization effects were expected. A specific analysis of this was not made. The controller design methods were developed to give a robust controller which handled the various sources of uncertainty.

7.5 Test of Decoupling

The decoupler was first implemented and tested. When the loops are decoupled the pressure controller output should affect only the pressure, and the flow controller output should affect only the flow. The first test was to do some steps in the pressure controller output. Figure 7.7 shows the result. At the approximate times of 37s and 85s the pressure controller output is raised by 3%.

It is obvious from the experiment that the cross coupling from the pressure controller output to the flow is negligible. It is interesting to

see that the decoupler action is as intuitively clear as increasing the signal to the pump and closing the valve a bit, when its input signals demand that the pressure should be raised and flow should be kept at the same level.

In the second test the pressure controller output was kept constant and some steps were performed in the second input signal. At the approximate times of 29s and 82s the flow controller output is raised by 3%. The results are shown in Figure 7.8.

It is obvious that cross coupling from the flow controller output to the pressure is negligible and it is interesting again to see that the action of the decoupler is quite intuitive. When its input signals demand that the pressure should be kept and flow should be raised, it increases the signal to the pump and opens the valve a bit.

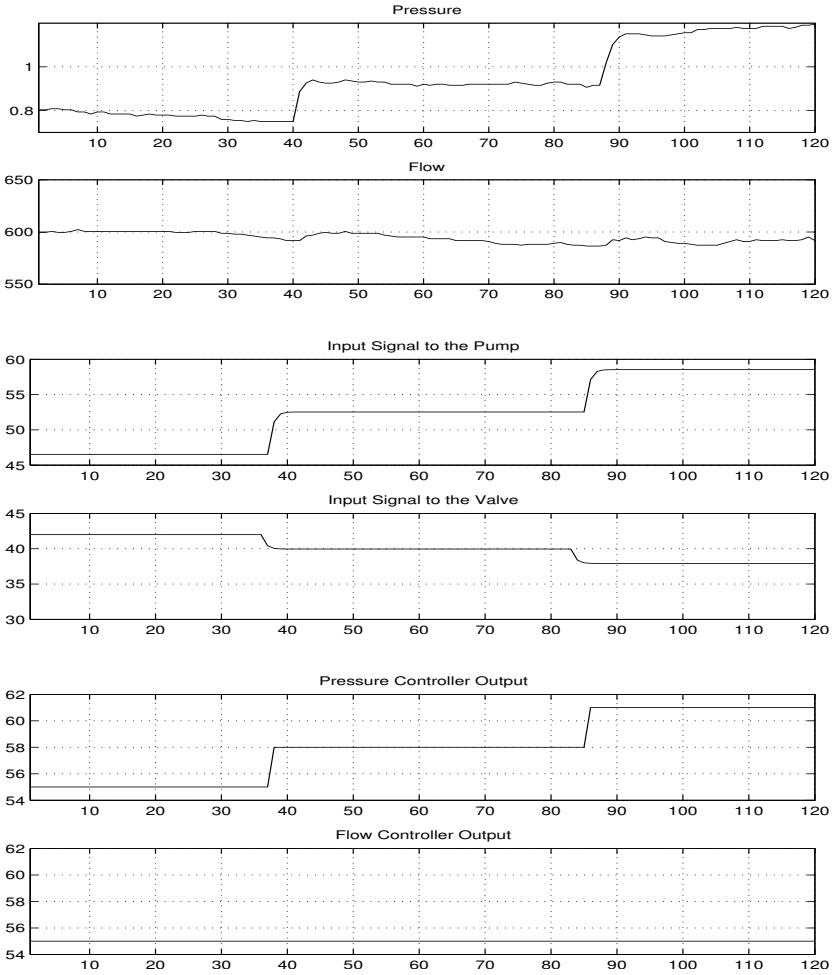

Figure 7.7 Data from test with the decoupled system. At the approximate times of 37s and 85s the pressure controller output is raised by 3%.

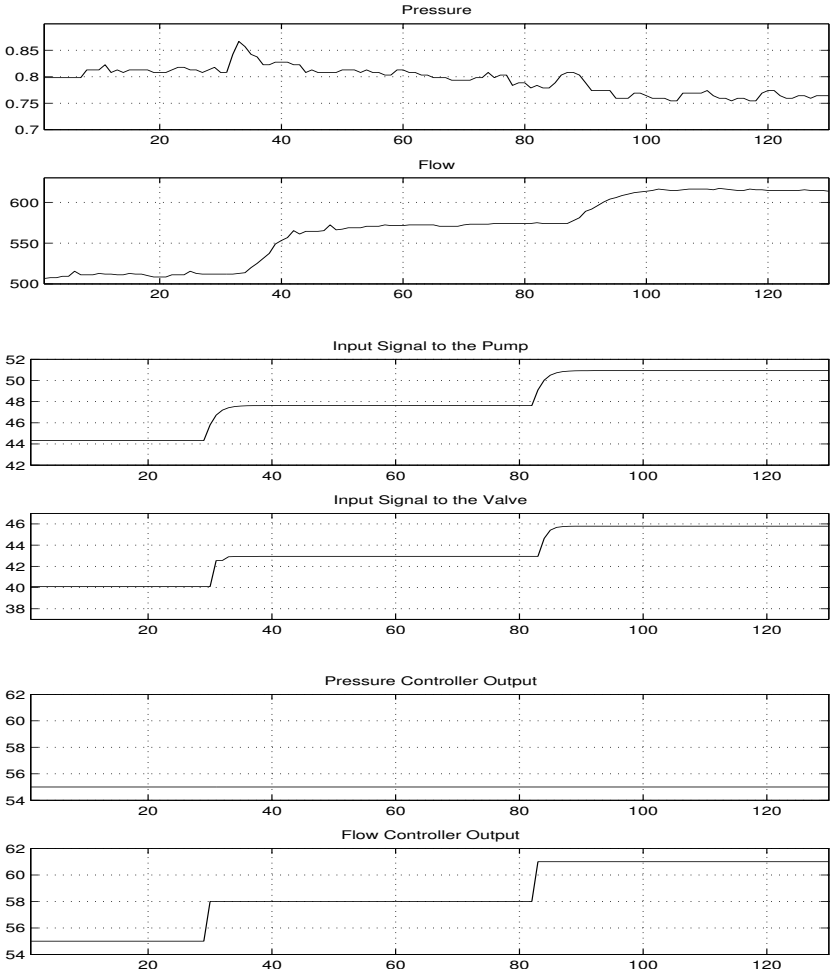

Figure 7.8 Data from test with the decoupled system. At the approximate times of 29s and 82s the flow controller output is raised by 3%.

7.6 Closed-loop Experiments

The proposed controller and the old controller were both tested in closed loop. These kinds of tests are perhaps the most important ones when a controller is to be evaluated.

Flow Setpoint Changes

Under normal operating conditions the pressure should be kept at a constant value while the flow should follow external setpoint changes. Thus, it is interesting to see what happens when this is done in closed loop. It was tested with both the old controller and the new decoupled controller. Figure 7.9 shows the results when the old controller is used. The reference value for the pressure is kept at 0.7 bar and the reference value for the flow is altered in steps of $120\text{m}^3/\text{h}$. The old controller does not manage to keep the pressure close to the reference value. The flow was logged between the low-pass filter and the flow controller when the old controller was used. This makes the flow changes appear to be a bit smoother and slower than they really are.

Figure 7.10 shows the results for a similar experiment with the proposed decoupled controller. This controller manages to keep the pressure much closer to the reference value and is able to perform faster setpoint changes in the flow.

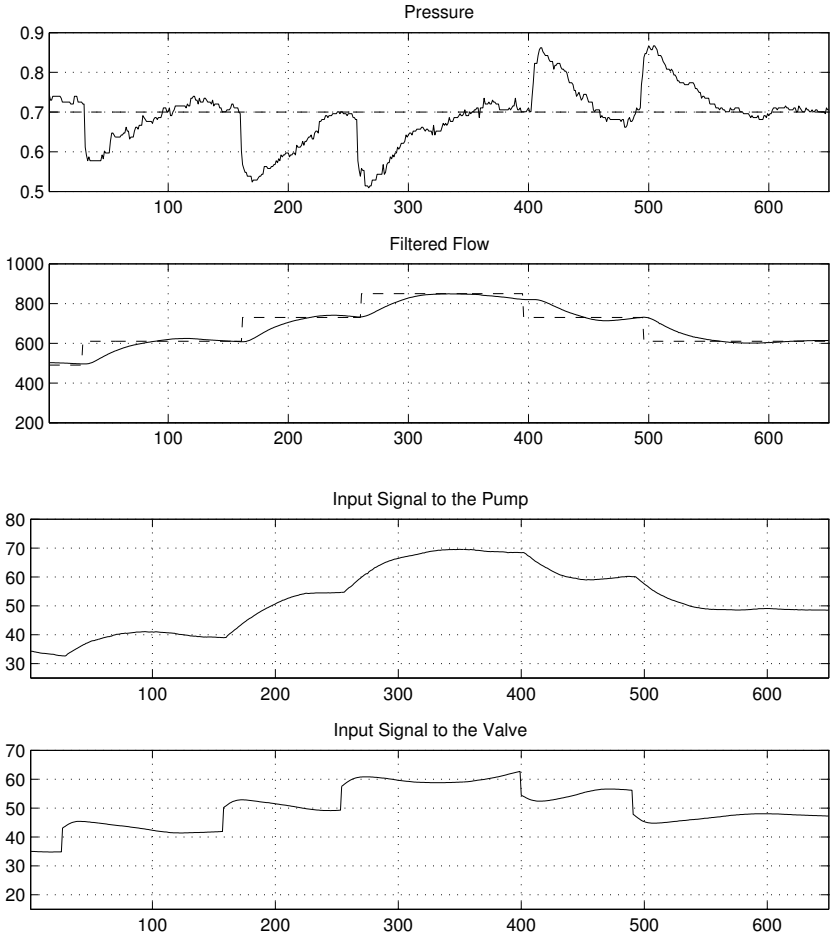

Figure 7.9 Data from reference tests with the old controller. The dashed lines indicate the reference signals for the pressure and the flow. The solid lines show the logged signals.

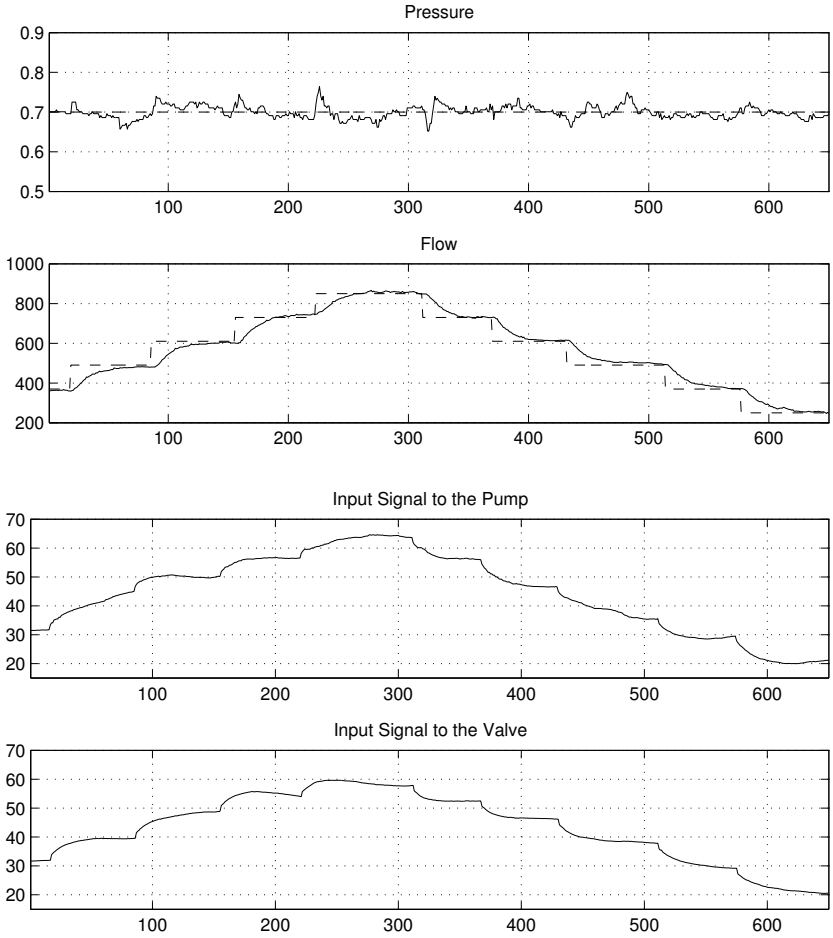

Figure 7.10 Data from reference tests with the proposed decoupled controller. The dashed lines indicate the reference signals for the pressure and the flow. The solid lines show the logged signals.

Load Disturbances

The proposed controllers were optimized to handle load disturbances. A load disturbance that entered the system as an increased flow from a connected pipe was generated. The pipe connects slightly upstream from the spot where the pressure and the flow are measured (see Figure 7.1). The disturbance had an amplitude of approximately $150 m^3/h$. The results of the tests are presented in Figure 7.11 for the old controller and in Figure 7.12 for the proposed new one. In the experiment with the old control controller the reference value for the pressure was 0.7 bar and the reference value for the flow was $400 m^3/h$. The low-pass filtered flow was again logged when the old controller was used. This again makes the flow changes appear to be a bit smoother and slower than they really are. In the experiment with the new decoupled control structure the reference value for the pressure was 0.7 bar and the reference value for the flow was $550 m^3/h$.

The load disturbance enters the system as an increased flow from a connected pipe. This may explain why the effect on the flow is small in the non-decoupled case. In that case the pressure loop is controlled by the pump only. A load disturbance in the form of an altered connected flow has approximately the same effect on the system as an altered signal to the pump. The dynamics of the pressure loop are faster than the dynamics of the flow. When an increased pressure is registered the control signal to the pump is decreased, which will prevent the flow from increasing. Thus, the cross coupling in the process happens to be an advantage in this particular case because it prevents a troubling disturbance in the flow from occurring. Another thing which makes the disturbance in the flow even less visible in this case is the low-pass filter. As mentioned earlier, the flow was logged after the low-pass filter when the old controller was used.

The pressure disturbance attenuation is, however, quite slow in the non-decoupled case and faster in the decoupled case.

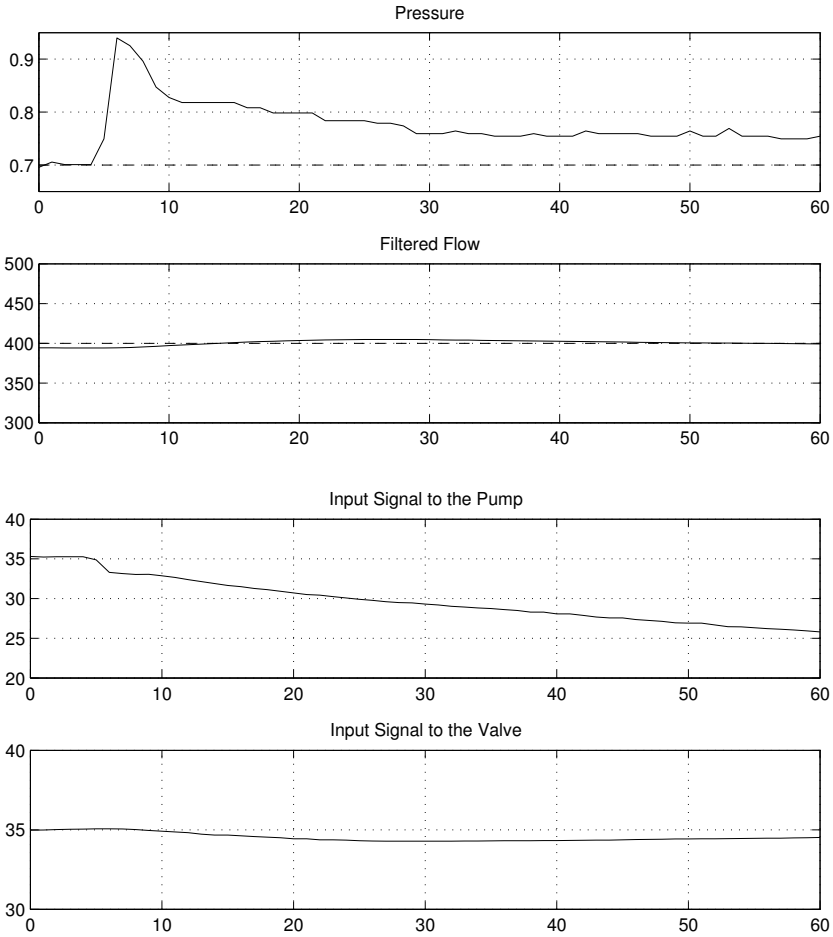

Figure 7.11 Data from load disturbance tests with the old controller. The dashed lines indicate the reference signals for the pressure and the flow. The solid lines show the logged signals.

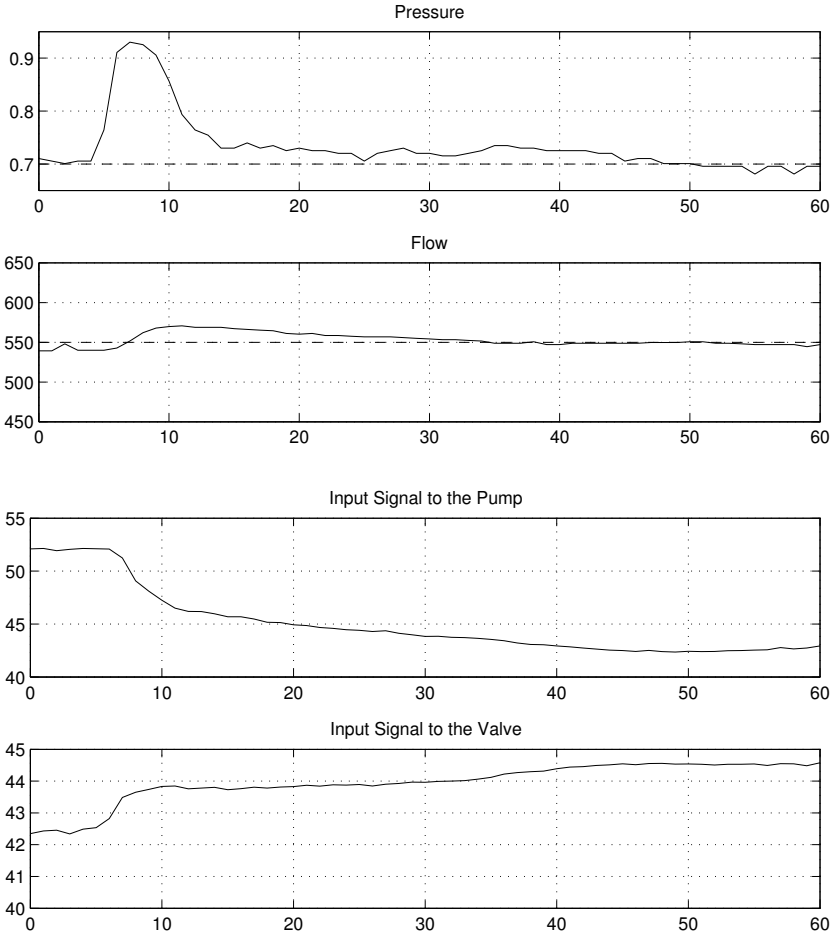

Figure 7.12 Data from load disturbance tests with the proposed decoupled controller. The dashed lines indicate the reference signals for the pressure and the flow. The solid lines show the logged signals.

7.7 Conclusion

Simulations are a good first test of a control strategy, but they should always be followed by real-world experiments, in this case industrial tests. There are differences between simulations and industrial tests. In the latter there are real disturbances, nonlinearities like the backlash and unmodeled dynamics (because it is impossible to find a perfect model). These things are often hard to incorporate in simulations.

In this chapter industrial tests of the developed control strategy were presented. The conclusion of the tests is that the proposed control strategy works well. The pressure loop and the flow loop are decoupled and the disturbance attenuation properties of the controller are good. Further, the controller design methods fulfills the special requirements of automatic methods (see Summary 2.1), which is necessary if they are to be used in an automatic design and tuning algorithm.

8

Summary and Future Work

TITO systems with strong cross couplings are an important class of systems in the process industry. Today these systems are not dealt with in an effective way and there is no satisfactory automatic controller design and tuning algorithm.

A new decoupler design method and a new PID controller design method that can make up a core in such an algorithm have been proposed in this thesis. The methods fulfill the special requirements of an automatic method (see Summary 2.1) and give the closed loop system good disturbance attenuation properties. Further, the closed loop system is decoupled, which is an advantage if the output signals of the system are to follow independent setpoint changes.

The methods were developed to work together in an algorithm, but each of them may also be useful alone in other contexts. Most PID design methods work for processes with simple dynamics but the one proposed in this thesis works for processes with complicated dynamics as well, which of course is a great advantage. The proposed decoupler design method may also be used in other contexts where a decoupler is needed.

The proposed methods were tested in both simulations and tests on a real process in an industrial environment. It was observed that the proposed methods gave the closed loop systems good disturbance attenuation properties, as they were designed to do. During the industrial tests the advantage of having decoupled loops when independent setpoint changes should be followed was also obvious (See Figure 7.9 and Figure 7.10).

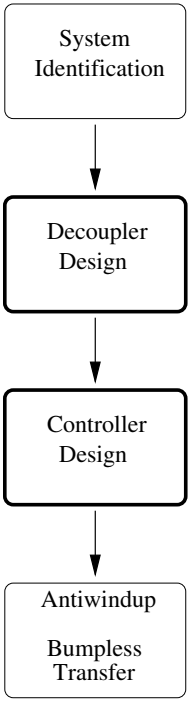

Figure 8.1 The different steps of the design and tuning algorithm are described in this picture.

The specific objective of the work presented in this thesis was to find methods that can make up a core of an automatic controller design and tuning algorithm. The proposed decoupler design method and the proposed PID controller design method can together make up such a core and the objective is thus satisfied.

There are some natural areas for continuation of the presented work. Figure 8.1 shows the different steps of the design and tuning algorithm.

The first step in the algorithm is to find a model of the TITO system by system identification. That part is not covered here and should receive future attention. The following two steps are the decoupler de-

sign and the PID controller design for which methods were proposed in this thesis (see Chapter 3 and Chapter 4). The last step of the algorithm is design for anti-windup and for bumpless transfer between manual and automatic mode. That part is not covered here and should, like the first step in the algorithm, receive future attention.

Another area that has not been covered is pre-filtering of reference signals, which is a common technique that probably could be incorporated in the PID design method fairly easily.

A

Matlab Code for the PID Design Method

```
clear all;
close all;
s=tf('s');

% Load the process
processes;
t=max([(sum(1./abs(pole(G2)))*2+G2.iodelay)*10 ...
       (sum(1./abs(pole(G)))*2+G.iodelay)*10])

Cpoint=-(Ms-Ms*Mp-2*Ms*Mp^2+Mp^2-1)/(2*Ms*(Mp^2-1));
R=(Ms+Mp-1)/(2*Ms*(Mp-1));
figure(1)
r=R;
x=-r:0.001:r;
x(length(x)+1)=r;
y=sqrt(r^2-x.^2);
hold on
plot(x-Cpoint,-y,'LineWidth',2);
hold on
plot(x-Cpoint,y,'LineWidth',2);
r=1/Ms;
x=-r:0.001:r;
x(length(x)+1)=r;
y=sqrt(r^2-x.^2);
hold on
```

```
plot(x-1,-y,'LineWidth',2)
hold on
plot(x-1,y,'LineWidth',2);

hold on
plot(-1/(1-1/Mp^2)+sqrt((1/(1-1/Mp^2))^2-1/(1-1/Mp^2)),0,'o')
hold on
plot(-1/(1-1/Mp^2)-sqrt((1/(1-1/Mp^2))^2-1/(1-1/Mp^2)),0,'o')

axis([-2,0.5,-1.5,1]);
axis square;
grid on


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
oldTf=Tf;
limes=2;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Some constants
Kmax=100;

Kmin=0.01;
IAEmin=inf;
w=logspace(-3,2,2000);
wa=w(length(w));
wb=w(1);

% Preparing the process for simulation
L=G.iodelay;
G.iodelay=0;
Gss=ss(G);
Ga=Gss.a;
Gb=Gss.b;
Gc=Gss.c;
Gd=Gss.d;
G.iodelay=L;

L2=G2.iodelay;
G2.iodelay=0;
G2ss=ss(G2);
```

```
G2a=G2ss.a;
G2b=G2ss.b;
G2c=G2ss.c;
G2d=G2ss.d;
G2.iodelay=L2;

% Test process sign by simulation
y=0;
sim('tests',t);
if y(length(y),2)>0
  sign=1;
else
  sign=-1;
end

% Calculating uncompensated frequency function
Gfr=squeeze(freqresp(G,w))+squeeze(freqresp(G2,w));
ar=arg(Gfr);
% Make sure that the process has positive sign
% during optimization
if sign==-1
  Gfr=-Gfr;
  G=-G;
  G2=-G2;
end

tic
K=Kmax;

% The gridding is repeated twice
nnn=12;
for gr=1:2
  if gr==1
    vidd=logspace(0,3,nnn);
    Tdi=logspace(log10(1/(2*wa)),log10(1/(2*wb)),nnn);
  else
    if finalwidthindex==1
      finalwidthindex=2;
    end
    if finalwidthindex==length(vidd)
      finalwidthindex=length(vidd)-1;
```

```
    end
    vidd=logspace(log10(vidd(finalwidthindex-1))...
                  ,log10(vidd(finalwidthindex+1)),nnn);
    if finalicount==1
      finalicount=2;
    end
    if finalicount==length(Tdi)
      finalicount=length(Tdi)-1;
    end
    Tdi=logspace(log10(Tdi(finalicount-1))...
                 ,log10(Tdi(finalicount+1)),nnn);
  end

  %%% The Td grid loop starts
  Td=Tdi(1);
  Ti=Td/4;
  for icount=1:length(Tdi)
    Td=Tdi(icount)

    %%% The Ti loop starts
    for widthindex=1:length(vidd)
      count=1;
      Ti=vidd(widthindex)*Td/4;
      Tf=50*(1/(2*Td)+real(sqrt(1/(4*Td^2)-1/(Ti*Td))));
      Tf=min([Tf,500]);
      p=1;
      l=R*2;
      K=Kmax;
      CuK=(1+1/(Ti*s)+Td*s)/(1+s/Tf)^(2);
      GOuK=G*CuK;
      G2OuK=G2*CuK;
      GOuKfr=0;
      GOuKfr=squeeze(freqresp(GOuK,w))...
             +squeeze(freqresp(G2OuK,w));

      %%% Determine K

      %%% Decrease K until the system is stable
      K=Kmax;
      GOfr=GOuKfr*K;
      l=abs(min(GOfr+Cpoint));
```

```
while (l<R | stabilitytest(G0fr)==0)...
      && ((K<=Kmax) && (K>=Kmin))
  K=K/2;
  C=K*CuK;
  G0fr=G0uKfr*K;
  l=abs(min(G0fr+Cpoint));
end

%%% Increase K until l is close to Ms
while l>R && K<Kmax && K>Kmin
  K=K*1.01;
  G0fr=G0uKfr*K;
  l=abs(min(G0fr+Cpoint));
end
K=K/1.01;
G0fr=G0uKfr*K;
l=abs(min(G0fr+Cpoint));
C=K*(1+1/(Ti*s)+Td*s);
C=K*(1+1/(Ti*s)+Td*s)/(1+s/Tf)^(2);

% Simulate the controller
if K<Kmax && K>Kmin
  reg=C;
  y=0;

sim('IAE',t);

integral=0;

% IAE
for i=1:length(y)-1
  dt=y(i+1,1)-y(i,1);
  integral=integral+dt*abs(y(i,2));
 end
 qr=integral;

if qr<IAEmin
  finall=l;
  IAEmin=qr
  finalreg=C;
  finalK=K;
```

```
        finalTi=Ti;
        finalTd=Td;
        finalwidthindex=widthindex;
        finalvid=vidd(widthindex);
        finalicount=icount;
        finalTf=pole(C);
        finalTf=finalTf(2);
        spar=[K Ti Td finalTf];

      end
    end
  end
end
end
toc

% Restore the process sign
if sign==-1
  Gfr=-Gfr;
  G=-G;
  G2=-G2;
end

Tf=oldTf;
```

References

Åström, K. J. and T. Hägglund (2005): *Advanced PID Control*. ISA - The Instrumentation, Systems, and Automation Society, Research Triangle Park, NC 27709.

Åström, K. J., K. H. Johansson, and Q. Wang (2002): "Design of decoupled PI controller for two-by-two systems." *IEE Proceedings on Control Theory and Applications*, **149**, January, pp. 74–81.

Åström, K. J. and B. Wittenmark (1990): *Computer Controlled Systems—Theory and Design*. Prentice-Hall, Englewood Cliffs, New Jersey.

Bialkowski, W. L. (1993): "Dreams versus reality: A view from both sides of the gap." *Pulp & Paper Canada 94:11*.

Goodwin, G. C., S. F. Graebe, and M. E. Salgado (2001): *Control System Design*. Prentice-Hall, Inc., Upper Saddle River , New Jersey 07458.

Hägglund, T. and K. J. Åström (2002): "Revisiting the Ziegler-Nichols tuning rules for PI control." *Asian Journal of Control*, **4:4**, pp. 364–380.

Hägglund, T. and K. J. Åström (2004): "Revisiting the Ziegler-Nichols step response method for PID control." *Journal of Process Control*, **14:6**, pp. 635–650.

Lancaster, P. (1969): *Theory of Matrices*. Academic Pres, Inc., 111 Fifth Avenue, New York, New York 10003.

Maciejowski, J. (1989): *Multivariable Feedback Design*. Addison-Wesley Publishing Company, Inc.

Panagopoulos, H., K. J. Åström, and T. Hägglund (2002): "Design of PID controllers based on constrained optimisation." *IEE Proc. Control Theory Appl.*, **149:1**, pp. 32–40.

Skogestad, S. (2001): "Probably the best simple PID tuning rules in the world." In *AIChE Annual meeting, Reno, NV, USA*.

Wang, Q.-G., B. Huang, and X. Gu (2000): "Auto-tuning of tito decoupling controllers from step tests." In *ISA Transactions 39 (2000) 407-418*.

Wang, Q.-G. and Y. Yang (2002): "Transfer function matrix approach to decoupling problem with stability." In *Systems & Control Letters 47 (2002) 103-110*.

Wang, Q.-G., Y. Zhang, and M.-S. Chiu (2002): "Decoupling internal model control for multivariable systems with multiple time delays." In *Chemical Engineering Science 57 (2002) 115-124*.

Wang, Q.-G., Y. Zhang, and M.-S. Chiu (2003): "None-interacting control design for multivariable industrial processes." In *Journal of Process Control, Volume 13, Issue 3, April 2003, Pages 253-265*.

Zhou, K. and J. C. Doyle (1998): *Essentials of Robust Control*. Prentice-Hall, Inc., Upper Saddle River , New Jersey 07458.

Ziegler, J. G. and N. B. Nichols (1942): "Optimum settings for automatic controllers." *Trans. ASME*, **64**, pp. 759–768.