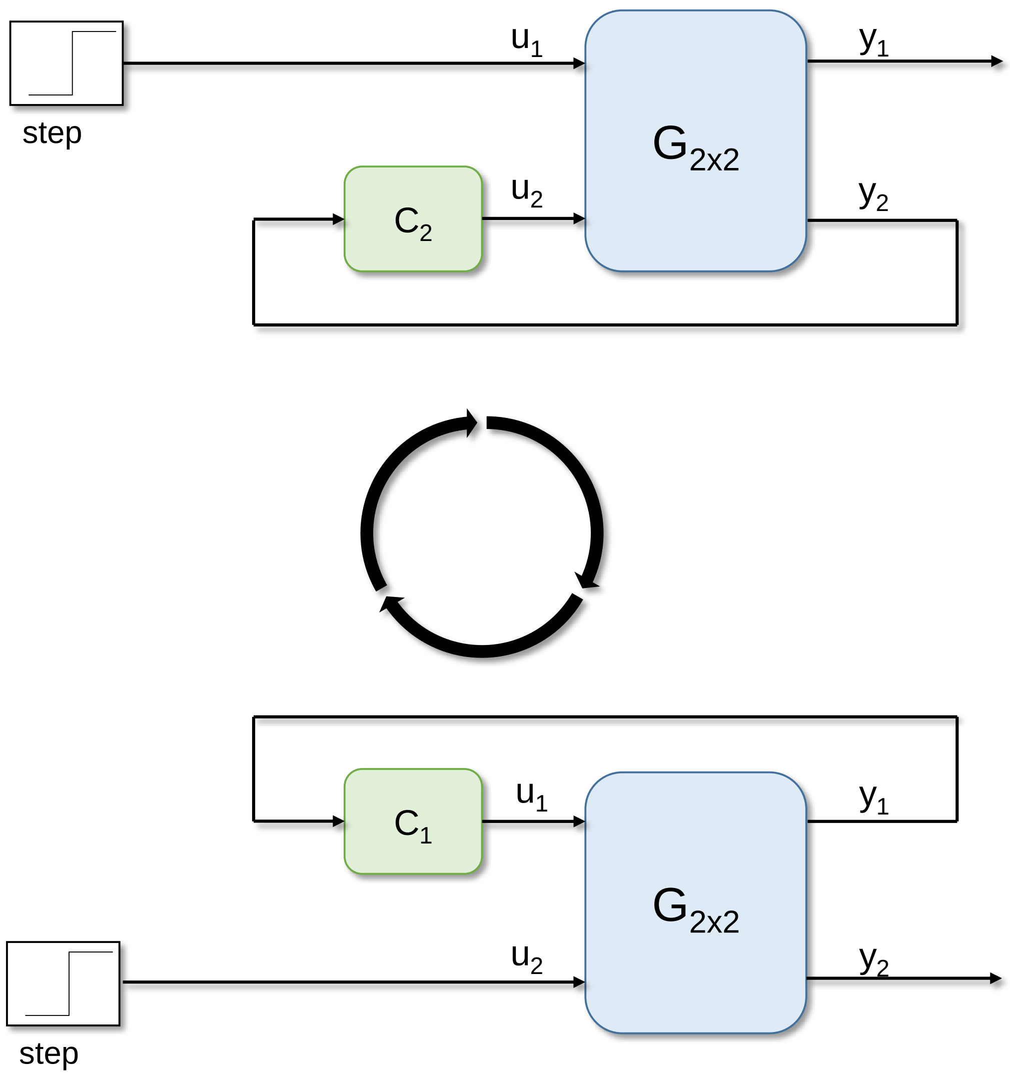
CHALMERS
UNIVERSITY OF TECHNOLOGY



Investigating an iterative method for tuning decentralised PI-controllers in TITO systems

Master's thesis in Systems, Control and Mechatronics

Andreas Gustavsson

Department of Signals and Systems
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2016

Master's thesis EX059/2016

Investigating an iterative method for tuning decentralised PI-controllers in TITO systems

ANDREAS GUSTAVSSON

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department Signals and Systems
Division of Automatic control, Automation and Mechatronics
Chalmers University of Technology
Gothenburg, Sweden 2016

Investigating an iterative method for tuning decentralised PI-controllers in TITO systems
ANDREAS GUSTAVSSON

Supervisor: Johan Rönnberg, Perstorp AB
Examiner: Torsten Wik, Chalmers

Master's Thesis EX059/2016
Department Signals and Systems
Division of Automatic control, Automation and Mechatronics
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: A schematic representation of the proposed iterative method when tuning single loop controllers for TITO systems.

Typeset in LaTeX
Gothenburg, Sweden 2016

Investigating an iterative method for tuning decentralised PI-controllers in TITO systems
ANDREAS GUSTAVSSON
Department Signals and Systems
Chalmers University of Technology

Abstract

Within process industry many MIMO processes are controlled using several single loop controllers (decentralised control). This project is limited to two-by-two processes (TITO). In order to account for interacting dynamics when tuning two single loop controllers in a TITO system the control engineers at the specialty chemicals manufacturer Perstorp AB have proposed an iterative method. A controller is tuned based on step response tests with the other controller in closed loop. Since any change in the parameters of a controller will affect the step response used to tune the other controller through the interacting dynamics, both controllers must be iteratively tuned and re tuned until the tunings converge. the iterative method has been investigating using two single loop design methods which generate PI-controllers; Skogestad IMC (SIMC) and λ-tuning.

By using the convention of Effective open-loop process (EOP) it is shown that the step response tests within the iterative method can show complex dynamics of high order. This proved to be a limitation in SIMC which is ruled out as a viable tuning method within the iterative method.

For λ-tuning an algorithm for approximating step response tests as first order was developed. The algorithm was compared with approximations made by process control engineers at Perstorp through visual trend curve inspection. The results from the algorithm differed somewhat from the control engineers. Closed loop behaviour of λ-tuned controllers was better when approximations from the algorithm were used, suggesting that the process control engineers should reconsider the way they intuitively reduce dynamics to first order for subsequent λ-tuning.

The iterative method was investigated using simulations in MATLAB. For evaluation, two simple TITO systems are used; a pipe-flow-pressure system and The Wood and Berry binary distillation column. Both systems have four first order transfer functions with or without time delays. The dynamics in step responses are reduced to first order using the developed algorithm, and PI-controllers are tuned based on λ-tuning. For the pipe-flow-pressure system 100 cases were investigated, and for the Wood and Berry system 121 cases were investigated. Each case corresponds to a specification of the tuning parameter λ for each loop. The iterative method method converges in all simulated cases, usually within 2 or 3 iterations for each controller. The closed loop behaviour, tested through step response tests in setpoints, are not always in line with what is desired in λ-tuning, but always stable.

No proof has been found when or when not the iterative method converges for TITO systems. It is shown that closed loop stability of a general TITO system without time delays is achieved if and only if the single loop controllers tuned in the iterative method are tuned for single loop stability. Thus the iterative method is a viable method when tuning with only stability in mind.

Keywords: PI-controllers, iterative method, Skogestad IMC, λ-tuning, Effective Open-loop Process, EOP, decentralised control

Contents

List of Figures

List of Tables

1 Introduction

Many real world dynamical systems are characterised as multi-input multi-output (MIMO) with interactions and couplings between different input and output variables. In order to optimally account for these interactions in a control system, multivariable controllers are needed. However, 95 % of the controllers used in process control are PI- or PID-controllers [1]. Being typical examples of single-input single-output (SISO) controllers, i.e. single-variable, they are still used for multivariable/MIMO systems giving rise to the term *decentralised control* (also referred to as *multiloop control*).

Centralised control is the family of controllers where controller output is based on a central decision taking into account multiple variables and objectives. Applying SISO-controllers to a MIMO-system means that each control loop only accounts for its own single-variable objective. In other words they operate in a decentralised fashion [2]. In the context of linear systems theory this can be viewed as a feedback controller, $C(s)$, represented in matrix form with nonzero elements only along its main diagonal as can be seen in Figure 1.1, assuming controller output u_i is paired with measured plant output y_i. This structure can always be achieved for a decentralised controller by interchanging rows in the transfer function matrix $G(s)$. A plant, $G(s)$, with n variables to be controlled have n decentralised SISO-controllers.

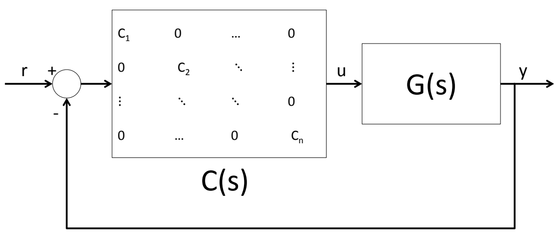


Figure 1.1: Visualisation of a decentralised control system. G(s) and C(s) are transfer function matrices of the plant and the controller, respectively, and r, u, and y are vectors.

1.1 Background

Even though there are a wide availability of multivariate controllers with mature theoretical background the dominating controllers in process industry are PI- or PID-controllers, even for MIMO-systems [3]. The main reasons being their ease of implementation and

PI/PID-controllers in a decentralised control scheme are more easily understood by people without a thorough knowledge about control systems (e.g. the workforce responsible for the day to day operation and supervision at a process industry plant) [4]. Despite the wide use of decentralised control the number of applicable tuning methods is limited [3].

The company Perstorp AB is a world leading supplier and innovator of specialty chemicals. Located in Stenungsund, Perstorp Oxo AB is a specialty chemicals plant producing mainly different aldehydes, organic acids, alcohols, plasticizer, biodiesel and pthalic anhydride [5]. In many ways a typical example of a process industry plant.

When the control engineers at Perstorp design and tune the control systems at different plants they start off with simple solutions which in most cases lead to a control philosophy based on decentralised control. It is also a common scenario that the engineers are given a problem with an already present decentralised control philosopy where they are tasked with retuning the SISO control loops due to poor controller performance of the present tunings without being able to change the control philosophy (e.g. loop pairings). If each SISO-controller (most often of PI-type) in a decentralised control scheme is designed loop-by-loop without accounting for interactions, due to possible multivariable dynamics, the resulting system when all control loops are closed might result in "bad" (destabilizing/oscillatory behaviour) overall system performance and low robustness. A proposed outline for tuning these systems is by tuning a given PI-controller based on the dynamics between the given input/output pair with all other control loops closed. The main notion is that this would capture the cross-interactions in the presence of MIMO-dynamics and making this part of the SISO-tuning problem.

1.1.1 The iterative method

Perstorp suggests an iterative tuning method which will henceforth be referred to as the "The iterative method". The main idea is to treat a MIMO system with n controlled variables as n single loop design problems. The notion is to tune a PI-controller for a controlled variable based on a step response test on the system with all other controllers in closed loop, and then repeat this procedure for all other PI-controllers. However, when tuning subsequent control loops this will change the dynamics of loops already tuned, thus requiring retuning. This implies that all PI-controllers must be tuned and re-tuned iteratively until the controller parameters converge and remain approximately unchanged.

1.2 Problem description

The proposed iterative method will be investigated. However, it is necessary to be more specific what the method entails to enable an analysis. As stated, the objective is to tune controllers for n single loop design problems. For that purpose it must be defined what single loop design techniques will be employed, and what their tuning objective is. The term "iterative" presupposes that iterations will take place and an answer to how many iterations to be carried out must be answered. It has been suggested that the n single loop design problems will be iterated until controller parameters converge to some value. This give rise to a new issue in determining when convergence happens or not based on, for example, the dynamics in the system studied and single loop design technique used.

Assuming the iterative method for some plant have PI-controller parameters that have converged an investigation of the resulting closed loop performance must be conducted. Specifications on what is "good", respective "bad" performance are needed. In order to analyse the iterative method without applying it on real life systems, computer simulations must be made. Thus, a major task will be in making an algorithm of the iterative method for use in simulations.

1.3 Purpose and Aims

The objective with this report is to help the control engineers at Perstorp with insight and understanding of the problem of using PI-controllers for MIMO-systems. It is thus important during application of the iterative method to mimic the design philosophy used by Perstorp when tuning each PI-controller.

The main goals when investigating the iterative method is (1), to determine if and when convergence of controller parameters occur, (2), whether a tuning series that has converged will yield a decentralised control scheme with desired stability and performance properties (response to setpoint changes) and (3), investigate if the single loop tuning methods used within the iterative method reflects the philosophies used by the control engineers at Perstorp.

1.4 Delimitiations

In order to deal with interactions due to MIMO-dynamics when using SISO-controllers in a decentralised scheme it is common to employ a decoupler [6]. In short, decouplers can be said to to pre-filter all control signals such that in the ideal case all cross-interactions are perfectly counteracted. Using a decoupler will result in an overall MIMO-type controller, though tuned by SISO controllers, and with this thesis being focused on utilizing only SISO-control the use of decouplers in the control design will be excluded.

Another important aspect in decentralised control is input/output pairing, i.e. which measured process variable should be used by which controller [7]. This freedom of design will not be considered within the scope of this project and only given pairings will be assumed. The reason is that this is usually the case for the control engineers at Perstorp when they are given a decentralised control problem.

Since the the control philosophy (i.e. input/output pairing) is fixed, for ease of reference in the sequel, when a dynamic two-input two-output (TITO) system is given by a square transfer function matrix between input signal-vector and output signal-vector, i.e.

$$\underbrace{\begin{bmatrix} y_1(s) \\ y_2(s) \end{bmatrix}}_{\text{y(s)}} = \underbrace{\begin{bmatrix} g_{11}(s) & g_{12}(s) \\ g_{21}(s) & g_{22}(s) \end{bmatrix}}_{\text{G(s)}} \underbrace{\begin{bmatrix} u_1(s) \\ u_2(s) \end{bmatrix}}_{\text{u(s)}} \tag{1.1}$$

it is assumed that y_1 is paired with u_1, y_2 paired with u_2 and so on.

The systems studied will be TITO in order to keep results clear and lucid. Also, everything will be assumed to be linear, which enables the use of linear systems theory. PI-controllers constitute the large bulk of controllers used in Perstorps plants so the SISO-controllers studied will be limited to PI and subsets thereof (pure P- and/or pure I-controllers).

2 Preliminaries and theoretical background

This chapter will delve into the theoretical framework used to analyze and interpret the findings. Justifications of choices made will also be stated. It is assumed that the reader is familiar with linear systems theory and control concepts as taught in a basic control course. Knowledge about linear multivariable control systems is also assumed.

2.1 Decentralised control

As stated in the previous chapter the topic of decentralised control deals with the concept of using n single loop controllers in order to control n variables in a multivariable system. There are two main classes of design methods: *independent design* and *sequential design* [8].

In independent design each individual controller is designed seperately based on information for the given input/output pair using some SISO design method. The effect of multivariable interactions on controller performance is then considered through robustness measures on the overall MIMO-system, which introduce bounds on the allowable SISO-designs. Thus, independent design requires some model of the system, such as a transfer function matrix, in order to mathematically consider interactions in robustness calculations. Independent design can be said to be n SISO-design problems coupled with one multivariable robustness problem[4].

Sequential design however is based on sequentially designing single variable control loops after closing already designed loops. For example, if we choose to design C_1 in Figure 1.1 first this tuning is based on a fully open loop system. When C_2 is subsequently designed the control loop with C_1 is closed. Sequential design is n SISO-design problems (no multivariable problem as was the case for independent design).

2.2 The PI-controller

The PI-controllers considered here are on the form

$$c(s) = K_c \frac{T_i s + 1}{T_i s}. \tag{2.1}$$

As stated, the controller-type studied will be PI-controllers and its subsets, P-controllers and I-controllers. In industry, PI-controllers are usually implemented in series form according to Equation (2.1). The controller gain is K_c and T_i is the integral time. A P-controller can be generated from a series form PI-controller by taking the limit as $T_i \to +\infty$. To get an I-controller the limits $T_i \to 0$ and $K_c \to 0$ are taken but with the constraint that the ratio $\frac{K_c}{T_i}$ is finite and nonzero.

2.3 A scalar relationship showing the effect of multivariable dynamics

For TITO systems with deccentralised control, by closing one control loop a scalar transfer function that contains information about the dynamical interactions between the two single loop controllers can be formed. Consider a general TITO system with transfer function matrix $G(s)$ and the corresponding decentralised controller matrix $C(s)$, i.e.

$$G(s) = \begin{bmatrix} g_{11}(s) & g_{12}(s) \\ g_{21}(s) & g_{22}(s) \end{bmatrix} \qquad C(s) = \begin{bmatrix} c_1(s) & 0 \\ 0 & c_2(s) \end{bmatrix} \tag{2.2}$$

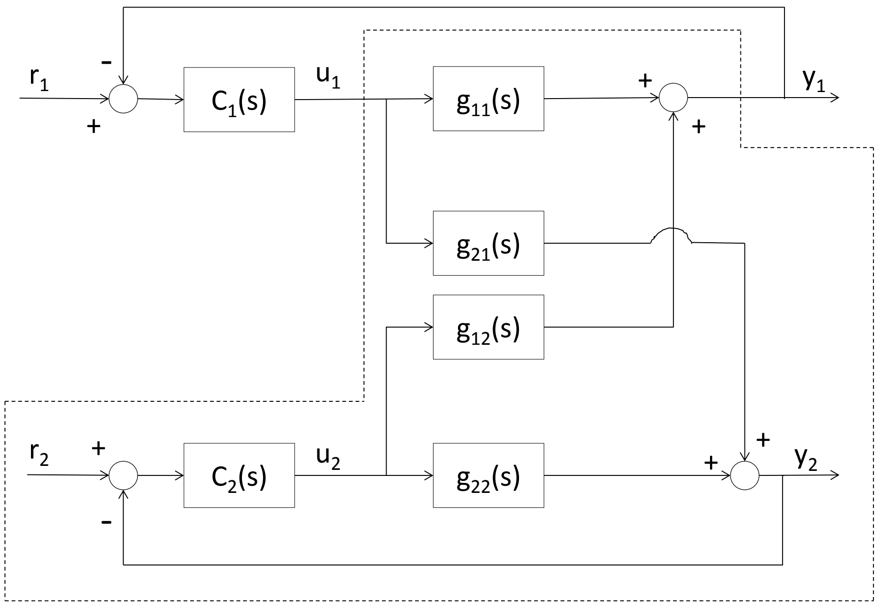


Figure 2.1: Block diagram representation of a general TITO control system with all transfer functions and signals being scalars.

Figure 2.1 depicts the block diagram of this 2-by-2 system. It is clearly shown how the two control loops interact with each other through the blocks $g_{12}(s)$ and $g_{21}(s)$. The open loop response from u_1 to y_1 is given by $g_{11}(s)$ which carries no information about possible interactions from the other loop. However if we close loop 2 the dynamics between u_1 to y_1 will also contain the interaction effects: If u_1 is perturbed this will not only show a response in y_1 through $g_{11}(s)$ but y_2 will also be affected through g_{21} and with loop 2 being closed this will necessarily perturb the control signal u_2. This will, in turn, give a response in y_1 through g_{12}.

2.3.1 The effective open-loop process, EOP

By looking at the relationship between u_1 and y_2 with control loop 2 closed, we have effectively included the cross-diagonal interaction terms (g_{21} and g_{12}) from the transfer function matrix of the system in the dynamic relationship between u_1 and y_1. This transfer function is considered as the effective open-loop process (EOP) since it is capturing the actual (i.e. "effective") dynamic transmission from u_1 to y_1 for the system with control loop 2 closed, as opposed to the "true" open loop process g_{11} [9]. The EOP transfer function between u_1 to y_1 is a combination of the blocks contained within the dashed lines in Figure 2.1 and it is given by:

$$g_{1,EOP}(s) = g_{11}(s) - \frac{g_{12}(s)c_2(s)g_{21}(s)}{1 + g_{22}(s)c_2(s)}. \tag{2.3}$$

The exact same reasoning can be applied to the dynamics between u_2 and y_2 by instead closing loop 1. The resulting EOP transfer function is then

$$g_{2,EOP}(s) = g_{22}(s) - \frac{g_{21}(s)c_1(s)g_{12}(s)}{1 + g_{11}(s)c_1(s)}. \tag{2.4}$$

2.3.2 Reducing a TITO controller design problem to two single loop design problems

In Figure 2.1 if the set-point for loop 2, r_2, is zero[1] this signal is effectively removed as an exogenous input and can be treated as a property within the system. This means that for the sub-system within the dashed lines in the figure the only input and output signals are u_1 and y_1. Thus, the transfer function of this dashed region is completely captured by the scalar transfer function $g_{1,EOP}$. An equivalent treatment (i.e. assuming r_1 is set to zero) can be made for $g_{2,EOP}$. This goes to show that the TITO problem can be reduced to the analysis of two scalar transfer functions which still captures the interactions from multivariate dynamics.

Using the framework of effective open-loop processes is the key in the iterative method enabling the treatment of the TITO design problem as two single loop design problems. EOPs can be readily extended to *nxn* MIMO systems [10]. For further information on using effective open-loop processes as a basis for controller design consult [11, 12, 13, 14, 15].

2.4 Single loop design methods in order to generate PI-controllers

The previous section showed how a TITO-system with decentralised control can be treated as two single loop problems still accounting for interactions. The next topic will be PI-controller design based on these single loop-problems. Two design methods formulated within the framework of internal model control (IMC) has been chosen, λ-tuning and SIMC. The appeal with internal model control is that the controller is designed by specifying the desired closed loop response to a set-point change [16]. Controller design then have clear connection to how the real world system should behave in closed loop. This is closely related to how the control engineers at Perstorp deal with a control problem since specifications that must be fulfilled is usually related to the closed loop response. Information on internal model control and applications can be found in [17, 18].

2.4.1 λ-tuning

The purpose with this tuning method is to specify the closed loop transfer function as a first order system plus time delay (FOPTD) with a time constant λ, i.e. a transfer function with one pole in $-\frac{1}{\lambda}$ and unity steady state gain according to

[1] Any constant value can be applied if it is further assumed the signal has been applied for an infinite time (the system is at steady state).

$$g_c(s) = \frac{g(s)c(s)}{1 + g(s)c(s)} = \frac{1}{\lambda s + 1} e^{-\theta s}. \tag{2.5}$$

In order to generate (approximately) this closed loop transfer function with a PI-controller the process $g(s)$ is modeled as FOPTD i.e.

$$g(s) = \frac{K}{sT + 1} e^{-\theta s} \tag{2.6}$$

Where K is the steady state gain, T is the time constant and θ is the time delay.

Then to achieve the closed loop response in Equation (2.5) the PI-controller parameters are chosen as a function of the process model (Equation (2.6)) according to

$$K_c = \frac{T}{K(\lambda + \theta)}, \qquad T_i = T. \tag{2.7}$$

The tuning parameter λ only affects the controller gain, the integral time T_i is always set equal to T. For a full derivation of the λ-tuning method see Appendix A.

2.4.2 SIMC - Skogestad IMC

If the transfer function of the process to be controlled is known SIMC provides an analytical and straighforward method to design a PID-controller. When generating PI-controllers the method has a lot in common with λ-tuning. SIMC specifies a closed loop transfer function with first order dynamics and by also approximating the transfer function of the process as a FOPTD model a PI-controller is generated by specifying the closed loop time constant τ_c (c.f. λ). An important difference to λ-tuning is how the integral time, T_i in Equation (2.1), is chosen.

One of the main aspects of SIMC is a template on how to carry out the approximation of the (possibly high order) process transfer function to a FOPTD model. Neighbouring stable poles and zeros are cancelled and approximated as either a constant gain or a fast pole. The time constant of the FOPTD model is then estimated as the largest remaing time constant (slowest pole) plus half the next largest time constant. The other half of the next largest time constant and the time constants of the remaining faster poles are summed with non-minimum phase zeros (zeros with positive real part) to form the time delay of the FOPTD approximation. SIMC was first formulated in [19] and further information and suggestions on improvements are provided in [20, 21].

2.5 The iterative method for TITO systems

The main idea of the iterative method as proposed by Perstorp is to be able to tackle a multivariable control problem using SISO-design methods. By having all control loops closed, except the one being tuned, and performing a step response test on the system we will gain information about the interactions caused by the multivariable dynamics. This will capture the actual operating conditions in the test and we will have the most relevant information about the system when tuning the controller compared to having all other loops open [22]. For a TITO system, the iterative method means that one of the two control loops is closed and the other one remains open resulting in step response tests of the EOPs given in Equations (2.3) and (2.4).

The iterative method has been suggested as a tuning method in [15, 23]. Both articles use SISO design methods based on loop shaping and specifying gain and phase margins for the SISO design problems. However, the more general case of PID-controllers are considered, not only PI-controllers. Since loop shaping is carried out the transfer functions of the TITO system must be known or modeled to use the procedures proposed in the articles.

The iterative method when no control loops are closed

It is important to note that having one control loop closed implies that the parameters of this controller must already have been determined. In other words there must already be controllers present. However for completeness, the case were there are no apriori tuned PI-controllers will be made part of the iterative method. In this instance, tunings will start based on sequential design as described in Section 2.1.

2.5.1 Controllers affect each other when tuned: the need for an iterative procedure

The term *iterative* stem from the fact that controller design is based on EOPs which contain dynamics subject to change. If control loop 2 in a TITO system is tuned and new parameters are assigned to controller $c_2(s)$ this will change $g_{1,EOP}(s)$ (as evident by Equation (2.3)). Since $c_1(s)$ was tuned based on the previous $g_{1,EOP}(s)$ the controller $c_1(s)$ should be retuned with a step response of the new EOP. If this leads to new controller parameters, $g_{2,EOP}(s)$ will, in turn, change and so on and so forth. An important property of the iterative method is if controller parameters converge or not. Even if the parameters have converged there is no guaranteed performance of the closed loop TITO system, or even stability. Thus, this also requires an investigation.

2.5.2 λ-tuning and SIMC within the iterative method

For TITO systems the iterative method can be summarized as the iteration of two single loop tuning problems. The design methods used to deal with these problems, and to limit the design to PI-controllers, are chosen to be *SIMC* and λ-*tuning*. In *SIMC*, an identification of the transfer functions of the system the controllers will be tuned for is required. This means that $g_{1,EOP}(s)$ and $g_{2,EOP}(s)$ must be known. The degree of freedom left to the designer is the specification of the closed loop time constant τ_c. In λ-*tuning* the controller design will be based on approximating step response tests of EOPs as first order plus time delay (FOPTD) and using this approximation together with specifying the design parameter λ.

FOPTD-approximations based on desired response time

How to carry out the FOPTD approximations requires investigation. According to the control engineers at Perstorp there are two distinctive cases; when fast control (small λ) is desired or when slow control (large λ) is desired. Their notion is that when tuning for fast control it is important that the FOPTD approximation best fits the parts of the step response caused by quick dynamics (low-valued time constants) since this is the dynamics the controller mainly will act on. A smaller λ gives a larger controller gain K_c (see Equation (2.7)) meaning a faster response to changes in the control error, $r - y$. Analogously for slow control the FOPTD approximation should best fit the slow dynamics of the step response including the steady state gain.

2.6 Evaluation of closed loop properties

The analysis of tuned PI-controllers is done through analysis of some measures after the closed loop system is subjected to changes in setpoint(s). The basis for the chosen measures is their pragmatic nature. They can be easily implemented for data obtained from actual process data or data from simulations and their interpretation is straightforward.

2.6.1 Assessing controller performance with T_r

For a closed loop system the rise time T_r is defined as the time it takes for the output to go from 10 % to 90 % of the final value after a unit step in the setpoint. It can be shown that if the closed loop transfer function is of first order plus time delay (FOPTD), as in Equation (2.5), the time constant λ is related to the rise time according to ([24])

$$\lambda = \frac{T_r}{ln(9)}. \tag{2.8}$$

2.6.2 Evaluating the presence of overshoot or oscillatory behaviour around the setpoint using IAE and IE

The integrated absolute error (IAE) and the integrated error (IE) are measures used to evaluate control performance [25]. They indicate how well the controller is able to attenuate perturbations of the system and the ability to achieve zero control error $e_i(t)$ in loop i (where $e_i(t) = r_i(t) - y_i(t)$). The measures are defined according to

$$IAE_i = \int_0^{+\infty} |e_i(t)| \,\mathrm{d}t, \tag{2.9}$$

$$IE_i = \int_0^{+\infty} e_i(t) \,\mathrm{d}t. \tag{2.10}$$

Smaller values of the integrals indicate better attenuation and thus better control performance. If the IE is smaller than the IAE then the control error must necessarily contain sign changes, which provides an indication of whether the control error overshoots and/or has an oscillatory behaviour around the setpoint $r_i(t)$. Thus, the ratio $\frac{IE}{IAE}$ can serve as a measure of this. If the ratio is equal to 1 no overshoot or oscillations of the output around the setpoint occur.

3 Creating an algorithm of the iterative method

Evaluations of the iterative method were based on algorithms in Matlab and step response tests of TITO systems simulated using Simulink. TITO systems considered were defined through transfer function matrices. Since one control loop should be closed in the iterative method and no example system with already present controllers were given it is also required in the algorithm to assign an initial tuning to both PI-controllers for the TITO control loops.

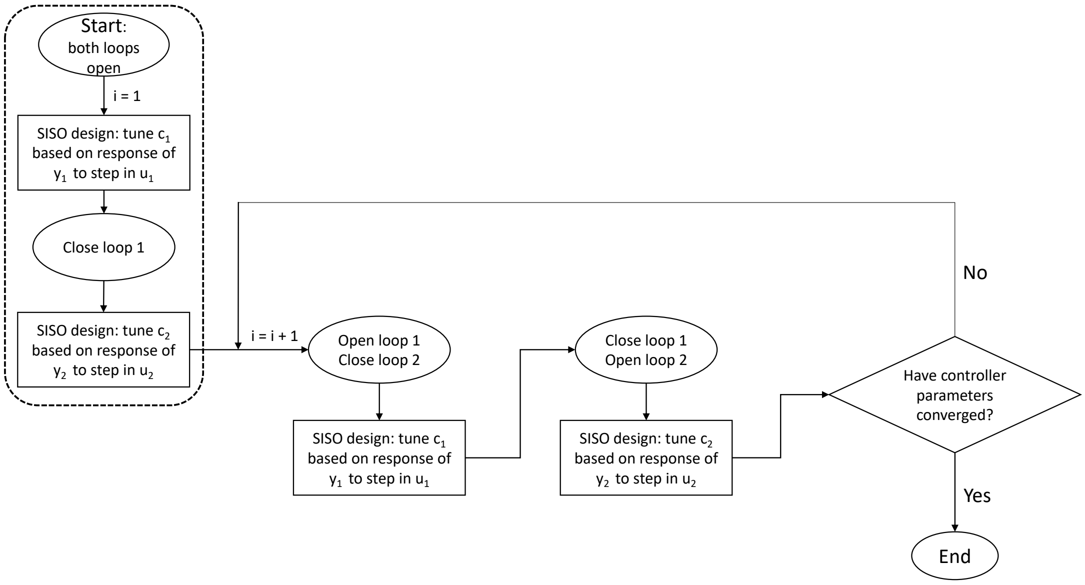


Figure 3.1: Flow-chart of the algorithm in the iterative method.

3.1 Outline of the iterative method

Using the notation of a general TITO transfer function matrix provided in Equation (2.2), Figure 3.1 represents an outline of the iterative method. Contained in the dashed region is what must be done when no controllers have been tuned yet, in the first rectangular block after "Start" a step response test with no controllers in closed loop is made, i.e. the system is completely in open loop. This initial assignment within the dashed region is along the lines of *sequential design* in decentralised control.

3.1.1 Definition of convergence in the iterative method

The algorithm ends when controller parameters converge. From a more pragmatic perspective it was decided that convergence of controller parameters in the iterative method was defined as when the parameters did not change more than ±5 % from the previous iteration. The actual tuning throughout the iterative method is based on some SISO design method (the rectangular blocks in Figure 3.1).

3.2 SIMC as the SISO design method in the iterative method

In order to use SIMC, the transfer functions of the SISO loops studied are required. In the case of TITO systems the transfer functions that had to be identified were the effective open-loop processes in Equation (2.3) and Equation (2.4), or, when no controller parameters are present, just the open loop transfer function element $g_{11}(s)$ from the transfer function matrix in Equation (2.2).

SIMC takes a transfer function and employs model reduction rules to reduce the transfer function to a first order model in order to generate a PI-controller. A Matlab algorithm was made of the model reduction. A PI-controller was then tuned based on the first order model using the SIMC tuning rules. Input to the algorithm was any SISO transfer function and a specification on the desired closed loop time constant τ_c.

3.3 λ-tuning as the SISO design method in the iterative method

The main difficulty in implementing an algorithm for tuning PI-controllers according to λ-tuning was the requirement of modelling the current loop being tuned as a first order system with time delay. With a first order model at hand, controller tuning according to λ-tuning is very straightforward (see Equation (2.7)). Data used for making the first order approximation is obtained from step response tests of effective open-loop processes. EOPs can contain many poles and zeros resulting in dynamics deviating from first order dynamics and it is not obvious what a good first order approximation should be. This, however, introduces an opportunity, there is a controller design freedom introduced by being able to make different approximations.

3.3.1 Approximating step response tests as first order dynamics with least-squares fitting

To best reflect what is done for real life systems, data on step response tests of control pair i (from u_i to y_i) with the other controller in closed loop, i.e. step response tests of EOPs were gathered. A first order plus time delay (FOPTD) model was then fitted to the measured output using least-squares fitting. This was done by minimizing the sum of squared errors between the step response of the control pair and the step response of the FOPTD approximation. The sum of squared errors

$$SSE = \sum_{i=1}^{N} (y_{true}(t_i) - y_{FOPTD}(t_i, K, T, \theta))^2 \tag{3.1}$$

was minimized with respect to the parameters K, T and θ, which is the steady state gain, time constant and time delay respectively, in the FOPTD model (see Equation (2.6)). Since the first order step response is not linear in its parameter a nonlinear least-squares fitting method is required. In Equation 3.1 $y_{true}(t_i)$ is the measured step response of the EOP being studied at discrete time instances t_i.

3.3.2 Method to let desired closed loop constant, λ, influence approximations

As was mentioned in Section 2.5.2 it is desired that the first order with time delay (FOPTD) approximation best fits the slow dynamics in the step response of the EOP when tuning a PI-controller for slow control and a good fit of the fast dynamics when tuning for fast control. In order for this to be reflected in the least-squares fitting a filter was introduced. The time domain signals y_{true} and y_{true} was filtered in frequency domain and the actual objective function that was minimized with respect to FOPTD model parameters were

$$SSE = \sum_{i=1}^{N}(y_{true,filtered}(t_i) - y_{FOPTD,filtered}(t_i, K, T, \theta))^2 \tag{3.2}$$

The purpose of the filter was to enable the designer influence over the approximations done by the least-squares fitting algorithm by weighting the frequency content of the errors in the sum of squared errors. High frequencies corresponds to fast dynamics, i.e. small time constants, in time domain and low frequencies to slow dynamics. The quickness in λ-tuned controlled systems should be directly related to the specification on λ, which represents the desired closed loop time constant. Large values mean slow control, small values fast control. Thus the larger the specified λ is, the FOPTD approximation should have increasingly better fit of the slow dynamics and vice versa. According to [19]: "[...] for control purposes it is most critical to have a good approximation of the plant behavior at about the bandwidth frequency"[1]. The desired closed loop transfer function in λ-tuning (see Equation (2.5)) has bandwidth frequency, $\omega_b = \frac{1}{\lambda}\ rad/s$.

Design of a filter in frequency domain

Combining the two ideas, bandwidth and control speed, gives a band-pass type filter in frequency domain, where frequencies close to $\frac{1}{\lambda}\ rad/s$ are passed through and other frequencies are attenuated, i.e. a filter which is a function of the tuning parameter λ. This means that a larger weight is put on minimizing the squared error in Equation (3.2) for frequencies close to $\frac{1}{\lambda}\ rad/s$, the desired bandwidth. This also means that there will be a better fit for low frequencies with a large λ, i.e. better fit for slow dynamics in the time domain. A small λ yields a better fit for higher frequencies, i.e. fast dynamics in the time domain.

[1]the frequency at which the gain of the closed loop transfer function is a factor $\frac{1}{\sqrt{2}}$ of the zero frequency (DC) gain

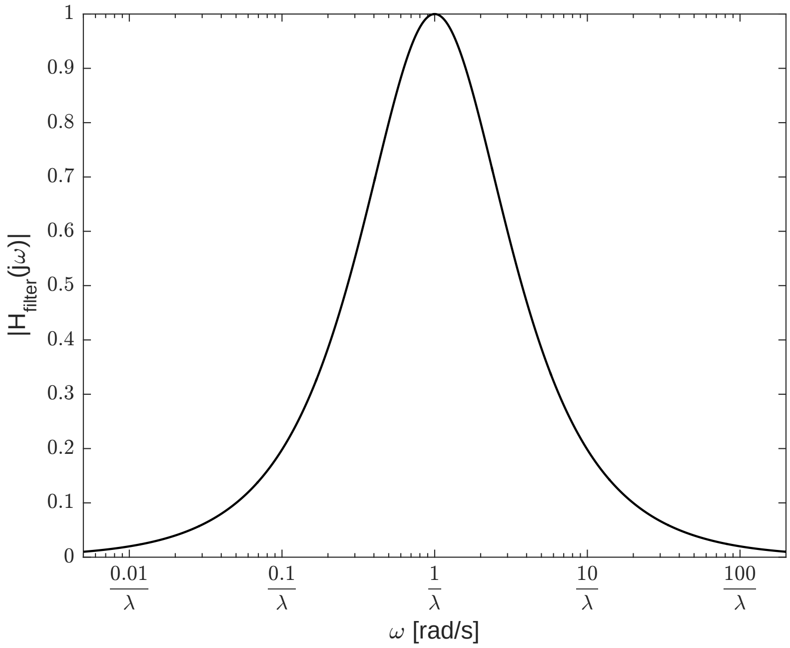


Figure 3.2: Amplitude response of the bandpass filter used in least-squares fitting algorithm.

A very simplistic (one zero and two poles) filter with bandpass characteristics is

$$H_{filter}(s) = \frac{2\lambda s}{(\lambda s + 1)^2} \tag{3.3}$$

The filter was used to filter the data used for FOPTD approximations. The magnitude as a function of frequency can be seen in Figure 3.2.

3.3.3 Implementation in Matlab

In short, the goal of the implementation in MATLAB was to generate the filtered sum of squared errors in Equation (3.2) through simulations in Simulink followed by minimizing the SSE with the nonlinear least-squares MATLAB function 'lsqnonlin' with respect to the parameters of the FOPTD approximation.

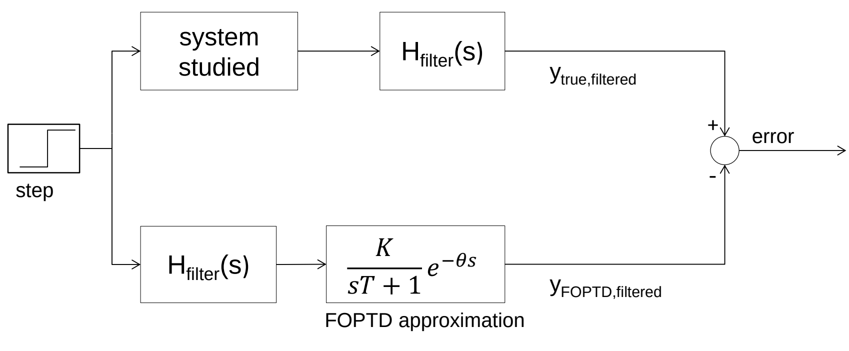


Figure 3.3: Representation of block diagram in Simulink in order to generate the error for the sum of squared errors in Equation 3.2

Simulink model and simulation time

Figure 3.3 is a portrayal of the Simulink model which calculated the errors used for least-squares fitting. A step input signal was fed to the current SISO control loop studied and the output was then filtered through the bandpass filter. Concurrently, the same step signal is fed through an exact copy of the filter and then through the FOPDT approximation and the difference (error) is formed between the two signals. Note that in the implementation the filter was put before the FOPTD approximation block. From a mathematical perspective this does not matter since SISO transfer functions are commutative, but it could matter from a simulation perspective. The simulation time used by the Simulink model is ten times the user specified tuning parameter, λ.

The Matlab function lsqnonlin used to minimize the sum of squared errors

The errors generated at different instances in time by the Simulink model was used by lsqnonlin to minimize the SSE with respect to the parameters K, T and θ. The algorithmic procedure in MATLAB is that *lsqnonlin* iteratively calls the Simulink model while varying the parameters until a stopping criterion is fulfilled. The desired criterion is that a minimum in the SSE has been found but other possible criteria are that either the change in the FOPTD parameters or the change of the SSE is small enough between two subsequent calls to the Simulink model. The user input to this procedure are the open loop transfer functions of the system studied (Equation (2.2)) and the specification on desired control speed through the tuning parameter λ. Since it is not realistic with a negative time delay and not desirable with an unstable FOPDT approximation a lower bound of 0 is specified for both T and θ in *lsqnonlin*.

Starting point for lsqnonlin

The MATLAB function *lsqnonlin* requires initial guesses of K, T and θ as a starting point for the algorithm. Within the iterative method the starting point during an iteration was taken as the resulting FOPTD approximation from the previous iteration. However for the very first iteration the initial guess is formed by gathering data from a simulation in Simulink of a step response of the SISO control loop studied. The initial guess $K_{initial}$ is then assigned the final value in the step response and $\theta_{initial}$ is set to the time instance when the step response has reached 5 % of the final value. The value of $T_{initial}$ is assigned the average of all time instances when the step response reaches 63 % of the final value from below (assuming the input step signal occur at t = 0) subtracted by the initial guess on θ.[2]

More options in the Matlab implementation

Both the Simulink model and *lsqnonlin* have more options and parameters to be specified, in addition to the ones already mentioned. In Table C.1 in Appendix C additional options and parameters for the MATLAB implementation that are different from the defualt values are listed.

[2]The step response of the general FOPTD model in Equation (2.6) is $y(t) = K(1-e^{\frac{-(t-\theta)}{T}}), \forall t \geq \theta$. Where $y(T+\theta) = K(1-e^{-1}) \approx 0.63K$. In other words at $t = T+\theta$ the step response has reached approximately 63% of the steady state gain. This shows why $T_{initial} = t_{63\%} - \theta_{initial}$ is chosen.

3.3.4 Evaluating whether approximations made by the Matlab implementation mimics a process control engineer

The degree of freedom when tuning according to the λ-method is the specification of the tuning parameter λ, a small value meaning fast control whereas a large value generally gives slow, but more robust control. When the control engineers at Perstorp want to achieve fast control they make sure their FOPTD approximation best fits the fast dynamics in the step response, i.e., it is important that the approximation has a better fit to the true response at times t in the beginning of the step response ($t_{step} \leq t << t_{steadystate}$). For slow control (large λ) they want a better fit of the slow dynamics i.e. best fit for times t close to when the step response has reached steady state ($t_{step} << t \leq t_{steadystate}$).

Table 3.1: Three examples of transfer functions deviating from FOPTD dynamics

$G_1(s)$	$G_2(s)$	$G_3(s)$
$\frac{5000s^2+100s+1}{(50s+1)(100s+1)}$	$\frac{(15s+1)}{(8s+1)(5s+1)}e^{-10s}$	$\frac{(22s+1)(16s+1)}{(45s+1)(8s+1)(2s+1)}$

Three transfer functions with complicated dynamics

The control engineers provided their FOPTD approximations of the step responses of the three transfer functions in Table 3.1 with two cases for each transfer function, one for slow control and one for fast control reflected in a specification of λ in each case. The approximations done by the LS-algorithm were then compared with these to see how close they resemble each other. Closed loop properties of controllers tuned according to λ-tuning were also investigated.

$G_1(s)$ in Table 3.1 is an example of an EOP of one of the TITO systems described in Section 3.4.1 below (the system with gas flow through a pipe with loop 1 (pressure control) closed). Note that the roots of the numerator polynomial is a complex conjugate pair. Due to the larger time constant of its zero than any of its poles the transfer function $G_2(s)$ has a step response with overshoot, its dynamics is also dominated by a large time delay relative to its time constants. The step response of $G_3(s)$ resembles a response of a real TITO system encountered by one of the control engineers at Perstorp.

3.4 Investigating the iterative method with two TITO systems with strong interactions between control loops

The algorithm of the iterative method was examined through two reference systems. One system prevalent at the different Perstorp plants and another system which is often used as a reference case when decentralised control is studied in the literature. The systems exemplify strong interactions between control loops by having large gains in off diagonal elements in their transfer functions matrices.

3.4.1 Gas flow through a pipe

A TITO system commonly encountered throughout the processes at Perstorp is gas flowing through a pipe where it is desired to control gas pressure and outgoing flow in a section of the pipe between two valves. Figure 3.4 shows a *piping and instrumentation diagram* of this system. It is assumed that in- and outgoing mass flowrate, q_{in} and q, in the pipe

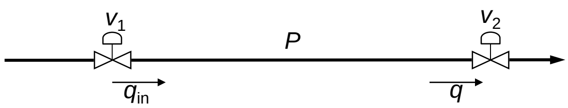


Figure 3.4: Diagram of outgoing, q and incoming, q_{in} gas mass-flow through a pipe between two control valves with valve openings v_1 and v_2. The pressure in the pipe is denoted by P.

is only dependent on the valve openings, v_1 and v_2, and the pressure drop over the valves according to:

$$q_{in} = v_1(P_{in} - P) \tag{3.4}$$

$$q = v_2(P - P_{out}) \tag{3.5}$$

The pressure variables P_{in} and P_{out} are upstream gas pressure (directly before the first valve), and downstream gas pressure (directly after the second valve), respectively. The gas pressure, P, is dependent on the total mass gas present at any instant in the pipe section between the two valves. Assuming that the gas is ideal and no change in temperature occur (isothermal) the rate of pressure change can be modeled as

$$\frac{\mathrm{d}}{\mathrm{d}t}P = C_p(q_{in} - q) \tag{3.6}$$

The constant C_p depends on temperature, the volume of the pipe section between the two control valves and the molar mass of the gas. It is desired to control both gas pressure, P, and outgoing flow, q. From Equations (3.4), (3.5) and (3.6) it can be seen that these two variables are co-dependent which results in TITO-dynamics.

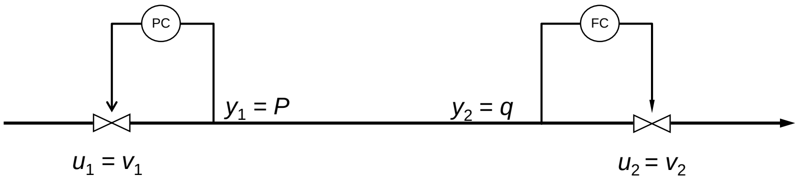


Figure 3.5: Diagram of gas flow through a pipe with two controllers controlling pressure and outgoing flow rate via two valves.

Two PI-controllers are added to the system, according to Figure 3.5, manipulating the positioning of the two valves. The output variable, y_1, is assigned to the pressure, P, which is controlled through the pressure controller PC by positioning of valve 1, v_1, which is thus assigned input signal u_1 i.e. loop 1. Output variable y_2 is the outgoing gas flow rate q trough valve 2 (the end of the pipe section). The flow controller FC (the loop 2 controller) manipulates the position of valve 2, v_2, which is the input signal u_2.

By assuming that all physical variables are unit-less and assigning

$$P_{in} = 1, \quad P_{out} = 1, \quad C_p = 0.01, \tag{3.7}$$

the system is linearized around the equilibrium point

$$P^0 = 0.5, \quad v_1^0 = 1, \quad v_2^0 = 1, \tag{3.8}$$

resulting in the transfer function matrix

$$\begin{bmatrix} y_1(s) \\ y_2(s) \end{bmatrix} = \begin{bmatrix} \frac{0.25}{50s+1} & \frac{-0.25}{50s+1} \\ \frac{0.25}{50s+1} & \frac{0.25(100s+1)}{50s+1} \end{bmatrix} \begin{bmatrix} u_1(s) \\ u_2(s) \end{bmatrix}. \tag{3.9}$$

Different cases investigated with λ-tuning as SISO design method

Table 3.2: Tuning parameters used in the iterative method with λ-tuning for the gas flow through a pipe system.

λ_1 [s] :	5	60	115	170	225	280	335	390	445	500
λ_2 [s] :	5	60	115	170	225	280	335	390	445	500

100 different tuning cases were studied by selecting different combinations of the tuning parameters λ_1 and λ_2 for the two control loops, as seen in Table 3.2. For each loop the tuning parameters was chosen as linearly spaced values between $\frac{1}{10}$ and 10 times the corresponding open loop time constant (the time constants for elements $(1,1)$ and $(2,2)$ in Equation 3.9.

3.4.2 Wood and Berry distillation column

A TITO system commonly encountered in articles on decentralised control is the Wood and Berry distillation column. Presented in [26] is a 2x2 transfer function matrix derived from system identification of a binary distillation system, separating methanol and water;

$$\begin{bmatrix} y_1(s) \\ y_2(s) \end{bmatrix} = \begin{bmatrix} \frac{12.8e^{-1s}}{16.7s+1} & \frac{-18.9e^{-3s}}{21.0s+1} \\ \frac{6.6e^{-7s}}{10.9s+1} & \frac{-19.4e^{-3s}}{14.4s+1} \end{bmatrix} \begin{bmatrix} u_1(s) \\ u_2(s) \end{bmatrix}. \tag{3.10}$$

This is an example of a system with strong interactions as apparent by the relatively large gains in the cross diagonal elements. Due to the presence of time delays the achievable controller performance is limited. Output y_1 is overhead composition of methanol and y_2 is the bottom composition of methanol. The inputs u_1 and u_2 are reflux flow rate from overhead condenser and steam flow rate to bottom reboiler, respectively.

Different cases investigated with λ-tuning as SISO design method

Table 3.3: Tuning parameters used in the iterative method with λ-tuning for the Wood and Berry distillation column.

λ_1 [s] :	1.67	2.65	4.19	6.65	10.54	16.70	26.47	41.95	66.48	105.37	167.00
λ_2 [s] :	1.44	2.28	3.62	5.73	9.09	14.40	22.82	36.17	57.33	90.86	144.00

The control pairs are u_1 with y_1 forming loop 1 and u_2 with y_2 forming loop 2. 121 different tuning cases were investigated with the iterative method with λ-tuning based on FOPTD approximations from the LS algorithm. The difference between the cases was the combination of different specifications on the tuning parameters λ_1 and λ_2 for each loop, as shown in Table 3.3. For loop 1 eleven logarithmically spaced (nearby values separated by multiplication of a common factor) values of λ_1 were chosen in relation to the open loop time constant for $g_{11}(s)$, i.e. 16.7 s. The smallest value is one tenth of the open loop time constant, the middle value equal to the open loop time constant and the largest value is ten times the open loop time constant. The different values for λ_2 were chosen in the same manner but instead related to the time constant of $g_{22}(s)$, i.e 14.4 s.

3.5 Evaluation of closed loop controller performance

The chosen measures to analyse closed loop properties were rise time, T_r, the integrated absolute error, IAE and the integrated error, IE. All measures were calculated through simulations in Simulink. Seperate simulations of a unit step in setpoint for each loop were made.

3.5.1 Calculating IAE and IE through simulations

The definitions of IAE and IE in Equations (2.9) and (2.10) will only result in finite values if the closed loop system is stable. This means that they can be used as measures for stability. Simulations cannot be run for infinite time and as such the upper limit of the integrals in the equations must be truncated to some finite time. This simulation time, T_{sim} was chosen sufficiently large such that the IAE value at $t = T_{sim}$ had not increased more than 0.001 % relative to the IAE value at $t = 0.9T_{sim}$. If this is satisfied the control error e has approximately converged to zero and the system is stable. If the system would be unstable the control error would increase exponentially and the integrated absolute error would blow up. This would directly be seen in the relative value stated above. In Simulink the variable step solver *ode45* was used to simulate IAE and IE values. Non defualt settings were the "Relative tolerance" of the solver set to $1e-12$ and the use of an adaptive zero-crossing algorithm with 'signal threshold' equal to $2000 \cdot eps$[3].

3.5.2 Calculating T_r through simulations

Calculations of T_r were made in seperate simulations to calculations of IAE and IE since a much higher resolution in simulation time steps is needed. Also, here the simulations were carried out in Simulink with the variable step solver *ode45*. It was important to obtain a high resolution in time to be able to extract the actual time points when the response in y_i is 10% and 90% of the final steady-state value. This was achieved in Simulink by setting the "Max step size" setting to one thousandth of the tuning parameter λ of the current loop being studied. Other non-defualt settings were the use of an adaptive zero-crossing algorithm with 'signal threshold' $2000 \cdot eps$ and the "Relative tolerance" set to $1e-12$.

[3] In Matlab and Simulink *eps* is the floating-point relative accuracy which is equal to $2.2204e-16$, i.e., $eps = 2.2204e-16$.

4 Simulations of the iterative method

4.1 Limitation of SIMC as SISO design method

When SIMC was used as single loop design method within the iterative method, the limitations in dealing with transfer functions containing poles and zeros with nonzero imaginary parts became evident. The SIMC tuning-rules approximate a given transfer function (such as an open loop transfer function or an EOP resulting from closing one of the control loops) with a first order system with time delay in order to generate a PI-controller. The rules used for model reduction can only deal with transfer functions having real-valued poles and zeros. Figure 4.1 shows the effect of the PI-controller parameters of loop 1 on the zeros of the resulting EOP from u_2 to y_2 for the system with gas flow through a pipe.

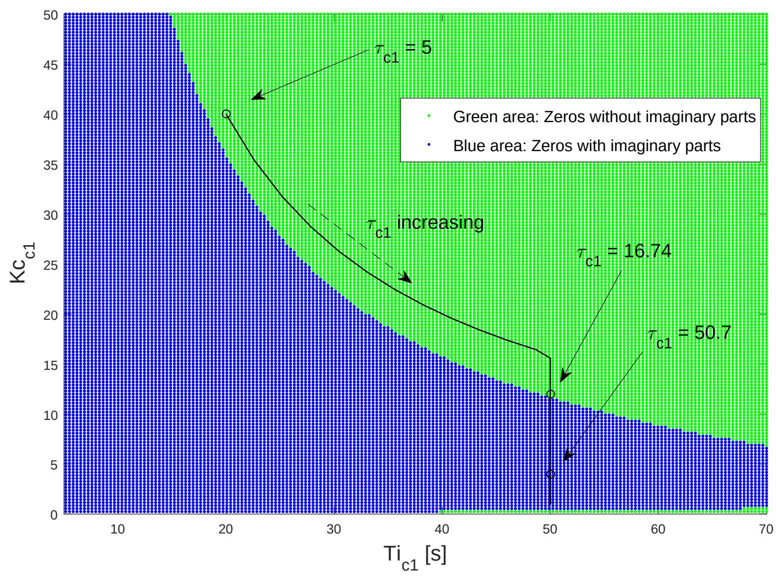


Figure 4.1: The case with gas flow through a pipe: The effect of the controller for loop 1 on the zeros of the effective open-loop process of u_2 to y_2, with loop 1 closed.

4.1.1 The iterative method with the gas flow through a pipe system

During the first iteration in the iterative method the tuning of the PI-controller for loop 1 is based on the open loop dynamic relationship between u_1 and y_1 with loop 2 open. In SIMC this means that the SISO transfer function to be used as input to the algorithm, when tuning controllers for the gas flow through a pipe system, is element $(1, 1)$ in Equation (3.9) which is already of first order and requires no FOPTD approximation. When SIMC is applied to this transfer function the black line in Figure 4.1 gives the PI-controller parameters as a function of the tuning parameter τ_{C1}.

4.1.2 The effect of controller parameters on the zeros for the EOP of loop 1

Next, the PI-controller for loop 2 is tuned with loop 1 closed and loop 2 open. Controller 1 (c_1) will necessarily affect the EOP of loop 2 as shown by Equation (2.4). This EOP will serve as input to SIMC and the coloured areas in Figure 4.1 show how the zeros of this EOP is affected by the controller parameters of c_1. As can be seen, when c_1 has been tuned for fast control (small τ_{C1}) the EOP has real valued zeros whereas slow control (large τ_{C1}) leads to zeros with imaginary parts. The boundary between the two regions happens when τ_{C1} is equal to $16.74s$. It should be mentioned that this is about $\frac{1}{3}$ of the time constant for the first order transfer function the controller is tuned for, i.e. $50s$.

4.1.3 Any realistic tuning case results in an EOP containing zeros with imaginary parts

τ_{C1} can be regarded as the desired time constant of the closed loop system in its response to a setpoint change (c.f. λ in λ-tuning). In the context of process control it is almost always desired that τ_{C1} is larger than the time constant of the system in order to achieve enough robustness. Thus, the smallest value chosen of the design parameter τ_{C1} would be $50s$ for the gas flow through a pipe-system. The conclusion is then that for all realistic choices of the tuning parameter we would always generate a EOP between u_2 and y_2 with zeros having imaginary parts. However, SIMC cannot deal with neither poles nor zeros that are not real. This in turn means that the SIMC-rules cannot be used in the iterative tuning whenever a transfer function having complex valued poles or zeros is encountered.

4.2 λ-tuning based on first-order-dynamics approximations of step response tests

The other utilised SISO-design method to generate PI-controllers in the iterative method is λ-tuning. A prerequisite to use λ-tuning is to model the process as first order plus time delay (FOPTD). The basis for the iterative method with λ-tuning for TITO systems is to perform a step response test with one control loop closed and one loop open, which results in step responses of EOPs. As evident from Equations 2.3 and 2.4 a TITO system with quite simple open loop dynamics can lead to EOPs with many poles and zeros being far from the simple dynamics of a FOPTD process. Thus it is of interest to study how the proposed least-squares (LS) algorithm, implemented in MATLAB, approximates transfer functions that deviate from FOPTD-dynamics and the resulting closed loop behaviour with λ-tuned controllers. Three example transfer functions $G_1(s)$, $G_2(s)$ and $G_3(s)$ according to Table 3.1 in Section 3.3.4 were studied.

4.2.1 Comparing approximations by the control engineers with the least-squares fitting algorithm

One of the main points of the developed LS algorithm is to capture the essence in how a control engineer at Perstorp approximates results from a step response as a FOPTD model. Approximations of the the three example transfer functions, according to the

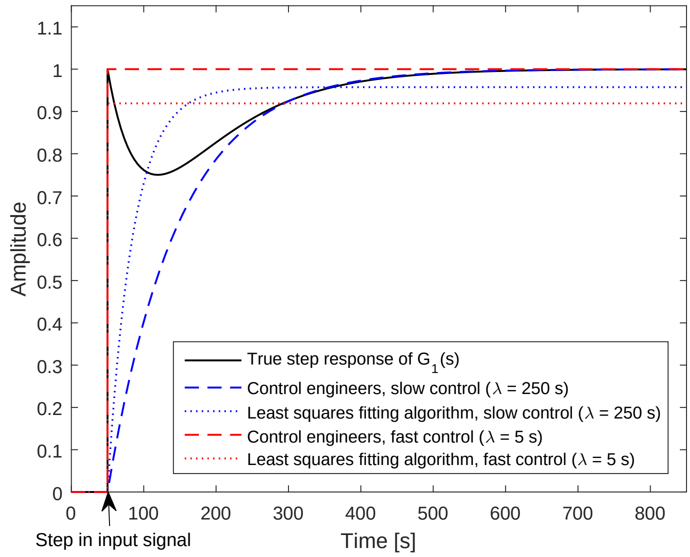


Figure 4.2: Unit step response of $G_1(s)$ and four FOPTD approximations. Step in input signal at time = 50 s.

control engineers, depended on whether "slow" or "fast" control was desired (as specified by λ). Comparisons of the step responses of these approximations with approximations done by the LS algorithm of $G_1(s)$, $G_2(s)$ and $G_3(s)$ can be found in Figure 4.2,Figure 4.3 and Figure 4.4. The true step response of the corresponding transfer function is also provided. For numerical values of the different FOPTD approximations consult Table B.1 in Appendix B.

FOPTD approximations of a lead-lag process with complex zeros

As can be seen in Figure 4.2 the step response of $G_1(s)$ instantly reaches the same value as the steady state gain (due to equal number of zeros as poles, i.e. not a strictly proper transfer function) then goes through a dip before it converges to a steady state. When tuning for fast control ($\lambda = 5$ s) the control engineer approximates the process as just a constant gain (with both time constant T and time delay θ set to zero). The LS algorithm results in a nonzero approximation of T. However, it is so small that from a practical viewpoint it can be regarded as zero. The main difference between the LS algorithm and control engineer is the approximation of the gain. For slow control ($\lambda = 250$ s) the time delay is again approximated as zero in both cases. The LS algorithm provides a smaller (faster) time constant and a smaller gain than the approximation done by the control engineer.

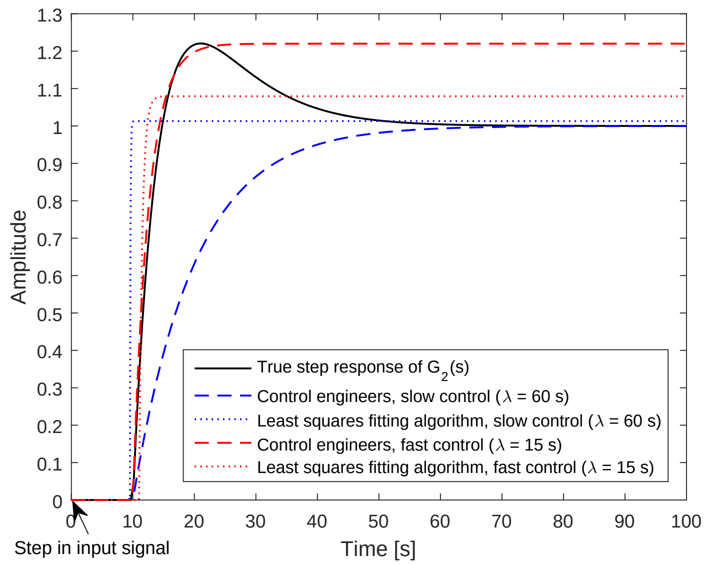


Figure 4.3: Unit step response of $G_2(s)$ and four FOPTD approximations. Step in input signal at time = 0 s.

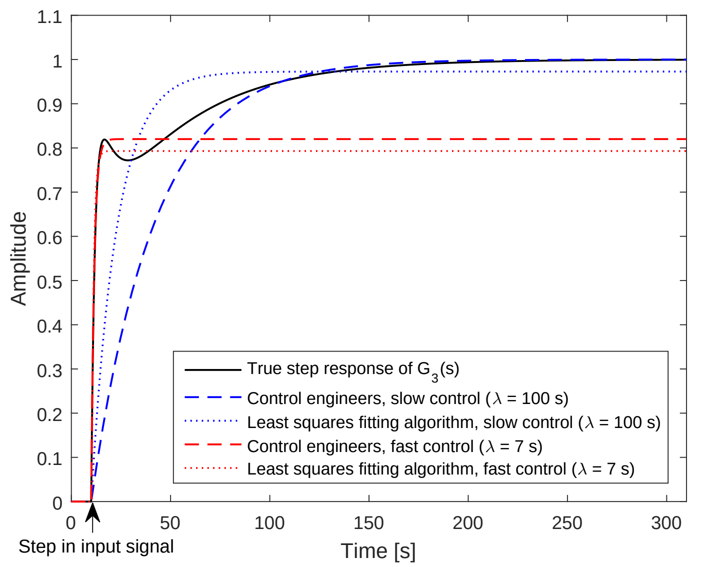


Figure 4.4: Unit step response of $G_3(s)$ and four FOPTD approximations. Step in input signal at time = 10 s.

FOPTD approximations of a lead dominant transfer function with significant time delay

The step response of $G_2(s)$ can be seen in Figure 4.3 clearly showing an overshoot. The control engineers choose to have a good fit of their approximation during the initial 15s when tuning for fast control ($\lambda = 15$ s). For slow control ($\lambda = 60$ s) they want a good fit of the steady state gain and they choose a time constant T that results in an approximation that settles at steady state at about the same time as the true step response. The appproximations from the LS algorithm differ quite a bit from the control engineers. For fast control the algorithm yields both a smaller gain K and a faster time constant T. In the case of slow control the least-squares algorithm pretty much results in the same gain as the approximation by the control engineers. However the approximated time constant is very small (smaller than in the case for fast control) and is two orders of magnitude smaller than the control engineers' approximation.

FOPTD approximations of a high order transfer function with large differences in time constants

Figure 4.4 shows that the step response of $G_3(s)$ is first characterised by a very quick response followed by a slow settling towards steady state. The control engineers choose an FOPDT-approximation for fast control ($\lambda = 7$s) that best follows the initial quick response whereas for slow control ($\lambda = 100$) a good fit of the step response for times after 100 s is desired. The approximation from the LS algorithm for fast control is quite close the control engineers'. In the case of slow control the LS algorithm results in a time constant about $\frac{1}{3}$ of the time constant in the FOPTD approximation by the control engineers.

4.2.2 Measuring closed loop performance of λ-tuned controllers using T_r and $\frac{IE}{IAE}$

Based on the FOPTD-approximations and the specified closed loop time constant λ, PI-controllers were tuned with λ-tuning for the three different transfer functions $G_1(s)$, $G_2(s)$ and $G_3(s)$. Simulating a unit step change in the setpoint, $r(t)$, some closed loop properties were evaluated and listed in Table 4.1. The objective in λ-tuning is to achieve a closed loop system with FOPTD dynamics (from setpoint to output) with a time constant = λ. As shown in Equation (2.8) the rise time T_r divided by $ln(9)$ is equal to the time constant for a first order system. IAE is a measure of performance, The ratio between IE and IAE shows if the sign of the control error changes during the setpoint change, giving an indication of oscillatory behaviour or overshoot.

4.2.3 Better controller performance when using approximations from least-squares fitting than approximations from control engineers

In comparing closed loop properties between the least-squares algorithm with the control engineers at Perstorp the results in Table 4.1 show quite small differences. In all cases the LS algorithm yields values of $\frac{T_r}{ln(9)}$ closer to the specified λ. All IAE values are finite meaning that all controllers result in stable closed loop systems. It is only in the case of fast control that $\frac{IE}{IAE}$ is less than 1 with the smallest value being 0.600 indicating some oscillatory behaviour or overshoot of the output y around the setpoint r. For a graphical representation of closed loop response from a unit step in setpoint c.f. Figure B.1,Figure B.2 and Figure B.3 in Appendix B.

Table 4.1: Closed loop properties of three example transfer functions with PI-controllers tuned according to the λ-method

	Slow control							Fast control						
		Control engineers			LS algorithm				Control engineers			LS algorithm		
	λ	$\frac{T_r}{ln(9)}$	IAE	$\frac{IE}{IAE}$	$\frac{T_r}{ln(9)}$	IAE	$\frac{IE}{IAE}$	λ	$\frac{T_r}{ln(9)}$	IAE	$\frac{IE}{IAE}$	$\frac{T_r}{ln(9)}$	IAE	$\frac{IE}{IAE}$
$G_1(s)$	250 s	289 s	250	1	249 s	239	1	5 s	5.69 s	8.33	0.600	5.46 s	7.67	0.600
$G_2(s)$	60 s	69.1 s	70.0	1	60.5 s	70.4	1	15 s	22.4 s	30.5	1	11.3 s	28.1	1
$G_3(s)$	100 s	113 s	100	1	99.8 s	97.3	1	7 s	7.73 s	8.68	0.661	7.36 s	8.81	0.657

4.3 The iterative method with λ-tuning as a SISO design method

The iterative method is investigated for the two TITO reference systems; The gas flow through a pipe system and the Wood and Berry distillation column. In order to design PI-controllers, the iterative method utilising λ-tuning based on FOPTD approximations from the proposed least-squares algorithm has been employed. Many different tuning cases have been investigated for both systems. The cases are separated by differences in desired control speed through different specifications of the tuning parameters λ_1 and λ_2 for each control loop in the TITO systems according to Tables 3.2 and 3.3 in Section 3.4.

4.3.1 Convergence in PI-controller parameters

For both TITO systems, the iterative method converged in all tuning cases studied. As a gentle reminder convergence here means that the resulting controller parameters in the last iteration is within ±5% of the previous iteration. Also as a reminder, the tuning of the first loop inte the first iteration is based on a step response test with both loops open in the TITO system (what is done during the first iteration is contained in the dashed region in Figure 3.1). The histograms in Figures 4.5 and 4.6 shows that in most cases, for both reference systems, convergence of the iterative method was reached within four iterations. The final PI-controller parameters, after the iterative method converged for the different cases, can be found in Tables B.2 and B.3 in Appendix B.

4.3.2 Converged controller parameters resulted in closed loop stability in all cases

It was investigated whether all the converged controller parameters yielded a stable closed loop TITO system. For the gas flow through a pipe system stability was achieved in all cases. This stability analysis was conducted by forming the closed loop transfer function matrix and calculating the poles through the *pole* function in MATLAB. The Wood and Berry column was also closed loop stable in all its investigated tuning cases. Since there are timedelays present in the system, stability was not checked by calculating poles. Instead, closed loop stability was found through simulations in Simulink. By using a large enough simulation time it was shown that the IAE-values in both loops after a perturbation of the system settled at some finite value which shows closed loop stability.

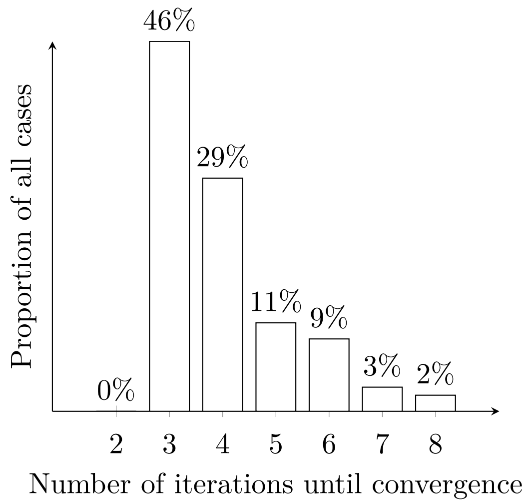


Figure 4.5: Distribution of the required number of iterations to achieve convergence in PI-controller parameters for the gas flow through a pipe system in the iterative method.

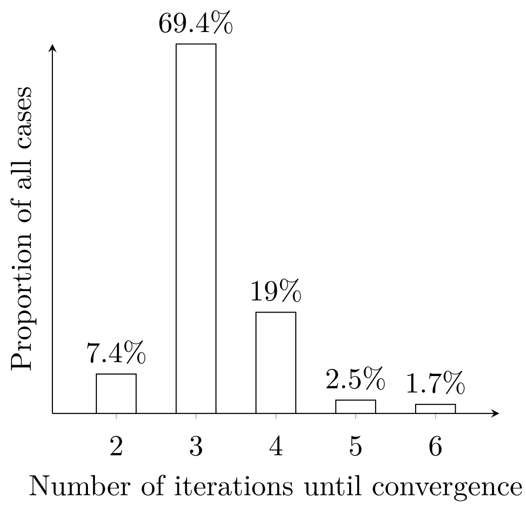


Figure 4.6: Distribution of the required number of iterations to achieve convergence in PI-controller parameters for the Wood and Berry system in the iterative method.

4.3.3 Simulations of closed loop properties of the system with gas flow through a pipe

PI-controllers for the system were tuned with the iterative method. As a SISO design method λ-tuning was used, based on FOPTD approximations obtained from the least-squares algorithm. The tuning started with loop 1 with the whole TITO system in open loop. The two control loops were then sequentially opened and closed during the iterations. The investigated tuning parameters λ_1 and λ_2 can be found in Table 3.2 in Section 3.4.1.

Table 4.2: Ratio between $\frac{T_r}{ln(9)}$ and λ_1 for loop 1 after a unit step in r_1 for the gas flow through a pipe system in closed loop with converged PI-controller parameters for different cases.

$\frac{T_r}{\lambda_1 ln(9)}$		λ_2 [s]									
		5	60	115	170	225	280	335	390	445	500
λ_1 [s]	5	1.03	1.02	1.01	1.01	1.02	1.00	1.00	1.00	1.00	1.00
	60	1.02	1.15	1.06	1.18	1.24	1.21	1.21	1.18	1.21	1.16
	115	0.91	0.86	0.92	0.96	1.23	1.25	1.20	1.21	1.12	1.21
	170	0.96	0.84	0.88	1.13	1.09	1.15	1.18	1.19	1.09	1.14
	225	1.00	0.95	0.82	0.84	0.92	1.19	1.06	1.15	0.98	1.22
	280	0.97	0.83	0.90	1.08	0.88	1.19	0.91	0.90	0.92	0.91
	335	1.02	0.89	0.82	1.07	0.82	0.86	0.87	0.91	0.86	0.89
	390	1.00	0.85	0.86	0.86	0.83	0.89	1.13	0.87	0.87	0.88
	445	0.89	0.85	0.80	0.86	0.83	0.83	0.84	0.86	0.88	0.96
	500	1.00	0.95	0.81	0.80	0.80	0.82	0.88	0.86	0.91	1.03

The closed loop rise time, T_r, is close to the objective in λ-tuning

Closed loop properties of all controllers were evaluated and as a first test a simulation of the response to a unit step in setpoint for loop 1, r_1, was conducted. The rise time T_r was calculated for y_1 and compared to λ_1 in the given case to see whether Equation (2.8) is fulfilled. In Table 4.2 the ratio between $\frac{T_r}{ln(9)}$ and λ_1 can be seen for all different cases of the iterative method. A value of 1 means that Equation (2.8) is satisfied. For $\lambda_1 = 5s$ this is pretty much true in almost all cases. The largest relative difference is 0.25 for $\lambda_1 = 115s$ and $\lambda_2 = 280s$. A graphical representation of the response to a unit step in r_1, for this case, can be seen in the first part of the plot in Figure 4.7.

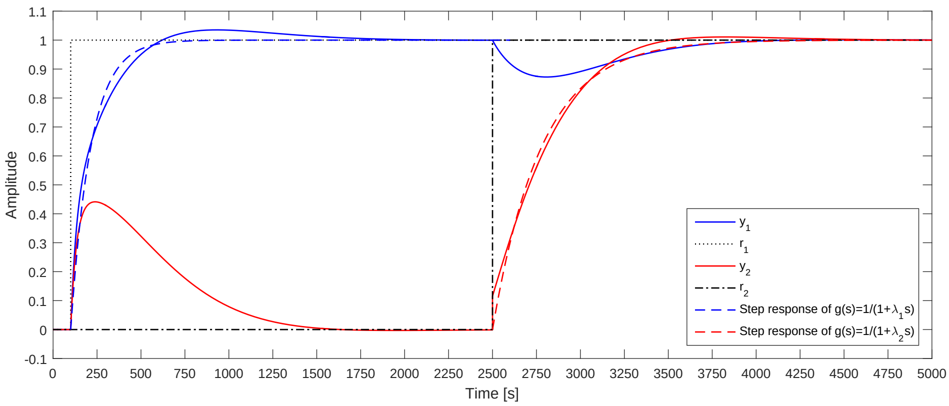


Figure 4.7: Closed loop response to changes in setpoints for the gas flow through a pipe system with controllers from the case with $\lambda_1 = 115s$ and $\lambda_2 = 280s$.

Figure 4.7 demonstrates a case where the designer desires tighter control of loop 1 than loop 2 (λ_1 smaller than λ_2). We have some overshoot for both y_1 and y_2 in response to their corresponding setpoint changes otherwise they do not look to differ too much from a first order response (as is the objective of λ-tuning). The setpoint change in r_1 give rise to a maximum control error in loop 2 more than triple the maximum control error in loop 1 after the setpoint change in r_2. The step responses of the desired closed loop responses for both loops have been plotted in the figure for reference.

Table 4.3: Ratio between $\frac{T_r}{ln(9)}$ and λ_2 for loop 2 after a unit step in r_2 for the gas flow through a pipe system in closed loop with converged PI-controller parameters for different tuning cases.

$\frac{T_r}{\lambda_2 ln(9)}$		λ_2 [s]									
		5	60	115	170	225	280	335	390	445	500
λ_1 [s]	5	1.02	0.99	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	60	1.06	1.03	0.99	0.97	0.98	0.94	0.96	0.99	0.99	0.97
	115	1.07	1.13	1.00	0.97	0.97	1.01	0.94	1.01	0.96	0.96
	170	1.07	1.20	1.03	1.06	1.07	1.03	1.02	1.02	0.92	0.98
	225	1.07	1.22	1.07	1.03	1.05	1.02	1.04	1.03	0.91	1.00
	280	1.07	1.27	1.08	1.04	1.05	1.01	1.01	1.05	0.93	0.91
	335	1.07	1.27	1.11	1.04	1.00	1.05	1.02	0.94	0.93	0.90
	390	1.07	1.31	1.13	1.05	1.19	1.08	1.10	1.02	1.00	0.99
	445	1.07	1.33	1.15	1.18	1.12	1.11	1.08	1.05	1.04	1.02
	500	1.07	1.31	1.16	1.20	1.13	1.12	1.11	1.08	1.06	1.06

Table 4.3 shows the ratio between $\frac{T_r}{ln(9)}$ and λ_2 for loop 2 after a unit step in r_2 for the closed loop TITO system. Most values range between 0.9 and 1.1 indicating that in the corresponding cases $\frac{T_r}{ln(9)}$ does not deviate more than 10% from the specified λ_2. For slow control of loop 1 and faster control of loop 2 (lower left part of Table 4.3) we have the largest deviations. The largest relative difference is given by the case with $\lambda_1 = 445s$ and $\lambda_2 = 60s$. The response of the closed loop system to a unit step in r_2, for this case, can be found in the latter half of the plot in Figure 4.8. This is a case where a tuner wants more than seven times faster control of loop 2 than loop 1. In comparison with the case depicted in Figure 4.7 y_2 is now much less affected by the change in r_1 whereas on the

other hand y_1 is much more affected by the change in r_2.

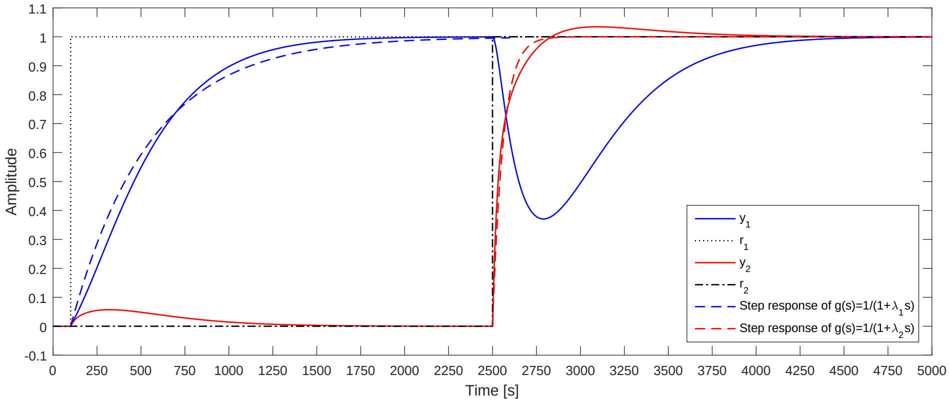


Figure 4.8: Closed loop response to changes in setpoints for the gas flow through a pipe system with controllers from the case with $\lambda_1 = 445s$ and $\lambda_2 = 60s$.

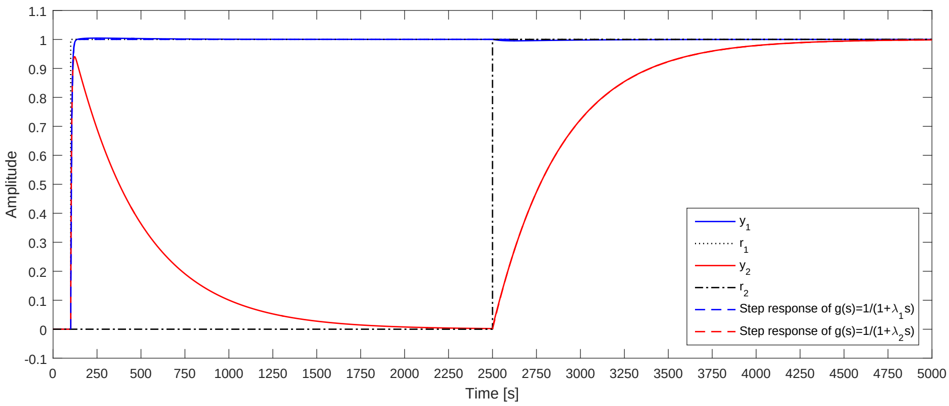


Figure 4.9: Closed loop response to changes in setpoints for the gas flow through a pipe system with controllers from the case with $\lambda_1 = 5s$ and $\lambda_2 = 390s$.

The situation where $\frac{T_r}{ln(9)}$ best follows the corresponding λ for both loops is found when $\lambda_1 = 5s$ and $\lambda_2 = 390s$ i.e. a very large difference in desired control speed. The closed loop system with unit step changes in r_1 and r_2 can be found in Figure 4.9. It is clear that the change in r_1 has a large effect on y_2 with a slow attenuation of the resulting control error. On the other hand, y_1 is pretty much unaffected by the unit step in r_2. For reference, plots of the step responses of the desired closed loop responses have been added to the figure.

Table 4.4: Ratio between IE and IAE for loop 1 after a unit step in r_1 for the gas flow through a pipe system in closed loop with converged PI-controller parameters for different cases.

	$\frac{IE_1}{IAE_1}$	λ_2 [s]									
		5	60	115	170	225	280	335	390	445	500
	5	0.88	0.41	0.38	0.36	0.35	0.35	0.35	0.35	0.34	0.34
	60	1.00	0.91	0.54	0.59	0.56	0.48	0.46	0.45	0.45	0.43
	115	0.95	0.71	0.63	0.55	0.70	0.64	0.58	0.56	0.48	0.49
	170	1.00	0.84	0.72	0.84	0.71	0.69	0.64	0.63	0.52	0.55
λ_1 [s]	225	1.00	0.98	0.78	0.69	0.68	0.83	0.66	0.69	0.54	0.65
	280	1.00	0.96	0.93	0.96	0.72	0.86	0.63	0.58	0.55	0.55
	335	1.00	1.00	0.93	0.98	0.76	0.70	0.66	0.63	0.57	0.57
	390	1.00	1.00	0.97	0.92	0.79	0.79	0.86	0.67	0.64	0.62
	445	1.00	1.00	0.98	0.94	0.85	0.79	0.74	0.71	0.68	0.68
	500	1.00	1.00	0.99	0.94	0.88	0.82	0.80	0.75	0.73	0.78

Table 4.5: Ratio between IE and IAE for loop 2 after a unit step in r_2 for the gas flow through a pipe system in closed loop with converged PI-controller parameters for different cases.

	$\frac{IE_2}{IAE_2}$	λ_2 [s]									
		5	60	115	170	225	280	335	390	445	500
	5	0.93	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	60	0.65	0.76	0.88	0.95	0.99	1.00	1.00	1.00	1.00	1.00
	115	0.54	0.60	0.71	0.82	0.88	0.94	0.97	0.99	1.00	1.00
	170	0.51	0.54	0.60	0.74	0.83	0.88	0.92	0.95	0.97	0.99
λ_1 [s]	225	0.49	0.52	0.55	0.63	0.75	0.79	0.87	0.90	0.92	0.93
	280	0.45	0.48	0.54	0.63	0.68	0.71	0.80	0.87	0.86	0.90
	335	0.46	0.48	0.51	0.59	0.60	0.69	0.74	0.76	0.81	0.84
	390	0.44	0.46	0.49	0.54	0.64	0.68	0.72	0.76	0.80	0.83
	445	0.42	0.45	0.47	0.55	0.58	0.64	0.69	0.73	0.79	0.79
	500	0.43	0.45	0.46	0.53	0.57	0.62	0.68	0.72	0.74	0.77

Slow control in a loop means less overshoot or oscillations if the other loop is tuned for fast control

Overshoot and changes between different signs of the control error in response to a unit step in r_1 was investigated for loop 1. Results showing the ratio between IE and IAE are shown in Table 4.4. There is a clear trend showing that slow control of loop 1 and fast control of loop 2 lead to an increase in value of the ratios. Table 4.5 show the ratios between IE and IAE for loop 2 after a unit step in r_1 instead of r_2. The tendency is that fast control of loop 1 and slow control of loop 2 gives less overshoot or less oscillations in the control error.

4.3.4 Simulations of closed loop properties of the Wood and Berry distillation column

The resulting controller parameters for two PI-controllers from the iterative method using λ-tuning were for all different tuning cases (see Table 3.3 in Section 3.4.2) simulated in closed loop with the dynamical system. The closed loop properties T_r, IE and IAE were calculated from simulated step responses.

Table 4.6: Ratio between $\frac{T_r}{ln(9)}$ and λ_1 for loop 1 after a unit step in r_1 for the Wood and Berry column in closed loop with converged PI-controller parameters for different cases.

	$\frac{T_r}{\lambda_1 ln(9)}$	λ_2 [s]										
		1.44	2.28	3.62	5.73	9.09	14.40	22.82	36.17	57.33	90.86	144.00
λ_1 [s]	1.67	1.11	1.05	1.03	1.00	0.98	0.95	0.93	0.91	0.89	0.89	0.89
	2.65	0.88	1.11	1.11	1.15	1.14	1.06	0.96	0.92	0.94	0.94	0.94
	4.19	0.82	0.73	0.86	0.78	0.90	0.88	0.92	1.05	1.05	1.07	1.08
	6.65	0.60	0.58	0.55	0.57	0.57	0.74	0.77	0.91	1.00	1.00	1.04
	10.54	1.65	1.44	1.67	1.87	0.80	0.77	0.78	0.83	0.86	0.90	0.94
	16.70	1.43	1.34	1.31	1.45	1.42	1.60	1.48	0.64	0.59	1.20	0.97
	26.47	1.28	1.27	1.30	1.31	1.33	1.44	1.55	1.70	1.57	0.72	1.52
	41.95	1.22	1.21	1.16	1.27	1.20	1.32	1.33	1.46	1.63	1.75	1.47
	66.48	1.15	1.15	1.16	1.19	1.19	1.19	1.26	1.33	1.29	1.47	1.56
	105.37	1.09	1.11	1.11	1.12	1.13	1.14	1.15	1.24	1.32	1.39	1.28
	167.00	1.09	1.07	1.08	1.10	1.09	1.12	1.14	1.19	1.23	1.28	1.37

The closed loop rise time, T_r, is reasonably close to the objective in λ-tuning

Comparisons between the rise time and the tuning parameter λ_1 for loop 1 in response to a unit step in r_1 can be seen in Table 4.6. Values larger than 1 indicate a slower closed loop response than desired and values smaller than 1 indicate a faster response. There appears to be no clear trends, though there is a slight indication that slow control for loop 1 (larger λ_1) results in ratios between $\frac{T_r}{ln(9)}$ and λ_1 being larger than 1. The largest relative difference is found in the case when $\lambda_1 = 10.54s$ and $\lambda_2 = 5.73s$ with the ratio equal to 1.87. The response of the closed loop system to sequential steps in r_1 and r_2 for this case can be found in Figure 4.10.

The case in Figure 4.10 have specified closed loop time constants, λ_1 and λ_2, smaller than the corresponding ones for the open loop elements $g_{11}(s)$ and $g_{22}(s)$ (see Equation (3.10)). There is some oscillatory behaviour but not necessarily accompanied by overshoot, as can be seen by the response in y_1 to the change in r_1. The largest magnitude in the control error of both loops when the setpoint is changed for the *other* loop is roughly the same. For comparison, the desired closed loop responses for both loops are also shown in the figure.

In response to a unit step in r_2, Table 4.7 indicates that for the smallest value in λ_2, the ratio between $\frac{T_r}{ln(9)}$ and λ_2 is always larger than 1, no matter the value of λ_1. For $\lambda_2 = 3.62$s, 5.73s and 9.09s the ratio is below 1 apart from two instances. For $\lambda_2 = 36.17$s and larger, all values in the table are larger than 1. The case with the largest relative difference in its ratio with the value 1.61 is when $\lambda_1 = 105.37s$ and $\lambda_2 = 36.17s$ and a visualisation of the closed loop dynamics can be seen in Figure 4.11.

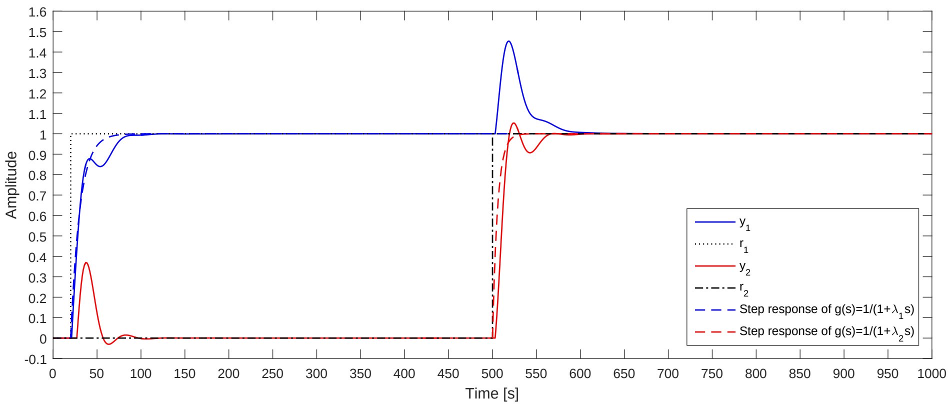


Figure 4.10: Closed loop response to changes in setpoints for the Wood and Berry column with controllers from the case with $\lambda_1 = 10.54s$ and $\lambda_2 = 5.73s$.

Table 4.7: Ratio between $\frac{T_r}{ln(9)}$ and λ_2 for loop 2 after a unit step in r_2 for the Wood and Berry column in closed loop with converged PI-controller parameters for different cases.

	$\frac{T_r}{\lambda_2 ln(9)}$	λ_2 [s]										
		1.44	2.28	3.62	5.73	9.09	14.40	22.82	36.17	57.33	90.86	144.00
λ_1 [s]	1.67	1.30	1.14	0.83	0.77	1.11	1.01	0.98	1.00	1.00	1.01	1.02
	2.65	1.28	1.17	0.85	0.79	1.35	1.11	1.01	1.01	1.01	1.02	1.03
	4.19	1.24	1.15	0.94	0.86	0.77	1.06	1.03	1.04	1.03	1.03	1.03
	6.65	1.26	1.15	0.94	0.88	0.96	1.02	1.07	1.06	1.05	1.06	1.04
	10.54	1.25	1.13	0.99	0.92	0.71	0.78	1.30	1.14	1.08	1.06	1.06
	16.70	1.22	1.09	0.86	0.89	0.79	0.87	1.02	1.19	1.08	1.11	1.08
	26.47	1.20	1.06	0.95	0.82	0.87	0.81	0.80	1.33	1.13	1.16	1.10
	41.95	1.19	1.03	0.79	0.83	0.81	0.78	0.73	1.39	1.33	1.16	1.12
	66.48	1.18	1.00	0.95	0.95	0.90	0.77	0.81	1.61	1.42	1.28	1.18
	105.37	1.18	0.98	0.92	0.93	0.88	0.85	0.87	1.61	1.42	1.44	1.26
	167.00	1.18	0.97	0.90	0.94	0.92	0.87	0.84	1.08	1.49	1.54	1.29

The closed loop response in Figure 4.11 is for λ_1 more than 6 times slower than the open loop time constant in $g_{11}(s)$ of the Wood and Berry system. The desired closed loop response time λ_2 is about 2.5 times slower than the time constant of $g_{22}(s)$. There is no overshoot for the response of each loop to their corresponding setpoint changes. The output y_1 seems to be more affected by the step in r_2 than y_2 is affected by the step in r_1. The step responses of the desired closed loop responses for both loops have been plotted in the figure for reference.

Figure 4.12 depicts the situation where $\frac{T_r}{ln(9)}$ best follows the corresponding λ for both loops. This case is when $\lambda_1 = 2.65s$ and $\lambda_2 = 22.82s$ i.e. fast control of loop 1 in comparison to the open loop time constant in $g_{11}(s)$, and slow control control of loop 2 in comparison to the open loop time constant in $g_{22}(s)$. The output y_1 has a very small overshoot after the step in r_1 and y_2 shows a smooth response with no overshoot to the step in r_1. The maximum control error for loop 1 after a setpoint change in the *other* loop is about five times larger than for loop 2. For reference, plots of the step responses of the desired closed loop responses have been added to the figure.

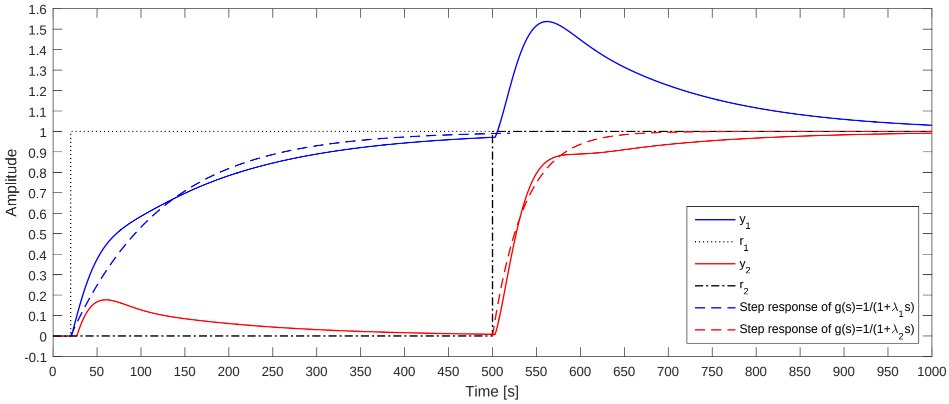


Figure 4.11: Closed loop response to changes in setpoints for the Wood and Berry column with controllers from the case with $\lambda_1 = 105.37s$ and $\lambda_2 = 36.17s$.

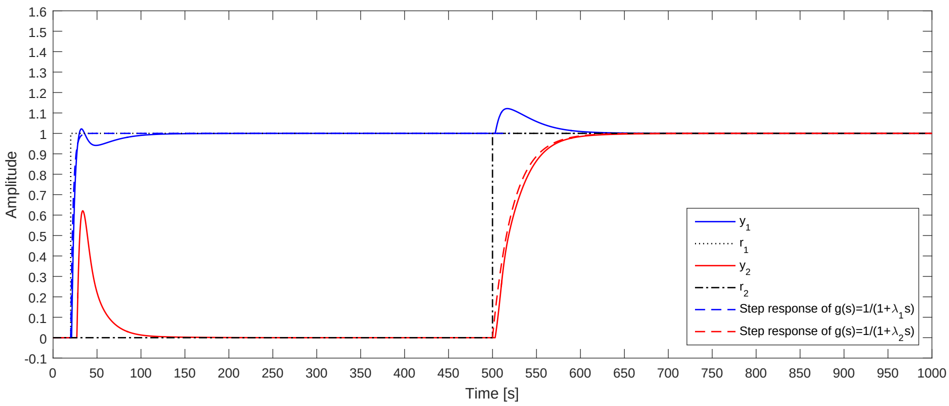


Figure 4.12: Closed loop response to changes in setpoints for the Wood and Berry column with controllers from the case with $\lambda_1 = 2.65s$ and $\lambda_2 = 22.82s$.

Table 4.8: Ratio between IE and IAE for loop 1 after a unit step in r_1 for the Wood and Berry column in closed loop with converged PI-controller parameters for different cases.

	$\frac{IE_1}{IAE_1}$	λ_2 [s]										
		1.44	2.28	3.62	5.73	9.09	14.40	22.82	36.17	57.33	90.86	144.00
λ_1 [s]	1.67	0.65	0.82	0.87	0.92	0.95	0.97	0.98	0.99	0.99	0.99	0.99
	2.65	0.72	0.67	0.70	0.77	0.87	0.86	0.97	1.00	1.00	1.00	1.00
	4.19	0.87	1.00	1.00	0.99	0.99	1.00	1.00	1.00	1.00	1.00	1.00
	6.65	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	10.54	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	16.70	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	26.47	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	41.95	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	66.48	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	105.37	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	167.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00

Table 4.9: Ratio between IE and IAE for loop 2 after a unit step in r_2 for the Wood and Berry column in closed loop with converged PI-controller parameters for different cases.

	$\frac{IE_2}{IAE_2}$	λ_2 [s]										
		1.44	2.28	3.62	5.73	9.09	14.40	22.82	36.17	57.33	90.86	144.00
λ_1 [s]	1.67	0.60	0.65	0.82	0.92	0.98	1.00	1.00	1.00	1.00	1.00	1.00
	2.65	0.64	0.64	0.74	0.91	0.98	1.00	1.00	1.00	1.00	1.00	1.00
	4.19	0.76	0.73	0.87	0.98	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	6.65	0.74	0.75	0.86	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	10.54	0.77	0.79	0.84	0.94	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	16.70	0.80	0.82	0.89	0.94	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	26.47	0.81	0.84	0.90	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	41.95	0.82	0.86	0.95	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	66.48	0.83	0.88	0.93	0.96	0.99	1.00	1.00	1.00	1.00	1.00	1.00
	105.37	0.82	0.89	0.95	0.97	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	167.00	0.82	0.89	0.96	0.98	1.00	1.00	1.00	1.00	1.00	1.00	1.00

No oscillations around the setpoint or overshoot when control loops are tuned for slow control relative to open loop time constants

As shown in Table 4.8, for all $\lambda_1 = 6.65$ s or larger (compared to the open loop time constant 16.7s for $g_{11}(s)$) there is neither overshoot nor oscillations of y_1 around the setpoint after a unit step in r_1. The smallest $\frac{IE}{IAE}$ ratio is found in the case with the fastest control of both loop 1 and 2. For $\lambda_2 = 14.4$ (the same as the open loop time constant of $g_{22}(s)$) or larger, it can be seen in Table 4.9 that there are no oscillations or overshoot around the setpoint for loop 2. The smallest value of $\frac{IE}{IAE}$ for loop 2 can be found for the fastest control of both loops, i.e. $\lambda_1 = 1.67$ s and $\lambda_2 = 1.44$ s.

5 Discussion

In this chapter the proposed iterative method and the SISO design method based on the least-squares fitting algorithm combined with λ-tuning will be discussed. The results from the previous chapter will be investigated and some discussion with respect to the methods of analysis (rise time T_r and the ratio between the integrated error and integrated absolute error) will be performed. Also, some insights in the use of EOPs are presented.

5.1 SISO design methods in the iterative method

It was evident from the application of the iterative method to the system with gas flow through a pipe that it would be problematic to use SIMC as a SISO design method with its inability to deal with poles and zeros with imaginary parts. There is also a problem with internal time delays. When forming EOPs for the Wood and Berry distillation column there will be time delays present in the denominators since there are time delays in the transfer function matrix elements $g_{11}(s)$ and $g_{22}(s)$ (see Equations (2.3) and (2.4)). This cannot be dealt with in SIMC.

In SIMC, the open loop transfer functions or the EOPs must be known or modeled. The least-squares algorithm with λ-tuning is in a sense a more complete SISO design method since with the FOPTD approximation it has a built-in way of obtaining transfer functions. Since the least-squares algorithm utilises data gathered from step response tests it is better at capturing the work carried out by the control engineers at Perstorp when tuning controllers. Their main tool to extract information about systems are step response tests. However, the modelling of transfer functions used for SIMC could of course be based on step response tests as well.

5.2 Least-squares algorithm for FOPTD approximations

It was investigated if the least-squares (LS) algorithm reflects how a FOPTD approximations are made according to the control engineers at Perstorp. In the approximations made of the three transfer functions in Table 3.1 it was shown that the LS algorithm always provided a faster time constant than what was provided by the control engineers. There are apparent differences between the LS algorithm and what the control engineers would do which directly result in a difference in the λ-tuned controllers. An interesting aspect is the fact that for all $\frac{T_r}{ln(9)}$-values in Table 4.1 controllers from the LS algorithm yielded closed loop responses more in line with the specified closed loop time constant λ than controllers from the control engineers. This could perhaps give an indication that the control engineers at Perstorp should rethink their FOPDT approximations rather than disqualifying the LS algorithm.

The bandpass filter and the simulation time for the LS algorithm have a direct effect on the approximation made since the cases for "slow control" and "fast control" give different FOPDT approximations. This can be stated with confidence since the only difference between the cases in the LS algorithm is the specified λ which only affects simulation time and the bandpass frequency of the filter.

5.3 The iterative method for TITO systems

With the limitations in SIMC it was decided to only use the LS algorithm with λ-tuning when applying the iterative method to the two reference systems, gas flow through a pipe and the Wood and Berry column. In all investigated cases (different combinations of desired response times of the two loops for the two systems) the iterative method converged. Convergence here means that the designed PI-controller parameters in the current iteration remain within $\pm 5\%$ of the values from the previous iteration. In the majority of the cases convergence was achieved already within three to four iterations. A possible explanation can be how the LS algorithm responsible of the FOPTD approximation works. To emulate some conservatism and that a control engineer would probably see if his/her last made approximation still seems to have a good fit, the LS algorithm reuses the FOPTD approximation from the previous iteration as a starting point in the nonlinear least-squares fitting algorithm. The MATLAB function 'lsqnonlin' used to minimize the sum of squared errors, starts in the user provided starting point and then iterates through different values of the FOPTD parameters through a steepest descent search. The iterations stops if a minimum is found but they will also stop if either the change in the sum of squared errors or the difference in FOPTD parameters between two iterations are below certain thresholds. Thus, the algorithm does not necessarily find the global minimum of the sum of squared errors. Hence, it is possible that the resulting FOPTD parameters when the iterative method has converged are not the best least-squares fit since they might not provide the global minimum for the sum of squared errors.

5.3.1 Closed loop properties for two TITO systems

The objective in λ-tuning is to achieve a first order closed loop response from setpoint to output with time constant λ. The purpose of calculating $\frac{Tr}{ln(9)}$ is to provide a measure reflecting this. The ratio between IE and IAE shows whether the system deviate from first order dynamics by indicating if there is any overshoot or oscillations around the setpoint. For the gas flow through a pipe system the $\frac{Tr}{ln(9)}$ values were relatively close to the specifed λ for each loop (see Tables 4.2 and 4.3). However in many cases the values were significantly smaller than 1 and in no case were the ratios equal to 1 for both loops. This is probably caused by the zero in the $g_{22}(s)$ transfer function. Overall the LS algorithm with lambda tuning achieved from a practical viewpoint closed loop behaviour close enough to what is desired in λ-tuning. For the Wood and Berry column the largest deviation from the specified λ is 87% for loop 1 and 61% for loop 2. Almost all of the $\frac{IE}{IAE}$ ratios for both loops are 1 or close to 1. The difference between actual and desired response time of the closed loop response is not too far off and it can be said that the objective of λ-tuning is fulfilled quite well. It is important to point out that for a TITO system to perfectly have a first order closed loop response in each loop with time constant λ_i, the open loop system must have first order dynamics without interactions (i.e. two separate SISO systems). Also there cannot be any time delays present [22].

For both reference systems studied the resulting controller parameters from the iterative method, with the LS algorithm followed by λ-tuning, achieved closed loop stability for all different tuning cases. Apart from the strong interactions through $g_{12}(s)$ and $g_{21}(s)$ and the time delays for the Wood and Berry column the systems do not have any hard-to-control open loop dynamics since there are neither unstable poles nor unstable zeros.

5.3.2 Limitations of the iterative method

The iterative method with the LS algorithm has limitations in what types of systems it can be used for. First, since the SISO design is based on a step response test between the control pair y_i and u_i, the open loop transfer function of the pair must be stable. This means that the transfer functions on the main diagonal of Equation (2.2) must be stable. Secondly, the individual control loop j that is currently closed must be stable, $1 + g_{jj}(s)c_j(s)$ cannot have RHP zeros, i.e. the denominators of the second terms in the EOPs in Equations (2.3) and (2.4). Finally, the same equations also express that the interaction terms $g_{12}(s)$ and $g_{21}(s)$ must be stable.

5.4 Effective open-loop processes, EOPs

The iterative method for TITO system uses step response tests of the system with one loop closed. In effect the step responses will show the dynamics of EOPs stated in Equations (2.3) and (2.4). The PI-controller design methods used (SIMC and λ-tuning) are based on approximating a dynamic system as first order with time delay. It is common in process industry to model systems as first order whenever general controllers are to be designed, not just decentralised PI-controllers. However, for controller design the approximations used should give an accurate enough description of the actual system. For example, approximating an unstable oscillatory second order system by a stable first order system is probably not a good idea.

5.4.1 Complicated dynamics might arise in EOPs

It can be seen that quite simple open loop dynamics in a TITO system controlled by PI-controllers can result in high order dynamics in EOPs. This can be exemplified using the expression for $g_{1,EOP}(s)$ in Equation (2.3), which is repeated here for clarity:

$$g_{1,EOP}(s) = g_{11}(s) - \frac{g_{12}(s)c_2(s)g_{21}(s)}{1 + g_{22}(s)c_2(s)}.$$

It assumed that each transfer function (e.g. $g_{ij}(s)$, $c_2(s)$) is rational in the Laplace variable s. Using the notation p_g for the denominator polynomial for any rational transfer function g, the roots of the polynomial are the poles of the transfer function. Let z_g denote the numerator polynomial for any rational transfer function g, then the roots of the polynomial are the zeros of g. Now, $g_{1,EOP}(s)$ can be rewritten as

$$g_{1,EOP}(s) = \frac{z_{g_{11}(s)}}{p_{g_{11}(s)}} - \frac{\frac{z_{g_{12}(s)}}{p_{g_{12}(s)}} \frac{z_{g_{22}(s)}}{p_{g_{22}(s)}} \frac{z_{g_{21}(s)}}{p_{g_{21}(s)}}}{1 + \frac{z_{c_2(s)}}{p_{c_2(s)}} \frac{z_{g_{22}(s)}}{p_{g_{22}(s)}}} \tag{5.1}$$

Collecting all terms in Equation (5.1) in a single rational transfer function gives

$$g_{1,EOP}(s) = \frac{z_{g_{11}(s)}p_{g_{12}(s)}p_{g_{22}(s)}p_{g_{21}(s)}p_{c_2(s)} + z_{c_2(s)}z_{g_{22}(s)}z_{g_{11}(s)}p_{g_{21}(s)}p_{g_{12}(s)} - z_{g_{12}(s)}z_{c_2(s)}z_{g_{21}(s)}p_{g_{22}(s)}p_{g_{11}(s)}}{p_{g_{12}(s)}p_{g_{21}(s)}p_{g_{11}(s)}\left(p_{c_2(s)}p_{g_{22}(s)} + z_{c_2(s)}z_{g_{22}(s)}\right)} \tag{5.2}$$

Assuming there are no common poles or zeros between the transfer functions in Equation (5.2) it is evident that the orders of the numerator and denominator can become quite large even with low order dynamics in the open loop transfer functions $g_{ij}(s)$.

Example

As an example, assume that each open loop transfer function in a TITO system (i.e. $g_{11}(s)$, $g_{12}(s)$, $g_{21}(s)$ and $g_{22}(s)$) is of first order with no zeros. In other words, each transfer function has one pole. The controller $c_2(s)$ is a PI-controller, which has one pole and one zero (see Equation (2.1)). The number of poles for $g_{1,EOP}(s)$ in Equation (5.2) is determined by the order of the polynomial $p_{g_{12}(s)}p_{g_{21}(s)}p_{g_{11}(s)}p_{c_2(s)}p_{g_{22}(s)}$ which is then 5. The number of zeros amounts to the order of the polynomial $z_{g_{11}(s)}p_{g_{12}(s)}p_{g_{22}(s)}p_{g_{21}(s)}p_{c_2(s)}$, which is 4. In summary, the EOP has quite complicated high order dynamics even though the open loop dynamics is very simple. Thus, it might not be feasible to approximate this high order dynamics as first order in a FOPTD approximation since too much information of the system is overlooked when the approximation is used for controller design. It is important to note that cancellations in Equation (5.2) can occur. If a TITO system share common poles and zeros in the open loop transfer functions $g_{11}(s)$, $g_{12}(s)$, $g_{21}(s)$ and $g_{22}(s)$ there will be cancellations between the numerator and denominator meaning the EOP will not necessarily have a much higher order than the open loop dynamics.

5.4.2 Tuning controllers for closed loop stability of a TITO system

The closed loop poles of a TITO system with a process transfer function matrix $G(s)$ and controller $C(s)$ can be found by finding the solutions to the closed loop characteristic equation:

$$\det(I + G(s)C(s)) = 0 \tag{5.3}$$

Where I is a 2-by-2 identity matrix. If the controller $C(s)$ only have nonzero elements along its main diagonal, i.e. a decentralised controller (as in Equation (2.2)), and using the expressions for EOPs in Equations (2.3) and (2.4), [14, 27] have shown that the following two equations are equivalent to each other and also equivalent to the characteristic Equation (5.3):

$$1 + g_{1,EOP}(s)c_1(s) = 0 \tag{5.4}$$

$$1 + g_{2,EOP}(s)c_2(s) = 0 \tag{5.5}$$

Equations (5.4) and (5.5) represent the denominators in the transfer functions from r_1 to y_1 and r_2 to y_2, respectively, and their solutions are the transfer functions' poles. They also show how SISO design problems in designing $c_1(s)$ and $c_2(s)$ based on EOPs can be formulated. Since Equations (5.3), (5.4) and (5.5) are equivalent, any SISO design method that can ensure closed loop stability of an EOP will also ensure closed loop stability of the TITO system. This shows the viability in reducing a TITO design problem to two SISO design problems from a stabiliy perspective! NB: the number of solutions to Equations (5.3), (5.4) and (5.5) will only be finite if there are no time delays present.

5.5 T_r and $\frac{IE}{IAE}$ as a measure of closed loop behaviour

The ratio between IE and IAE provide a measure that captures if there is any overshoot present. It does however only indicate oscillations of the output *around* the setpoint (i.e.

if there are sign changes in the control error). For example the response of y_1 to a unit step in r_1 in Figure 4.10 has some oscillations but they are not captured by the $\frac{IE}{IAE}$ measure. One of the strengths in using T_r, IE and IAE is that they can be easily calculated from data gathered during step response tests of real life systems.

5.6 Suggestions for future work

The main contribution of this thesis is an attempt to make an algorithm of how a control engineer at Perstorp iteratively tunes PI-controllers in a decentralised control scheme for TITO systems. The evaluation is based on a few example systems and transfer functions. A suggestion would be to make a more general analysis based on different classes of TITO systems. Examples of such classes could be certain types of dynamics (such as systems containing pure integrations, second order dynamics, etcetera) or the severity of interactions (as measured by for example relative gain arrays or the structured singular value [20]). The goal would then be to establish for what classes the iterative method will work. It was shown for the cases studied that closed loop stability was always achieved. However, the matter of just showing stability is often not enough within process control. Rather it is the matter of robust stability and performance that is of importance. Since the iterative method utilise SISO design it would be of interest to investigate if SISO robustness margins can guarantee some degree of MIMO robustness margins.

6 Conclusion

The main aim of this project has been to investigate an iterative method to tune two SISO PI-controllers for TITO systems. The method reduced the TITO controller design problem to two SISO design problems which were iterated. This reduction was based on open loop step response tests of one of the control loops with the other loop closed. The point being that, through the dynamics of the loop that was closed, the step response test also captured interactions between the two loops stemming from the TITO dynamics. A PI-controller was then tuned based on the test. The same type of step response test was then repeated but for the other loop to tune the other PI-controller. The problem with this procedure, however, is that the controller parameters in the loop currently closed affect the step response test meaning that any change in controller parameters for one loop requires retuning of the controller in the other loop. Hence the two PI-controllers must be iteratively tuned and re-tuned until their tunings converge. The iterative method has been investigated using simulations in Simulink and in order to generate PI-controllers two different design methods were examined; Skogestad IMC (SIMC) and λ-tuning.

Using the convention of effective open-loop process (EOP) it has been shown that low order open loop dynamics in a TITO system can lead to complex dynamics. In the iterative method step response tests are carried out with one control loop closed and the dynamics from interactions between loops and the PI-controller will result in the step responses exhibiting high order dynamics. The PI-controller tuning methods used require a first order with time delay model. Thus, the step responses must be approximated to first order responses in order to use these tuning methods.

SIMC is ruled out as PI-controller design method for the iterative method. SIMC provides its own set of rules to reduce higher order dynamics to first order. However, the tuning method can only handle dynamics with poles or zeros that are real valued. For one of the TITO systems studied (gas flow through a pipe) it was shown that in any realistic tuning case the EOP transfer function for one of the loops contained zeros with imaginary parts. It is difficult to know apriori if a TITO system with PI-controllers will not generate EOPs with poles or zeros having imaginary parts. The conclusion is that SIMC is problematic to use in the iterative method since whenever transfer functions with poles or zeros with imaginary parts arise, SIMC is unable to deal with this.

The iterative method is a viable procedure to tune decentralised controllers for closed loop stability of TITO systems. Equations (5.3),(5.4) and (5.5) are equivalent and when solved they provide the exact same roots, i.e. the closed loop poles of the TITO system. This goes to show that reducing the two variable TITO design problem to two SISO problems which are iterated is not a problem in guaranteeing overall closed loop stability. If PI-controller design which guarantees SISO closed loop stability is used in the iterative method, the stability of the closed loop TITO system is ensured. From a mathematical perspective the TITO system is stable if and only if any effective open-loop process is closed loop stable with its corresponding PI-controller.

It is suggested that the control engineers at Perstorp should reconsider their intuition when approximating a step response as a first order system. It was shown for three example transfer functions that the implementation in MATLAB of a least-squares algorithm to make FOPTD approximations of step response tests did not mimic the approximations made by control engineers at Perstorp. PI-controllers tuned with λ-tuning based on the approximations from the algorithm were, however, better at achieving the closed loop objectives in λ-tuning than the PI-controllers based on the FOPTD approximation made by the control engineers. Thus, the least-squares algorithm did not really reflect the design philosophy of the control engineers at Perstorp.

No proof has been found under what circumstances the iterative method converge with respect to what kind if TITO system is studied and PI tuning method used. The iterative method with λ-tuning was evaluated using two reference TITO systems, pressure and flow control of gas flow through a pipe and the Wood and Berry binary distillation column. The λ-tuning required data from step response tests to be modeled as first order plus time delay (FOPTD) and this was achieved using a least-squares algorithm. For the gas flow through a pipe system 100 tuning cases were investigate through 10 different specifications on the tuning parameter λ for each control loop. The chosen values of the tuning parameters λ_1 and λ_2 were related to the open loop time constants. In the case of the Wood and Berry system 121 different tuning cases were investigated; 11 different values on λ for each loop, related to known open loop time constants, were investigated. Convergence of the iterative method was achieved in all cases for both systems. Closed loop properties were also shown to be in line with the desired objectives in λ-tuning, and closed loop stability was ensured for all cases studied. However, it has not been established exactly why the iterative method converges in the investigated cases.

Bibliography

[1] Åström, Karl J. and Hägglund, Tore. *PID Controllers - Theory, Design and Tuning.* North Carolina: Instrument Society of America, 1995.

[2] Glad, T. and Ljung, L. *Control Theory: multivariable and nonlinear methods.* London: Taylor & Francis, 2000.

[3] Halevi, Y., Palmor, Z.J., and Efrati, T. "Automatic tuning of decentralized PID controllers for MIMO processes". In: *Journal of Process Control* 7.2 (Jan. 1997), pp. 119–128.

[4] Hovd, M. and Skogestad, S. "Improved independent design of robust decentralized controllers". In: *Journal of Process Control* 3.1 (Feb. 1993), pp. 43–51.

[5] URL: `https://www.perstorp.com/en/about/site_stenungsund/` (visited on 05/10/2016).

[6] Lee, Jietae, Kim, Dong Hyun, and Edgar, Thomas F. "Static decouplers for control of multivariable processes". In: *AIChE Journal* 51.10 (2005), pp. 2712–2720.

[7] Hovd, M. and Skogestad, S. "Sequential design of decentralized controllers". In: *Automatica* 30.10 (Oct. 1994), pp. 1601–1607.

[8] Skogestad, S. and Morari, M. "Robust performance of decentralized control systems by independent designs". In: *Automatica* 25.1 (1989), pp. 119–125.

[9] Huang, Hsiao Ping et al. "A direct method for multi-loop PI/PID controller design". In: *Journal of Process Control* 13.8 (2003), pp. 769–786.

[10] Skogestad, S. and Postlethwaite, I. *Multivariable Feedback Control Analysis and Design.* 2nd edition. Chichester: John Wiley and Sons, Ltd, 2005.

[11] Zhu, Z.X. and Chiu, M.S. "Dynamic Analysis of Decentralized 2×2 Control Systems in Relation to Loop Interaction and Local Stability". In: *Industrial and Engineering Chemistry Research* 37.2 (1998), pp. 464–473.

[12] Vu, Truong Nguyen Luan and Lee, Moonyong. "Independent design of multi-loop PI/PID controllers for interacting multivariable processes". In: *Journal of Process Control* 20.8 (Sept. 2010), pp. 922–933.

[13] Liu, Qie and Jin, Qibing. "Multi-loop PI/PID controllers design for disturbance rejection based on non-parametric effective model and non-convex optimisation". In: *IET Control Theory & Applications* 8.15 (2014), pp. 1499–1512.

[14] Shen, Shih-Haur and Yu, Cheng-Ching. "Use of relay-feedback test for automatic tuning of multivariable systems". In: *AIChE Journal* 40.4 (1994), pp. 627–646.

[15] Euzebio, Thiago A.M. and Barros, Pericles R. "Iterative Procedure for Tuning Decentralized PID Controllers". In: *IFAC-PapersOnLine* 48 (2015), pp. 1180–1185.

[16] Economou, Constantin G and Morari, Manfred. "Internal Model Control. 6. Multiloop Design". In: *Industrial & Engineering Chemistry Research* 25.2 (1986), pp. 411–419.

[17] Garcia, Carlos E and Morari, Manfred. “Internal Model Control . 1 . A Unifying Review and Some New Results”. In: *Industrial & Engineering Chemistry Design and Development* 21 (1982), pp. 308–323.

[18] Rivera, D. E., Morari, M., and Skogestad, S. “Internal model control: PID controller design”. In: *Industrial & Engineering Chemistry Process Design and Development* 25.1 (Jan. 1986), pp. 252–265.

[19] Skogestad, S. “Simple analytic rules for model reduction and PID controller tuning”. In: *Journal of Process Control* 13.4 (2003), pp. 291–309.

[20] Skogestad, Sigurd. “Tuning for smooth PID control with acceptable disturbance rejection”. In: *Industrial and Engineering Chemistry Research* 45.23 (2006), pp. 7817–7822.

[21] Skogestad, Sigurd and Grimholt, Chriss. “The SIMC method for smooth PID controller tuning”. In: *PID Control in the Third Millennium*. Ed. by Vilanova, R and Visioli, A. London: Springer-Verlag, 2012. Chap. 5, pp. 147–175.

[22] Forsman, Krister. *Reglerteknik för processindustrin*. Lund: Studentlitteratur, 2005.

[23] F. Vázquez, F. Morilla, S. Dormido. “an Iterative Method for Tuning Decentralized Pid Controllers”. In: *14th IFAC WORLD CONGRESS* (1999), pp. 491–496.

[24] Glad, Torkel and Ljung, Lennart. *Reglerteknik Grundläggande teori*. 4th edition. Lund: Studentlitteratur AB, 2006.

[25] Garpinger, Olof, Hägglund, Tore, and Åström, Karl Johan. “Performance and robustness trade-offs in PID control”. In: *Journal of Process Control* 24.5 (2014), pp. 568–577.

[26] Wood, R K and Berry, M W. “Terminal composition control of a binary distillation column”. In: *Chemical Engineering Science* 28.9 (1973), pp. 1707–1717.

[27] Balchen, Jens G. “The Stability of 2x2 Multivariable Control Systems”. In: *Modeling, Identification and Control: A Norwegian Research Bulletin* 11.2 (1990), pp. 97–108.

A

A.1 Derivation of the λ-tuning method

Here it will be shown why choosing the controller parameters according to the λ-method for a process modeled as FOPTD will result in a closed loop transfer function that is approximately also FOPTD. Utilizing the equation of the process model and the PI-controller, i.e. Equations (2.6) and (2.1), the closed loop transfer function is given by

$$g_c(s) = \frac{g(s)c(s)}{1+g(s)c(s)} = \frac{K_c K(1+sT_i)e^{-sL}}{(1+sT)sT_i + K_c K(1+sT_i)e^{-sL}} \tag{A.1}$$

Approximating the time delay in the denominator by a first order taylor approximation $e^{-sL} \approx 1 - sL$ gives

$$g_c(s) \approx \frac{K_c K(1+sT_i)e^{-sL}}{(1+sT)sT_i + K_c K(1+sT_i)(1-sL)} \tag{A.2}$$

Choosing the integral time of the controller according to λ-tuning, $T_i = T$ (cancelling the process pole with a controller zero), Equation (A.2) is simplified to

$$g_c(s) \approx \frac{K_c K e^{-sL}}{sT + K_c K(1-sL)} \tag{A.3}$$

By assigning $K_c = \frac{T}{K(\lambda+L)}$, Equation (A.3) becomes

$$g_c(s) \approx \frac{Te^{-sL}}{(\lambda+L)(sT + \frac{T(1-sL)}{(\lambda+L)})} = \frac{e^{-sL}}{(\lambda+L-L)s+1} = \frac{e^{-sL}}{\lambda s+1}. \tag{A.4}$$

Equation (A.4) shows that the closed loop transfer function has a time constant equal to λ and the equation is equal to Equation (2.5).

B

B.1 λ-tuning based on FOPTD approximations of three transfer functions

Table B.1 shows the numerical values of the first order plus time delay approximations in Figures 4.2, 4.3 and 4.4 in Chapter 4.

Using PI-controller parameters according to Table B.1, closed loop tests were conducted through unit setpoint changes. A graphical representation of the results can be seen in Figures B.1, B.2 and B.3.

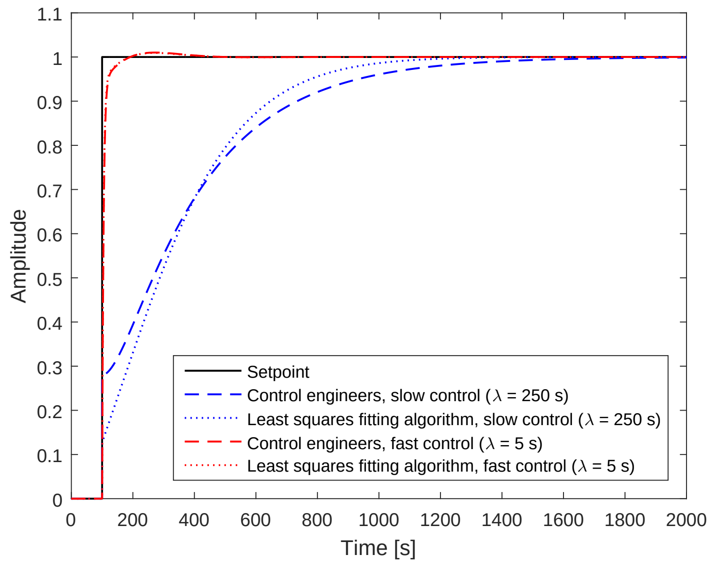


Figure B.1: Closed loop response of $G_1(s)$ in closed loop with four different PI-controllers. Unit setpoint change at time = 100 s.

B.2 PI-controller parameters from the iterative method for the gas flow through a pipe system

When the iterative method was used for the gas flow through a pipe system all cases investigated resulted in convergence in the series PI-controller paremeters. The values of these parameters can be seen in Table B.2. These are the values used for the investigation of closed loop properties in Section 4.3.3.

Table B.1: FOPTD approximations of three transfer functions according the control engineers at Perstorp and the least squares algorithm.

		FOPTD-approximation $g(s) = \frac{K}{1+sT}e^{-\theta s}$			λ-tuning: PI-controller		
	λ [s]	K	T [s]	θ [s]	Kc	Ti [s]	Comments
$G_1(s) = \frac{5000s^2+100s+1}{(50s+1)(100s+1)}$							
"Slow", control engineers	250	1	97	0	0.388	97	-
"Slow", least squares algorithm	250	0.957	34.77	0	0.145	34.77	-
"Fast", control engineers	5	1	0	0	-	-	I-controller: $C(s) = \frac{0.2}{s}$
"Fast" least squares algorithm	5	0.919	$0.224{\cdot}10^{-2}$	$0.187{\cdot}10^{-2}$	$4.883{\cdot}10^{-4}$	$0.224{\cdot}10^{-2}$	in practice an I-controller
$G_2(s) = \frac{(15s+1)}{(8s+1)(5s+1)}e^{-10s}$							
"Slow", control engineers	60	1	10	10	0.143	10	
"Slow", least squares algorithm	60	1.013	$0.781{\cdot}10^{-1}$	9.527	$0.111{\cdot}10^{-2}$	$0.781{\cdot}10^{-1}$	
"Fast", control engineers	15	1.22	2.5	10	$0.820{\cdot}10^{-1}$	2.5	
"Fast" least squares algorithm	15	1.079	0.509	11.01	$0.181{\cdot}10^{-1}$	0.509	
$G_3(s) = \frac{(22s+1)(16s+1)}{(45s+1)(8s+1)(2s+1)}$							
"Slow", control engineers	100	1	32	0	0.32	32	
"Slow", least squares algorithm	100	0.973	12.76	0	0.131	12.76	
"Fast", control engineers	7	0.82	1.5	0	0.261	1.5	
"Fast" least squares algorithm	7	0.793	1.022	0.297	0.177	1.022	

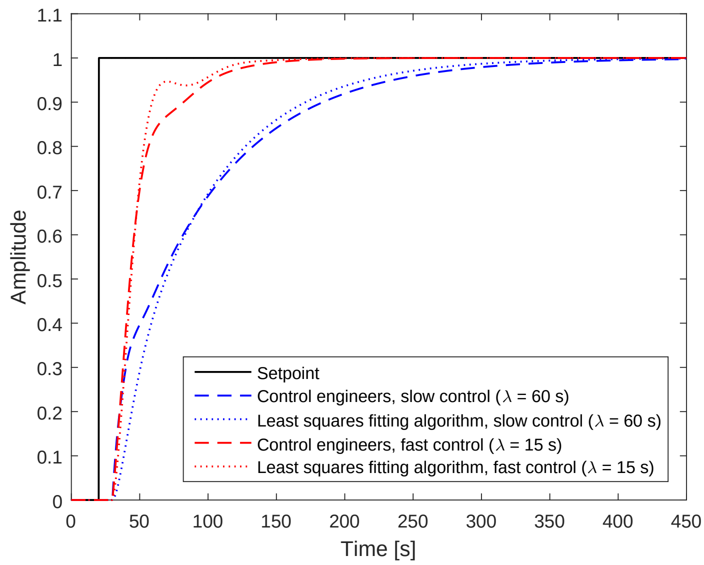


Figure B.2: Closed loop response of $G_2(s)$ in closed loop with four different PI-controllers. Unit setpoint change at time = 20 s.

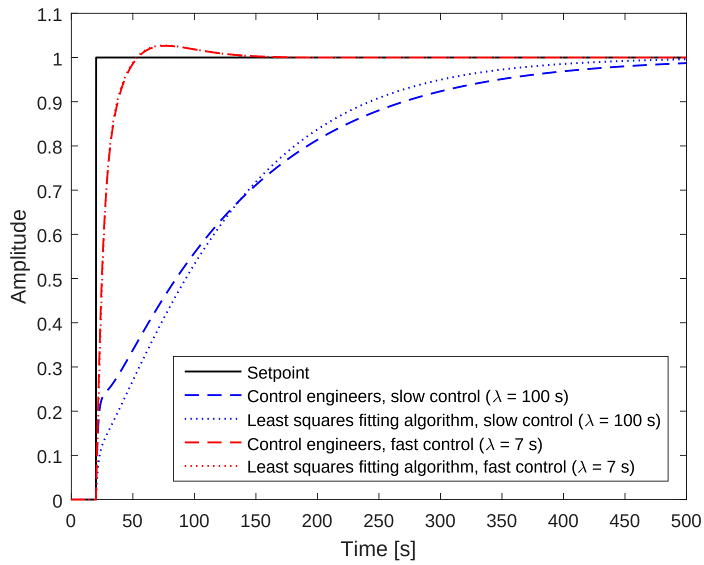


Figure B.3: Closed loop response of $G_3(s)$ in closed loop with four different PI-controllers. Unit setpoint change at time = 20 s.

Table B.2: Parameters of two PI-controllers for the gas flow through a pipe system as found through the iterative method.

Kc_1 Ti_1 Kc_2 Ti_2	λ_2 [s]									
λ_1 [s]	5.00	60.00	115.00	170.00	225.00	280.00	335.00	390.00	445.00	500.00
5.00	40.8	40.5	40.2	40.2	39.7	40.3	40.2	40.2	40	40.1
	96.5	59.2	55.3	53.7	52.6	52.5	52	51.8	51.4	51.3
	1.84E-14	0.0177	0.0215	0.0178	0.0113	0.0114	0.000516	0.0087	0.00787	1.22E-08
	4.46E-14	0.525	1.23	1.51	1.27	1.59	0.0869	1.7	1.75	3.06E-06
60.00	3.42	4.6	2.98	4.92	4.93	3.74	4.1	4.01	4.36	4.25
	104	133	67.2	108	108	78.3	82.1	77.9	84.3	77
	9.72E-15	8.86E-16	0.14	0.0865	0.142	0.074	0.0641	0.116	0.124	0.0399
	2.24E-14	2.24E-14	7.16	6.97	15.3	10	10.7	22.5	27.2	10
115.00	1.29	1.2	1.48	1.46	3.41	3.2	3.32	3.05	2.54	2.8
	72.4	58.5	66.8	62.1	161	146	137	125	93.8	108
	9.78E-15	9.96E-16	0.0424	0.0441	0.0983	0.264	0.082	0.248	0.00795	0.0473
	2.24E-14	2.24E-14	1.99	3.41	10	34.1	13.1	46.4	1.84	12.1
170.00	1.06	0.746	0.768	1.97	1.84	2.16	1.92	2.43	1.55	2.01
	89.6	58.5	57.4	156	130	150	133	158	93.8	121
	1.94E-14	1.06E-15	5.24E-16	0.333	0.323	0.38	0.409	0.357	0.000116	0.205
	4.45E-14	2.24E-14	2.24E-14	23	32.7	47.5	62.2	64.5	0.026	49
225.00	0.925	0.78	0.461	0.471	0.735	2.36	1.41	2.08	0.958	2.42
	104	89.5	46.5	44.9	68.2	237	126	186	76.7	211
	9.79E-15	2.15E-15	5.53E-16	0.155	0.543	0.194	0.378	0.408	0.0473	0.214
	2.24E-14	4.46E-14	2.24E-14	9.98	50	23.4	57.3	71.4	9.99	49.9
280.00	0.444	0.411	0.693	1.36	0.508	1.49	0.533	0.469	0.582	0.628
	66.6	55.9	92.4	187	60.8	209	59.1	49.6	59.2	62
	1.95E-14	1.12E-15	5.64E-16	0.0915	0.464	0.179	0.369	0.366	0.00111	0.131
	4.45E-14	2.24E-14	2.25E-14	5.87	40.3	20	53.7	68.1	0.239	30
335.00	0.65	0.53	0.358	1.12	0.376	0.392	0.408	0.57	0.357	0.463
	110	86.2	56.8	190	54.1	55	54.9	75.2	44.4	56.8
	1.93E-14	2.22E-15	1.17E-15	0.0405	0.121	0.356	0.324	0.0586	0.0489	0.0629
	4.43E-14	4.46E-14	4.46E-14	2.5	10	40.6	45.8	10	10	14.1
390.00	0.461	0.306	0.33	0.431	0.271	0.471	1.1	0.377	0.314	0.319
	92.4	59	63.7	79.5	46.7	80.2	207	58.9	48.3	48.5
	9.81E-15	1.15E-15	6.02E-16	0.043	0.801	0.817	0.659	0.417	0.52	0.401
	2.24E-14	2.24E-14	2.24E-14	2.5	72.3	86.6	89.1	68.4	97.2	87.4
445.00	0.2	0.249	0.198	0.324	0.224	0.227	0.232	0.293	0.3	0.465
	45	54.8	42.9	68.9	46.2	44.6	44.4	54.4	54.5	87.1
	9.82E-15	1.16E-15	6.17E-16	0.939	0.738	0.782	0.763	0.617	0.695	0.495
	2.24E-14	2.24E-14	2.25E-14	53.3	58.3	83.2	100	98.2	130	106
500.00	0.398	0.43	0.196	0.131	0.155	0.166	0.355	0.221	0.372	0.976
	99.8	107	47.5	31.3	35.9	37.2	77.6	47.3	79.6	207
	1.95E-14	2.27E-15	6.23E-16	1.14	0.887	0.877	0.94	0.945	0.62	0.528
	4.45E-14	4.46E-14	2.24E-14	62.1	68.1	90.7	122	146	114	112

B.3 PI-controller parameters from the iterative method for the Wood and Berry distillation column system

In the case of the Wood and Berry column the iterative method lead to convergence in the PI-controller parameters for all cases studied. The value of these parameters can be found in Table B.3, which was used for the closed loop analysis in Section 4.3.4.

Table B.3: Parameters of two PI-controllers for the Wood and Berry column as found through the iterative method.

λ_1 [s]	Kc_1 / Ti_1 / Kc_2 / Ti_2	λ_2 [s] 1.44	2.28	3.62	5.73	9.09	14.40	22.82	36.17	57.33	90.86	144.00
1.67	Kc_1	0.352	0.392	0.401	0.423	0.433	0.45	0.461	0.471	0.478	0.482	0.482
	Ti_1	8.7	10.5	11	12.2	12.7	13.8	14.6	15.2	15.8	16.1	16.2
	Kc_2	-0.136	-0.0769	-0.0702	-0.0489	-0.0434	-0.0291	-0.0203	-0.0141	-0.00724	-0.00493	-0.00506
	Ti_2	8.2	4.37	4.78	4.28	5.23	5.03	5.14	5.44	4.31	4.55	7.22
2.65	Kc_1	0.281	0.184	0.19	0.197	0.213	0.239	0.31	0.36	0.356	0.356	0.356
	Ti_1	8.06	5.79	6.1	6.86	7.65	8.25	12	16.4	16.6	16.6	16.6
	Kc_2	-0.142	-0.0823	-0.07	-0.0431	-0.042	-0.029	-0.0202	-0.0142	-0.00724	-0.00494	-0.00506
	Ti_2	9	5.23	5.01	4.09	5.22	5.05	5.14	5.44	4.31	4.55	7.23
4.19	Kc_1	0.154	0.288	0.229	0.248	0.192	0.242	0.233	0.213	0.211	0.209	0.208
	Ti_1	5.82	13.9	11.8	11.7	9.34	13.3	13.4	13.8	13.8	13.9	13.9
	Kc_2	-0.152	-0.083	-0.0614	-0.0381	-0.0427	-0.0288	-0.0201	-0.0142	-0.00724	-0.00494	-0.00506
	Ti_2	10.7	5.12	4.69	3.74	5.25	5.1	5.11	5.44	4.31	4.55	7.22
6.65	Kc_1	0.236	0.241	0.255	0.251	0.248	0.203	0.2	0.176	0.162	0.158	0.155
	Ti_1	14.3	14.1	14.7	14.9	14.8	14.3	14.8	14.7	14.9	14.8	15
	Kc_2	-0.147	-0.087	-0.0585	-0.0384	-0.0417	-0.0278	-0.0192	-0.0141	-0.00724	-0.00493	-0.00507
	Ti_2	9.74	5.55	4.4	3.84	5.28	4.95	5.08	5.41	4.31	4.59	7.22
10.54	Kc_1	0.172	0.185	0.122	0.0949	0.0743	0.0476	0.0646	0.0607	0.0646	0.07	0.0823
	Ti_1	14.5	14.7	10.9	8.39	6.64	4.59	6.43	6.71	7.58	8.84	10.8
	Kc_2	-0.148	-0.094	-0.0621	-0.0367	-0.0351	-0.0253	-0.0191	-0.0123	-0.00725	-0.00495	-0.00507
	Ti_2	9.99	6.34	5.25	4.22	4.87	4.98	5.17	4.88	4.31	4.55	7.22
16.70	Kc_1	0.125	0.109	0.14	0.105	0.128	0.164	0.189	0.0646	0.0839	0.0823	0.0584
	Ti_1	15.9	13.3	15.8	13.4	14.5	19.1	19.1	8.22	10.4	15.2	11.7
	Kc_2	-0.156	-0.106	-0.0813	-0.0425	-0.0388	-0.0253	-0.0184	-0.0132	-0.00725	-0.00477	-0.00506
	Ti_2	11.5	7.84	6.65	4.81	5.49	4.69	4.84	5.36	4.31	4.48	7.21
26.47	Kc_1	0.0869	0.0857	0.0859	0.0823	0.0865	0.0868	0.0702	0.0266	0.0735	0.0188	0.0567
	Ti_1	16.2	15.9	16.1	15.3	15.8	16.5	14.1	5.84	14.1	4.84	15.8
	Kc_2	-0.159	-0.112	-0.0778	-0.0648	-0.0288	-0.0231	-0.00929	-0.00392	-0.00408	-0.00477	-0.00506
	Ti_2	12.2	8.78	7.19	7.79	4.75	4.88	2.75	1.67	2.5	4.53	7.21
41.95	Kc_1	0.0532	0.0543	0.0564	0.0521	0.0522	0.0539	0.0508	0.0443	0.0381	0.0459	0.0414
	Ti_1	15.5	15.7	15.5	15.5	14.5	15.8	14.8	13.8	12.9	16.4	14.5
	Kc_2	-0.162	-0.121	-0.106	-0.0676	-0.0416	-0.0253	-0.018	-0.0125	-0.00642	-0.00494	-0.00506
	Ti_2	13	10.2	10.4	8.63	6.95	5.77	5.19	5.42	4.25	4.55	7.21
66.48	Kc_1	0.0328	0.033	0.0333	0.0331	0.0315	0.0343	0.0346	0.0286	0.0135	0.0308	0.0261
	Ti_1	14.8	14.8	15	15	14.2	15	15.6	13.4	6.24	14.9	13.8
	Kc_2	-0.165	-0.127	-0.0903	-0.0567	-0.0339	-0.0256	-0.0167	-0.0106	-0.00631	-0.00378	-0.00496
	Ti_2	13.9	11.4	9.75	7.96	6.52	6.2	5.67	5.2	4.25	3.76	7.17
105.37	Kc_1	0.0229	0.0226	0.023	0.0231	0.0228	0.023	0.0232	0.022	0.0221	0.0172	0.0233
	Ti_1	15.8	15.7	15.9	15.9	15.7	15.8	15.7	15.7	16.2	13	15.8
	Kc_2	-0.165	-0.131	-0.0968	-0.0646	-0.0415	-0.0224	-0.0196	-0.0102	-0.00615	-0.00411	-0.00462
	Ti_2	13.9	12.2	10.9	9.41	8.11	6.26	7.31	5.4	4.29	4.42	7.01
167.00	Kc_1	0.0152	0.0153	0.0154	0.0153	0.0156	0.0152	0.0153	0.0151	0.0152	0.0151	0.014
	Ti_1	16.7	16.5	16.6	16.8	16.8	16.7	16.8	16.8	17.1	17.1	16.6
	Kc_2	-0.165	-0.134	-0.101	-0.0693	-0.0449	-0.0278	-0.0218	-0.0112	-0.00663	-0.00237	-0.00477
	Ti_2	13.9	12.8	11.9	10.6	9.33	8.05	8.56	6.46	5.14	2.77	7.11

C

C.1 Settings used in Matlab and Simulink for FOPTD approximations

Table C.1: Additional non-default options and parameters used for FOPTD approximations from least squares fitting.

Simulink model	
SolverType	'Variable-step'
Solver	'ode15s'
RelTol	1E-06
OutputOption	'SpecifiedOutputTimes'
OuputTimes	2000 evenly spaced points between 0 and the simulation time
lsqnonlin	
Algorithm	'trust-region-reflective'
TolX	1E-09
TolFun	1E-06