LUND UNIVERSITY

Automatic Tuning of PID Controllers Based on Asymmetric Relay Feedback

Berner, Josefin

2015

Automatic Tuning of PID Controllers based on Asymmetric Relay Feedback

Josefin Berner

LUND
UNIVERSITY

Department of Automatic Control

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

Abstract

This thesis presents an improved version of the classic relay autotuner. The proposed autotuner uses an asymmetric relay function to better excite the process in the experiment phase. The improved excitation provides the possibility to obtain better models and hence better tuning, without making the autotuner more complicated or time consuming.

Some processes demand more accurate modeling and tuning to obtain controllers of sufficient performance. The proposed autotuner can classify these processes from the experiment. In an advanced version of the autotuner an additional experiment could be designed for these processes, in order to further increase the possibilities in modeling and tuning. The experiment design would then rely on information from the relay experiment. A simple version of the autotuner could instead make a somewhat better model estimation immediately, or suggest that some extra effort may be put in modeling if the control performance of the loop is crucial. The main focus in this thesis is on the simple version of the autotuner.

The proposed autotuner uses the process classification for model and controller selection also in the simple version. The processes are classified according to their normalized time delays. In this thesis a simple method of finding the normalized time delay from the asymmetric relay experiment is presented and evaluated.

Research presented on different versions of the relay autotuner is often based solely on simulations. In large simulation environments, the ability to automatically tune the large amount of PID controllers is practical and time-saving. However, the ability to use the autotuner in an industrial setting, requires considerations not always present in a simulation environment. This thesis investigates many of these issues, regarding parameter settings and possible error sources. The proposed autotuner is implemented, tested and evaluated both in a simulation environment and by industrial experiments. The simple version of the autotuner gives satisfactory results, both in simulations and on the industrial processes. Still, there is a possibility to further increase the performance by an advanced version of the autotuner.

Acknowledgments

There are many people that in one way or another have helped me writing this thesis. I thank everyone at the Department of Automatic Control for making it a fun and inspiring workplace. I am very grateful for the possibility to collaborate with such amazing researchers and personalities as my supervisor Professor Tore Hägglund and Professor Karl Johan Åström. The endless source of enthusiasm and ideas that Karl Johan contributes with, is complemented in a perfect way by Tore's down-to-earth and practical attitude. The encouragement from both of you means a lot to me, and I am looking forward to the continuation of our work.

Dr. Mattias Grundelius at Schneider Electric Buildings AB has helped me with issues regarding the implementation of the autotuner in their software, as well as provided the possibility to run some experiments on their test facility in Malmö. Thanks to him the experiments were performed smoothly, and for this I am very grateful. I am also thankful to Professor Anders Rantzer for allowing me to work on this sidetrack of what I was supposed to work with. The sidetrack has now become a thesis.

Going to work every day would not be half as fun if it was not for my office colleagues Martin Hast, Olof Sörnmo and Andreas Stolt. I wish you all could write your theses a little slower, and stay at the department longer. The office will feel empty without you. The ability to discuss anything, from complicated work-related issues to pure nonsense, with the three of you is something I will miss. A special thanks is given to Martin Hast for convincing me to accept the PhD position during a late night of Singstar, I have not regretted it yet.

The department administrators Eva Westin, Ingrid Nilsson, Monika Rasmusson and Mika Nishimura, as well as former administrator Lizette Borgeram, are worthy of all possible thanks. They are always helpful and encouraging, no matter if your problem is of a personal nature or about a travel bill.

All my fellow PhD students and Postdocs are thanked for their contributions to the nice working environment and all the discussions concerning both work and life. The research engineers are much appreciated for their work on all technical issues.

I would also like to thank the Toughest-team, my handball team and my football team for trying to keep me in shape, in spite of all the candy-eating during the thesis

writing.

Last, but definitely not least, I thank my friends and family. My parents, sisters, nieces and nephews mean everything to me. The love and support from all of you is something I cherish, and I promise you that the next thing I write will be a fairytale.

Financial Support

Contents

Nomenclature

Here some notations and abbreviations used in the thesis are given.

Notation	Description
γ	Asymmetry level of the relay
d_1	Positive relay amplitude
d_2	Negative relay amplitude
ε	Convergence limit for relay experiment
h	Hysteresis of the relay
I_u	Integral of the relay output over one oscillation period
I_y	Integral of the process output over one oscillation period
K	Proportional gain of PID controller
K_p	Static gain of process
k_v	Gain of integrating process
L	Time delay of process
M_S	Maximum of the sensitivity function
M_T	Maximum of the complementary sensitivity function
n_0	Noise level
ρ	Half-period ratio
T	Time constant of process
τ	Normalized time delay
τ_s	Normalized time delay obtained from step response
t_{on}	Time period where the relay output is u_{on}
t_{off}	Time period where the relay output is u_{off}
u	Output signal from relay, control signal
u_{on}	Relay output when y is below the hysteresis band
u_{off}	Relay output when y is above the hysteresis band
y	Process output

Abbreviation	Description
FOTD	First Order Time Delayed
IAE	Integrated Absolute Error
IFOTD	Integrating plus First Order Time Delayed
ITD	Integrating Time Delayed
PID	Proportional Integral Derivative
SOTD	Second Order Time Delayed

1

Introduction

This thesis presents, investigates, and evaluates an automatic tuner for PID controllers based on an asymmetric relay feedback experiment. The aim is to find low-order models from the relay experiment and then use simple rules to tune controllers from the obtained models. The main objectives are that the autotuner should be fast and simple, yet give satisfactory results. The thesis also aims to give an opening for a more advanced version of the autotuner, that could provide more accurate modeling and controller tuning for processes with higher performance requirements.

1.1 Motivation

An industrial process facility may contain hundreds or thousands of control loops. The majority of these are using PID controllers. Even though the PID controller is simple, many of the controllers operating in industry today are performing unsatisfactory due to poor tuning of the controller parameters. This can be due to either lack of time, or lack of knowledge in control theory, among the staff. To have an automatic method of finding satisfactory controller parameters is therefore highly desirable. The method should ideally be fast and reliable, and should not require an extensive control education for the users. One method that has been successful in industry is the relay autotuner. The main advantage of the relay autotuner is that it is simple, fast, and does not require any (or little) prior process knowledge, since the relay feedback automatically excites the process in the frequency range interesting for PID control. A short experiment time is essential, not only to reduce the overall time-consumption, but also to minimize the risk of disturbances entering during the experiment. Since the original relay autotuner was presented in the mid-eighties [Åström and Hägglund, 1984], the increase in computational power as well as new insights in PID control, has provided the possibility to improve the relay autotuner.

Depending on the desired use of the autotuner, some different use cases can be established. One is to provide a simple, yet satisfactory, autotuner that should be able to run in stand-alone industrial systems with limited computational facilities. Another use case is to find an autotuner aimed for use in large simulation environ-

ments, where there are less restrictions on parameters, and no unforeseen disturbances. A third use case is to provide the best possible autotuner, with the assumption that computational power and time consumption are not restricted. This autotuner could use extensive system identification, add more experiments if needed, and also use optimization programs to find controller parameters.

The relay autotuner proposed in this thesis is mainly focused on the first use case. An improvement from the classic relay autotuner, is that the proposed one uses an asymmetric relay function to increase the excitation in the experiment. This gives better models without increasing the complexity or time consumption of the tuning process. A low-order transfer function model is obtained from the proposed autotuner, while the original autotuner only gave one frequency point. Another improvement is that the proposed autotuner uses a classification measure of the process to make automatic choices on model and controller selection. For many industrial processes the low-order model is sufficient. To put more time and effort to the modeling of all processes is therefore unnecessary. The process classification provides information on which processes may benefit significantly from more advanced modeling. The extra effort could then be restricted to these processes if the control performance of that loop is crucial.

1.2 Contributions

The main contributions of this thesis are:

- An automated procedure, including parameter choices, for an asymmetric relay feedback experiment.

- A simple method of classifying the process during the experiment.

- Automatic model and controller selection from process classification.

- Implementation and evaluation of the autotuner, both in simulations and on an industrial process.

1.3 Publications

Parts of this thesis are based on the following publications:

Berner, J., K. J. Åström, and T. Hägglund (2014). "Towards a New Generation of Relay Autotuners". In: *19th IFAC World Congr.*

Theorin, A. and J. Berner (2015). "Implementation of an Asymmetric Relay Autotuner in a Sequential Control Language". In: *IEEE Int. Conf. Autom. Sci. Eng.* Submitted.

1.4 Thesis outline

In this thesis an asymmetric relay autotuner is proposed for the tuning of PID controllers. In Chapter 2 the PID controller, and ways of tuning it, are described. The chapter also includes the model types that will be used in the thesis, and definitions of some important concepts. A description of the relay autotuner and its development is given. The next three chapters explain the proposed autotuner in detail. Chapter 3 contains definitions and equations for the asymmetric relay feedback experiment. It also explains how to get models from the experiment. The overall picture of the automatic tuning procedure is given in Chapter 4, and some practical issues are listed and discussed in Chapter 5. Subsequently come two chapters evaluating the performance of the autotuner. In Chapter 6 the evaluation is done in a simulation environment, while Chapter 7 explains and evaluates experiments performed on an air handling unit with an industrial control system. Conclusions from the thesis are summarized in Chapter 8, this chapter also contains some suggestions for future research.

2

Background

In this chapter some of the concepts used later in the thesis are described. The chapter starts with an explanation of the PID controller, followed by the model types that will be used in the thesis and that are commonly used for tuning PID controllers. The normalized time delay is defined and its use in an autotuner is explained in Section 2.3. In Section 2.4 some robustness and performance measures are defined and the used controller tuning methods are explained briefly. Subsequently a short explanation and history of the concept of relay autotuning is given. In the last section some concepts from system identification are listed, and the relay experiment is compared to some other common system identification methods.

2.1 PID Control

The PID controller is by far the most used controller type in industry [Desborough and Miller, 2002]. A typical control system is shown in Figure 2.1, along with some signal definitions. The PID controller calculates the control signal at time t, based on the actual control error, the integral of the error and the derivative of the error. A
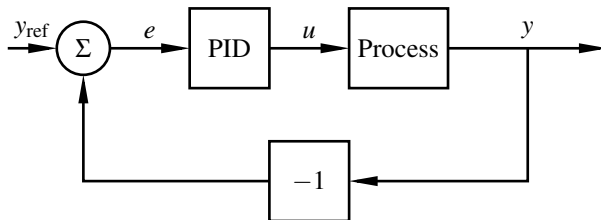

Figure 2.1 A feedback system where a PID controller controls the process output y to be close to the setpoint. The control signal is denoted u, the reference signal y_{ref}, and the control error $e = y_{\text{ref}} - y$.

basic version of the PID controller is described by

$$u(t) = K\left(e(t) + \frac{1}{T_i}\int_0^t e(\theta)d\theta + T_d\frac{de(t)}{dt}\right), \tag{2.1}$$

where the proportional gain K, the integral time T_i, and the derivative time T_d are controller parameters. The corresponding transfer function for the PID controller is

$$C(s) = K\left(1 + \frac{1}{sT_i} + sT_d\right). \tag{2.2}$$

In practice the PID controller contains more parameters, since the derivative part needs to be filtered, the integral part needs to have some anti-windup implementation, and the proportional part usually have some setpoint weighting. These parameters will, however, not be considered in this thesis, for further information about them read for example [Åström and Hägglund, 2006].

Even if the PID controller is simple, especially in the basic version, the tuning of the three controller parameters K, T_i and T_d can be a tedious task. An automatic procedure to find the controller parameters is therefore very useful.

2.2   Models

Many existing tuning rules for PID controllers rely on a model of the process. Even though processes can be of high complexity, many of them can be controlled sufficiently well by a PID controller based on a low-order model of the process. One of the most common low-order model approximations is a first order system with time delay, henceforth called the FOTD model. The FOTD model can be defined with two different parametrizations

$$P(s) = \frac{K_p}{1 + sT}e^{-sL}, \tag{2.3}$$

$$P(s) = \frac{b}{s+a}e^{-sL}. \tag{2.4}$$

Notice that the definition (2.3) can not be used to describe a process with integral action, while a pure time delay can not be represented by (2.4). Throughout this thesis the definition (2.3) will be used whenever an FOTD model is referred to.

Another common, slightly more advanced, low-order model approximation is the second order time delayed model, or SOTD model. This model is defined as

$$P(s) = \frac{K_p}{(1 + sT_1)(1 + sT_2)}e^{-sL}. \tag{2.5}$$

Since (2.3) can not be used to describe integrating processes, an integrating time delayed model, henceforth called the ITD model, will be used. It is defined as

$$P(s) = \frac{k_v}{s}e^{-sL}. \tag{2.6}$$

15

The same goes for the SOTD model in (2.5), and therefore the integrating plus first order time delayed model, IFOTD model, will also be used. It is defined as

$$P(s) = \frac{k_v}{s(1+sT)} e^{-sL}.$$ (2.7)

2.3 Normalized Time Delay

The normalized time delay, τ, for an FOTD process is defined as

$$\tau = \frac{L}{L+T}, \qquad 0 \leq \tau \leq 1.$$ (2.8)

The normalized time delay is used to characterize whether the behavior of the process is most influenced by its time delay L, or the dynamics described by its time constant T. If τ is close to 1, the time delay is much larger than the time constant, and the system is said to be *delay dominated*. If the time constant is much larger than the time delay, τ will be small and the process is said to be *lag dominated*. For intermediate values of τ, the system is said to be *balanced*. For processes that are not of the FOTD structure, the "true" normalized time delay will be denoted τ_s, and is calculated from the *apparent time constant* and the *apparent time delay*. These are achieved from the FOTD model approximation given by a step response analysis of the process.

Depending on the classification of the process, some tuning choices can be made. One is that it has been shown [Åström and Hägglund, 2006] that derivative action can be very beneficial for processes with small τ, but will only give marginal effects for $\tau \approx 1$. It is also shown that while an FOTD model is sufficient for controller tuning for processes with high τ, processes with low τ can gain a lot from more accurate modeling. Knowledge of τ is therefore essential for making choices in the autotuner procedure, something that will be discussed further in Section 4.2 and Section 4.3.

The idea of using the information from τ in a relay autotuning procedure is not new. In [Luyben, 2001], a so called *curvature factor* and its relation to the ratio L/T was calculated and used for decisions on which tuning method to use, and to find an FOTD model from the relay test. This thesis proposes a simpler method to find this information, which will be described in Section 3.2.

2.4 PID Tuning

Requirements

Typical requirements for PID control are related to load disturbance attenuation, and robustness to process variations and measurement noise. One criteria for load

disturbance attenuation is the integrated absolute error, or IAE-value, defined as

$$\text{IAE} = \int_0^\infty |e(t)|dt, \qquad (2.9)$$

for a unit step change in the load.

Robustness to process variations can be measured by the maximum sensitivities M_S and M_T, which are the largest absolute values of the sensitivity function S,

$$S(s) = \frac{1}{1+P(s)C(s)}, \qquad (2.10)$$

and the complementary sensitivity function T,

$$T(s) = \frac{P(s)C(s)}{1+P(s)C(s)}, \qquad (2.11)$$

respectively. The notation

$$M_{ST} = \max(M_S, M_T) \qquad (2.12)$$

will be used as a robustness measure in this thesis.

In addition to the requirements on IAE and M_{ST}, many other constraints could be added. For example the controlled system should be able to follow setpoint changes in a satisfactory way. This could be measured by the *rise time*, *settling time*, *overshoot* and *steady-state error*. There are also alternatives to IAE, like for example the *integral error*, IE, or the *integral squared error*, ISE. However, in this work the performance and robustness measures will be restricted to IAE and M_{ST}.

Tuning Methods

There are many methods for tuning of PID controllers, ranging from the classic rules proposed in [Ziegler and Nichols, 1942], to advanced optimization programs. Examples of existing tuning rules based on an FOTD model of the process are λ-tuning [Sell, 1995], the SIMC [Skogestad, 2003; Skogestad, 2006] and AMIGO [Åström and Hägglund, 2006]. The different tuning rules all have their benefits and drawbacks. In this work the AMIGO method and the optimization based tuning described in [Garpinger and Hägglund, 2008], where IAE is minimized with constraints on M_{ST}, are the two methods used. Modification to another tuning method is straight forward.

The AMIGO rules are described in [Åström and Hägglund, 2006]. The rules are based on an approximation of the optimization method MIGO, also described in [Åström and Hägglund, 2006], that optimizes the integral error IE with restrictions based on M_S and M_T. The AMIGO rules were derived from the same test batch that is used in this thesis, which is listed in Section B. The model approximates were obtained from step response experiments, or from a combination of step and frequency responses. The AMIGO method contains tuning rules for PI controllers

Table 2.1 The AMIGO tuning rules for PI controllers.

Model	PI parameters
FOTD	$K = \dfrac{0.15}{K_p} + \left(0.35 - \dfrac{LT}{(L+T)^2}\right)\dfrac{T}{K_p L}$ $T_i = 0.35L + \dfrac{13LT^2}{T^2 + 12LT + 7L^2}$
ITD	$K = \dfrac{0.35}{k_v L}$ $T_i = 13.4L$

based on FOTD and ITD models, as well as PID controllers based on FOTD, ITD, SOTD and IFOTD models. The PI tuning rules are listed in Table 2.1 and the PID tuning rules are listed in Table 2.2. Note that for the SOTD model and the IFOTD model, the listed controller parameters k_i and k_d, are from a different parametrization of the PID controller. This is done for practical reasons, and the conversion back to T_i and T_d is easily made by using that $T_i = K/k_i$ and $T_d = k_d/K$.

2.5   Relay Autotuning

The relay autotuner was first described in [Åström and Hägglund, 1984]. The idea is to find the critical gain and critical period used by [Ziegler and Nichols, 1942] in an automized way. By introducing a relay function in the control loop, as shown in Figure 2.2, most processes will start to oscillate. From these oscillations the critical frequency ω_c and the critical gain k_c can be retrieved and used for controller tuning. The main advantage with this method is that it is easy to use, and that no a priori information about the process is needed. The relay feedback finds the interesting frequency area automatically. In the experiment in [Åström and Hägglund, 1984] the zero-crossings and the peak amplitudes of the process output was measured. The describing function approximation (DFA) was then used to find k_c and ω_c. For an explanation of describing functions, see e.g. [Khalil, 2000]. The proposed controller tuning was based on either a specified amplitude or phase margin. A relay with hysteresis was introduced to deal with measurement noise. With hysteresis the achieved point is no longer the critical point, but instead the point where the Nyquist curve intersects the negative inverse of the describing function for the relay with hysteresis. However, for a small hysteresis this point is close to the critical point.

Table 2.2 The AMIGO tuning rules for PID controllers.

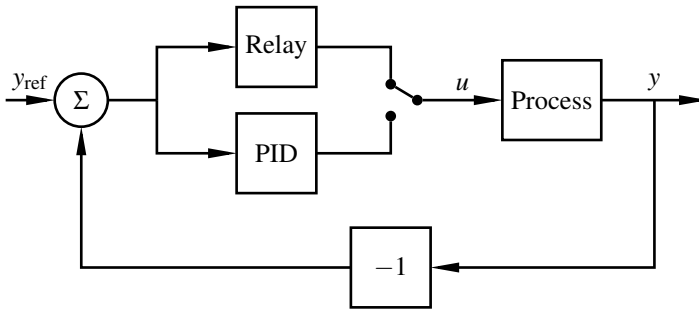| Model	PID parameters
FOTD	$$K = \frac{0.2L + 0.45T}{K_p L}$$ $$T_i = \frac{0.4L + 0.8T}{L + 0.1T} L$$ $$T_d = \frac{0.5LT}{0.3L + T}$$
ITD	$$K = \frac{0.45}{k_v L}$$ $$T_i = 8L$$ $$T_d = 0.5L$$
SOTD	$$K = \frac{0.19}{K_p} + \frac{0.37T_1 + 0.18T_2}{K_p L} + \frac{0.02T_1 T_2}{K_p L^2}$$ $$k_i = \frac{0.48}{K_p L} + \frac{0.03T_1 - 0.0007T_2}{K_p L^2} + \frac{0.0012T_1 T_2}{K_p L^3}$$ $$k_d = \frac{T_1 + T_2}{K_p(T_1 + T_2 + L)}\left(0.29L + 0.16T_1 + 0.2T_2 + \frac{0.28T_1 T_2}{L}\right)$$
IFOTD	$$K = \frac{0.37}{k_v L} + \frac{0.02T}{k_v L^2}$$ $$k_i = \frac{0.03}{k_v L^2} + \frac{0.0012T}{k_v L^3}$$ $$k_d = \frac{0.16}{k_v} + \frac{0.28T}{k_v L}$$

Figure 2.2   The setup for the relay feedback experiment. When the experiment starts, the PID controller is disconnected, and instead the process output *y* is controlled by a relay function. When the experiment is done and the PID controller parameters are tuned, the system switches back to PID control.

The relay autotuner has since its introduction been widely used in industry. Apart from that no prior information about the process is needed, some additional benefits of the relay autotuner has ensured its successful use in process industry. One advantage is the rather short experiment time. The fact that the relay experiment is performed in closed loop and does not make the process drift away from its setpoint is another advantage. This makes it a good identification method for nonlinear processes, since it stays in the linear region for which the transfer function is wanted, something emphasized in [Luyben, 1987] where the relay experiment was used as a part in finding low-order transfer functions for nonlinear distillation columns.

During the years since the original relay autotuner was proposed, many modifications and improvements of it have been suggested in literature. The most common modification is to find one of the low-order models described in Section 2.2 from the experiment. This is not done in the original autotuner since the single frequency point, given by ω_c and k_c, only allows estimation of two parameters. A thorough review of the advances in model estimation from relay feedback experiments has been presented in [Liu et al., 2013]. In the review they separate the relay experiments according to two different aspects. The first is whether a symmetric or asymmetric relay function is used. The other aspect is whether the modeling is based on the describing function approximation (DFA), a curve-fitting approach, or some frequency response estimation. The original autotuner in [Åström and Hägglund, 1984] falls into the category of a symmetric relay autotuner that uses DFA. The autotuner presented in this thesis would instead fall into the category of curve-fitting based on an asymmetric relay feedback experiment.

The use of an asymmetric relay function has the benefit of better excitation of the process, which allows estimation of the static gain from the relay experiment. The use of an asymmetric relay was first presented in [Shen et al., 1996b], where

the asymmetry was introduced in the switching conditions of the relay. In this and most later versions, the asymmetry is instead introduced in the relay amplitudes. The possibility to estimate the static gain from the relay experiment provides a way to get an FOTD model from the experiment, instead of the single point on the Nyquist curve, which was obtained in the classic version. Some attempts of finding an FOTD model from the symmetric relay experiment was done in [Luyben, 1987] where it was assumed that the static gain was either known or estimated through a separate experiment, and in [Li et al., 1991] where an extra relay experiment, with different parameters, was made to remove the need of knowing the static gain a priori. However, the extra relay experiment doubles the experiment time which is an obvious drawback. Since the asymmetric relay gives the static gain and the two other FOTD parameters from a single relay experiment, that is preferred.

The asymmetric relay autotuner in [Shen et al., 1996b] used DFA, which is not recommendable when the relay is asymmetric. This since the asymmetry deteriorates the accuracy of the obtained critical point, since the oscillation is no longer close to a sine wave. The choice of asymmetry level is therefore a trade-off between getting a good value of the critical point and getting a good estimate of the static gain. To avoid this trade-off, either the curve-fitting approach, or some improved frequency response estimation, could be used instead of the DFA. Two examples of improved frequency response estimation are presented in [Friman and Waller, 1997] and [Wang et al., 1997a]. In [Friman and Waller, 1997] multiple relays in parallel was used to find more than one frequency point on the Nyquist curve, and then fit a model to the obtained points. In [Wang et al., 1997a] the approach is instead to use a single relay, and then multiply the input and output with a decay exponential and Fourier transform it to get $G(i\omega_i)$ for some different frequencies ω_i.

The approach in this thesis is to use curve-fitting to find the model parameters from the experiment. The main reason is that it permits modeling based on clearly visible characteristic features of the oscillation. Some of them are the time period of the oscillation, the amplitudes of the oscillation, the times of the maximum amplitudes, maximum slope of the output data, and the time from the relay switch to the turning of the output signal. If noise-free simulations are performed, all of these measures are easy to obtain. Measures that are easily and robustly determined even in the presence of noise, are preferred when the autotuner is used in an industrial setting. The only data used from the relay experiment in this thesis, are the integral of the output signal during one period of oscillation, and the half-periods of the oscillation given by the relay switching times. Some alternative ways of finding low-order models from curve-fitting of asymmetric relay data are given in [Wang et al., 1997b], [Kaya and Atherton, 2001b], [Lin et al., 2004] and [Liu and Gao, 2008]. All of these methods use the half-periods and the integrated output signal as well. In addition to these measures [Wang et al., 1997b] and [Lin et al., 2004] has expressions for the output amplitudes for an FOTD model under asymmetric relay feedback. In [Liu and Gao, 2008] they also measure the time delay as the time between the relay switch and the amplitude peak. This measure is, however, quite

sensitive to noise and in the results they used an average of 10 stable cycles to obtain their values when noise was added. This gives a rather long experiment time, which is not useful in practice.

2.6 Process Identification Methods

The relay feedback experiment is not the only way to find a low-order model from experiment data. This section presents some other common strategies. All system identification methods start with the design of the input signal to the process that should be identified. Experiments could be done either in open loop or in closed loop. The relay feedback is an example of closed-loop identification. Some examples of common input signals for open-loop identification are *Filtered Gaussian White Noise*, *Pseudo-Random Binary Signals* (PRBS), or *Chirp Signals*. Details about these signal types can be found in e.g., [Ljung, 1999]. The input signal should excite the process in the frequency range where good model accuracy is required. The frequency range will depend on the use of the model. For PID control the frequencies where the process has a phase lag of $90° - 180°$ are of particular interest.

All the mentioned signal types for open-loop identification has the drawback that process information is needed, in order to design the input signals to give the desired excitation. This is, however, not a problem for the relay feedback since it will provide excitation in the interesting frequency range for PID control automatically. Another common and simple open-loop identification method is to look at a step response. Some difficulties with step-response identification are to decide the amplitude of the input step, and to determine when the process has reached its steady state. It can also be difficult to determine the wanted points and slopes from the experiment data accurately.

When the experimental data is obtained, it needs to be analyzed to find the desired model. A common way to do this is to use some parameter estimation method to obtain process models, and then apply various testing methods like estimation error, Akaike's Information Criteria, parameter variances etc, to determine a proper model structure.

The analysis of the experiment in this thesis follows the lines of traditional system identification, but is guided by the fact that we want models suitable for design of PID controllers. We therefore restrict ourselves to the model types described in Section 2.2. Two important aspects of the system identification process are discussed in further detail in the remaining parts of this section. The first is whether the excitation of the data is good enough to estimate the desired model. The other is how to decide if the obtained model is sufficiently good.

Excitation of input data

The excitation of the input data is important. The excitation needs to be in the right frequency range, and it also needs to be exciting enough to permit estimation of
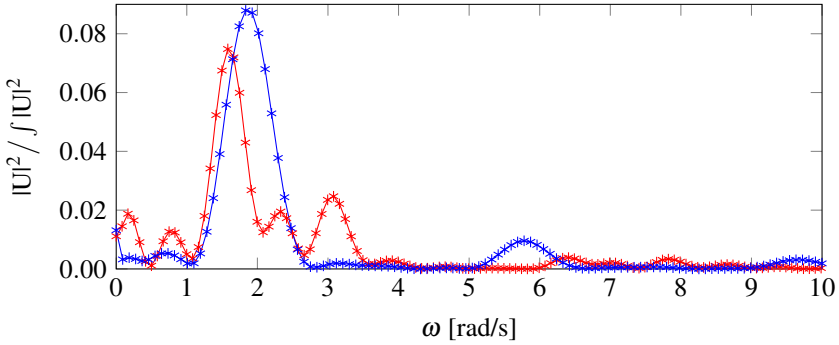
Figure 2.3   Frequency spectra for two relay experiments, performed by the proposed autotuner, on $P(s) = \dfrac{1}{s+1}e^{-sL}$. The notation U is used for the Fourier transform of u. The blue line shows the spectrum for a symmetric relay experiment. The red line shows the spectrum for an asymmetric relay experiment, where one of the relay amplitudes were five times the size of the other.

the desired number of model parameters. The signal is *persistently exciting of order n* if a model with n parameters can be reliably determined from the data. To find out how many parameters that can be estimated, the singular values of the input covariance matrix are considered. The number of singular values above a certain threshold gives the number of parameters that can be estimated. For more details, see [Ljung, 1999]. Some examples are that white noise is persistently exciting of any order, a step input is persistently exciting of order 1, a sinusoid input is persistently exciting of order 2, and a PRBS input is persistently exciting of order M, where M is the period of the PRBS.

For a symmetric relay experiment the excitation is considered close to a sinusoidal, which gives that approximately two parameters can be estimated. For the asymmetric relay the excitation can be interpreted as two different sinusoids plus a step which would give that approximately five parameters could be estimated from the experiment data. An example of the frequency content for the symmetric and an asymmetric relay, where one of the relay amplitudes are five times higher than the other amplitude, is shown in Figure 2.3. The process simulated in this figure is

$$P(s) = \frac{1}{s+1}e^{-s} \tag{2.13}$$

which has the critical frequency $\omega_c \approx 2$ rad/s. The figure shows that both the relays have most of their frequency content around ω_c. However, the asymmetric relay has a more spread-out frequency content.

Model evaluation

To evaluate whether the obtained model is close to the real process or not is a difficult issue. A number of different measurements can be used to compare the models. One common way is to compare the step response for the model with the one for the real process. This is simple, but can also be misleading since there are processes with very similar open-loop step responses that differ significantly when the loop is closed and vice versa, see e.g., [Åström and Murray, 2008]. In Section 3.4 a parameter estimation method used in this thesis is described. For that method, the cost function is obtained by comparing the measured process output with the model output fed with the same input signal. This comparison is, as well as the step response, made in open-loop. So are comparisons between the Bode diagrams and Nyquist diagrams for the model and true process. All these comparison methods can be interesting, but since the aim of the autotuner is to get a good controller for the process, the performance in closed loop is more interesting than similarity in open loop. One way to compare two models P_1 and P_2 in closed loop, is given by the *Vinnicombe metric* or *ν-gap metric* [Vinnicombe, 2001], which for scalar systems is defined as

$$\delta_\nu(P_1, P_2) = \left\| \frac{P_1 - P_2}{(1 + P_1)(1 + P_2)} \right\|_\infty.$$ (2.14)

This measure can be interpreted as the largest difference between the closed loop systems obtained by unit feedback for the two processes. It is, however, not restricted to unit feedback, but ensures that any controller that is good for P_1 is also good for P_2 if the metric δ_ν is small [Vinnicombe, 2001].

To be more specific, consider the stability margin $0 \leq b_{P,C} \leq 1$, defined by Vinnicombe as

$$b_{P,C} = \left\| \begin{array}{cc} \dfrac{PC}{1+PC} & \dfrac{P}{1+PC} \\[2mm] \dfrac{C}{1+PC} & \dfrac{1}{1+PC} \end{array} \right\|_\infty^{-1}.$$ (2.15)

A controller designed for P_1 will decrease this stability margin with at most δ_ν, as described by

$$b_{P_2,C} \geq b_{P_1,C} - \delta_\nu(P_1, P_2),$$ (2.16)

when applied to the process P_2. This property makes the Vinnicombe metric a good measure for the autotuner, since what is interesting is not the estimated model itself, but rather that the controller obtained from the model gives satisfactorily results when controlling the true process.

3

Asymmetric Relay Feedback

In this chapter the asymmetric relay function is described. It is also explained how the normalized time delay, as well as low-order models, can be found from the asymmetric relay feedback experiment.

3.1 Definitions

It is assumed that the system is at equilibrium at the working point (u_0, y_0) before the relay experiment is started. The asymmetric relay function used in this thesis is

$$u(t) = \begin{cases} u_{\text{on}}, & y(t) < y_0 - h, \\ u_{\text{on}}, & y(t) < y_0 + h, \quad u(t^-) = u_{\text{on}}, \\ u_{\text{off}}, & y(t) > y_0 - h, \quad u(t^-) = u_{\text{off}}, \\ u_{\text{off}}, & y(t) > y_0 + h, \end{cases} \tag{3.1}$$

where h is the hysteresis of the relay and $u(t^-)$ is the value u had the moment before time t. The output signals of the relay, u_{on} and u_{off}, are defined as

$$u_{\text{on}} = u_0 + \text{sign}(K_p)d_1, \tag{3.2}$$

$$u_{\text{off}} = u_0 - \text{sign}(K_p)d_2. \tag{3.3}$$

The sign of the process gain K_p (or k_v if the process is integrating) may be determined during the startup of the experiment, as will be described in Section 5.2.

The name *asymmetric* relay reflects that the amplitudes d_1 and d_2 are not equal. This creates the asymmetric oscillations. Whether d_1 is larger than d_2 or vice versa depends on if it is desired to have the large step down or up, and if it is the deviation of the control output or the process output that is most restricted.

The asymmetry level of the relay is denoted γ and defined as

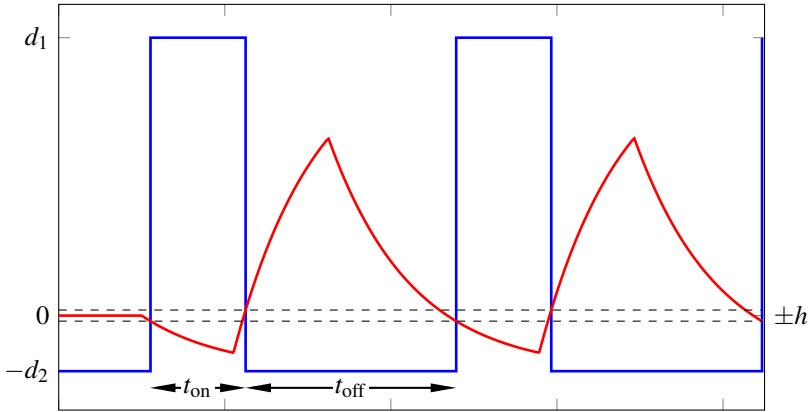$$\gamma = \frac{\max(d_1, d_2)}{\min(d_1, d_2)} > 1. \tag{3.4}$$

Figure 3.1 An example of the signals from the asymmetric relay feedback experiment. The relay output u is shown in blue, the process output y is shown in red. The black dashed lines show the hysteresis levels, $\pm h$. The experiment is started when the system is in equilibrium at the point (u_0, y_0), which in the figure is only denoted with a zero. The asymmetric oscillations is due to the different relay amplitudes d_1 and d_2. The time intervals t_{on} and t_{off} illustrate when the relay output has been u_{on} and u_{off} respectively. The relay output switches between u_{on} and u_{off} every time the process output leaves the hysteresis band.

An illustrative example of the inputs and outputs of the asymmetric relay feedback, when the static gain of the process is positive, is shown in Figure 3.1. The half-periods t_{on} and t_{off} are defined as the time intervals where $u(t) = u_{\text{on}}$ and $u(t) = u_{\text{off}}$ respectively.

3.2   Estimating the Normalized Time Delay

The normalized time delay, $\tau \in [0, 1]$, is an important parameter when tuning PID controllers, as discussed in Section 2.3. A method to rapidly determine the normalized time delay is therefore of significant value, since it provides information on how to continue the autotuning procedure.

It turns out that asymmetric relay feedback offers an effective way of estimating τ. This is due to the fact that the half-period ratio ρ, defined as

$$\rho = \frac{\max(t_{\text{on}}, t_{\text{off}})}{\min(t_{\text{on}}, t_{\text{off}})}, \tag{3.5}$$

is related to the normalized time delay of the process. If the system is lag dominated, i.e., if τ is small, the time intervals will be more or less symmetrical even though the amplitudes are asymmetric. When the process is delay dominated, τ close to 1, the half-period ratio instead reflects the asymmetry of the amplitudes.
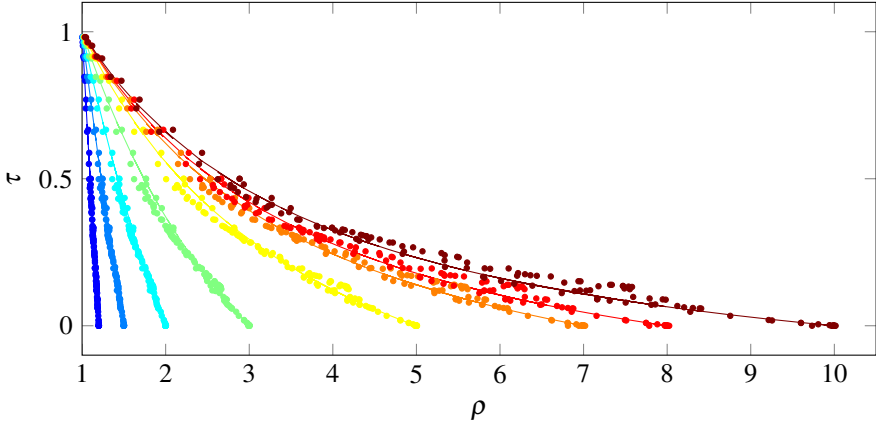
Figure 3.2 Validation results of the equation for τ, stated in (3.6). The figure shows the results for $\gamma = [1.2, 1.5, 2, 3, 5, 7, 8, 10]$ in different colors. The solid lines show the τ-values calculated from (3.6), while the dots show the relation between ρ and τ_s for the processes in the test batch (Section B).

For FOTD processes under asymmetric relay feedback with no hysteresis, this follows from (A.21) and (A.27), where the half-periods and their ratio have been derived in the limits $\tau = 0$ and $\tau = 1$ respectively. Results that are only valid for FOTD processes with a relay without hysteresis are of limited practical use. However, the observation above is valid for a wide range of process types. Figure 3.2 shows the simulation results for a test batch consisting of 134 different processes typical for the process industry. The test batch is taken from [Åström and Hägglund, 2006] and is listed in Section B. From the simulation data, an expression for τ, as a function of the asymmetry level γ and the ratio ρ, was fitted under the constraints that the endpoints should be $\tau(\rho = 1, \gamma) = 1$ and $\tau(\rho = \gamma, \gamma) = 0$, according to the derived limits. The result is the following equation for the normalized time delay

$$\tau(\rho, \gamma) = \frac{\gamma - \rho}{(\gamma - 1)(0.35\rho + 0.65)}. \tag{3.6}$$

The equation was validated against the test batch, for some different asymmetry levels γ, and the results are shown in Figure 3.2.

The errors in determining τ using (3.6) are shown in Figure 3.3 for $\gamma = 2$. The errors for some other values of γ are shown and discussed in Section 5.1. For all processes in the batch, the estimate stays within 0.08 of the correct value, and the median error is about 0.02. The obtained results are accurate enough to use the estimated τ for classifying the process and decide on what, if any, additional steps are required by the autotuner algorithm.
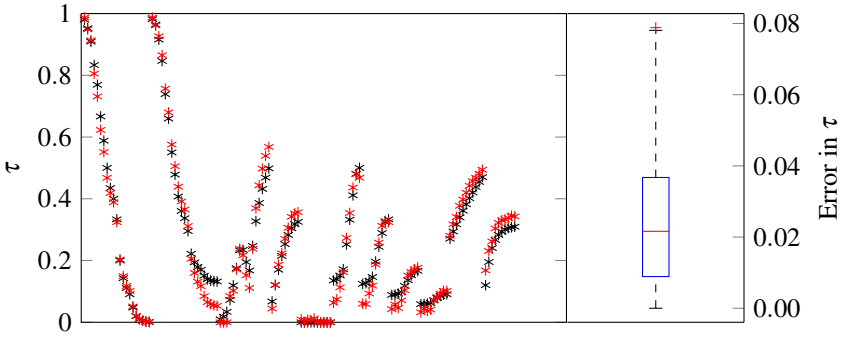
Figure 3.3   Results of the τ-estimation for the processes in the test batch. The left plot shows the estimated τ in red, and the true values τ_s in black. The right plot shows a boxplot of the absolute errors between τ and τ_s. Here, the central mark is the median, the edges of the box are the 25th and 75th percentiles, the whiskers extend to the most extreme data points the algorithm does not consider to be outliers, and the outliers are plotted individually.

3.3   Modeling

Different methods can be used to find model parameters from the experiment data, some examples were given in Section 2.5. We have focused on finding simple, intuitive equations that use measurements that are robust to noisy data. We have found equations where the only measurements needed are the time intervals t_{on} and t_{off} and the integral of the process output I_y defined as

$$I_y = \int_{t_p} \left(y(t) - y_0\right) dt \qquad (3.7)$$

where $t_p = t_{\mathrm{on}} + t_{\mathrm{off}}$ is the period time of the oscillation and y_0 is the stationary operation point we started the experiment from. All these parameters are easy to measure from the experiment data, and they show small sensitivity to noise. In addition to these values, the equations also contain the relay amplitudes d_1 and d_2, the hysteresis h, the normalized time delay τ which is derived in Section 3.2, and the integral of the relay output I_u, which analogously to I_y is defined as

$$I_u = \int_{t_p} \left(u(t) - u_0\right) dt. \qquad (3.8)$$

This integral, however, does not need to be measured from the experiment since it is given by

$$I_u = u_{\mathrm{on}} t_{\mathrm{on}} + u_{\mathrm{off}} t_{\mathrm{off}}. \qquad (3.9)$$

FOTD Models

The FOTD model defined in (2.3) has three parameters: K_p, T and L. One benefit of using the asymmetric relay, is the possibility to calculate the static gain, K_p, from

$$K_p = \frac{I_y}{I_u}. \tag{3.10}$$

Note that this does not apply to the symmetric relay, where I_u would always be zero. It follows from (3.9) that I_u can become zero with the asymmetric relay as well, but only if $t_{\text{off}}/t_{\text{on}} = -u_{\text{on}}/u_{\text{off}}$. As is shown in Section A.2, this implies that the process is integrating, and for those processes we will instead use the ITD model.

To find T and L we use the equations for t_{on} and t_{off}

$$t_{\text{on}} = T \ln \left(\frac{h/|K_p| - d_2 + e^{L/T}(d_1 + d_2)}{d_1 - h/|K_p|} \right) \tag{3.11}$$

$$t_{\text{off}} = T \ln \left(\frac{h/|K_p| - d_1 + e^{L/T}(d_1 + d_2)}{d_2 - h/|K_p|} \right) \tag{3.12}$$

that are derived in Section A.1. Since K_p can be found from (3.10), the results in (3.11) and (3.12) give two equations for the two unknown process parameters T and L. However, these equations can not be solved analytically for T and L. They can be solved numerically, but that requires proper initial guesses. Our approach is instead to find the normalized time delay τ as in Section 3.2, and use its definition (2.8) to solve the equations. Rewriting (2.8) gives the following expression for the ratio between L and T

$$L/T = \frac{\tau}{1 - \tau}. \tag{3.13}$$

Knowing this ratio, T can be found from either of the two equations (3.11) or (3.12), or from an average of both. If (3.11) is used, T is given by

$$T = \frac{t_{\text{on}}}{\ln \left(\dfrac{h/|K_p| - d_2 + e^{L/T}(d_1 + d_2)}{d_1 - h/|K_p|} \right)}. \tag{3.14}$$

With T known from (3.14) it is straightforward to obtain L from (3.13)

$$L = T \frac{\tau}{1 - \tau}. \tag{3.15}$$

In conclusion, by measuring t_{on}, t_{off} and I_y from the relay experiment and then use (3.6) to find τ, the parameters of the FOTD model (K_p, T, L) can be found from (3.10), (3.14) and (3.15).
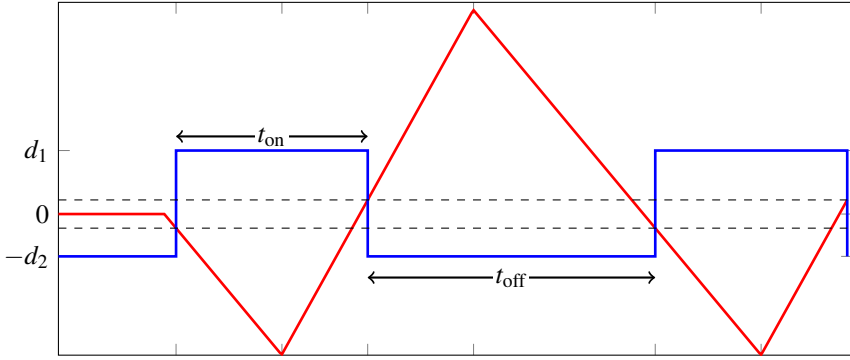
Figure 3.4   An example of the signals from a relay experiment with an ITD process. The blue line shows the relay output u, the red line shows the process output y. The dashed black lines show the hysteresis. The time intervals t_{on} and t_{off} are denoted in the figure and correspond to the times that the relay output has been u_{on} and u_{off} respectively. The relay amplitudes d_1 and d_2 are also shown in the figure. Note the triangular shape of the process output y that is characteristic for an ITD process.

ITD Models

An integrating process on the form

$$P(s) = \frac{k_v}{s} e^{-sL} \qquad (3.16)$$

can be written as the differential equation

$$\dot{y}(t) = k_v u(t - L). \qquad (3.17)$$

Since $u(t)$ is piecewise constant, so is $\dot{y}(t)$, and hence the shape of y will be triangular, see Figure 3.4. By considering the output curves, equations for k_v and L can be obtained, see Section A.2 for full derivation. The equations are

$$k_v = \frac{2I_y}{t_{on}t_{off}(u_{on} + u_{off})} + \frac{2h}{u_{on}t_{on}}, \qquad (3.18)$$

$$L = \frac{u_{on}t_{on} - 2h/k_v}{u_{on} - u_{off}}. \qquad (3.19)$$

The only measurements needed from the experiment are t_{on}, t_{off} and the integral of the process output y.

3.4   Improved Modeling by System Identification

If the FOTD and ITD models are not considered sufficient to describe the process, the parameters of a higher order model can be estimated from the experiment data. Let (u_m, y_m) be the input output data obtained from a relay experiment of length t_m, and let $P(s)$ be the transfer function of the process model with parameters p. The output generated by $P(s)$ with the input u_m is denoted $\hat{y}$. Denote the error between the generated output and the experiment output $e(t) = \hat{y}(t) - y_m(t)$. The parameters p can then be obtained by minimizing the quadratic loss function

$$J(p) = \frac{1}{2} \int_0^{t_m} e(t)^2 dt. \tag{3.20}$$

The optimization can be performed by computing the gradient J_p and the Hessian J_{pp} given by

$$J_p = \int_0^{t_m} \hat{y}_p(t) e(t) dt, \tag{3.21}$$

$$J_{pp} = \int_0^{t_m} \hat{y}_p(t) \hat{y}_p^T(t) dt + \int_0^{t_m} \hat{y}_{pp}(t) e(t) dt. \tag{3.22}$$

A good approximation of the Hessian is obtained by dropping the second term in (3.22). Newton's method can then be used to obtain the parameters that minimize the cost function $J(p)$.

Using this method, the SOTD model from (2.5) and the IFOTD model in (2.7) can be estimated. The initial parameters required by Newton's method can for these models be obtained from the relay experiment. For the SOTD models, the initial parameters used are

$$\begin{aligned} K_p^* &= K_p, \\ T_1^* &= T/1.86, \\ T_2^* &= T/1.86, \\ L^* &= \max(0, L - 0.28T), \end{aligned} \tag{3.23}$$

where K_p, T and L are the FOTD parameters obtained in the relay experiment. These initial values are based on the comparison between systems with poles of different multiplicity on pages 29–31 in [Åström and Hägglund, 2006]. For the IFOTD model the initial parameters used are

$$\begin{aligned} k_v^* &= k_v, \\ T^* &= 2L, \\ L^* &= 0, \end{aligned} \tag{3.24}$$

where k_v and L are the parameters of the ITD model found in the relay experiment. These initial parameters were found experimentally.

In this work, only SOTD and IFOTD models are estimated, but the same parameter estimation method could naturally be used for other model types as well.
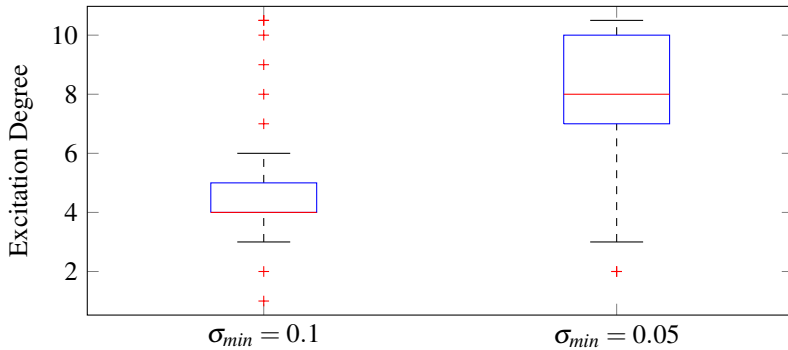
Figure 3.5 Boxplots of the degree of persistent excitation of the relay experiment data, for two different threshold values of the smallest singular value σ_{min}. The experiment data for each of the processes in the test batch is tested. All data sets with a degree of persistent excitation larger than 10 were set to have the degree 10.5 to keep focus on the interesting areas in the plots.

Looking at the excitation of the input data shown in Figure 3.5, it is clear that at least four parameters can be reliably estimated from almost all processes. There are a few data sets that are not exciting enough to do this reliably, but for most of the processes, more parameters could be estimated. Further investigation has shown that if the threshold value of σ_{min} was decreased to 0.01, all the data sets were considered persistently exciting of at least order 9. It is, however, doubtful if that many parameters could reliably be estimated in practice. Note that having data that is exciting enough for the number of parameters in the desired model is not all that is needed to find higher order models. It is also necessary to find the initial parameters for Newton's method, which is usually not straightforward.

4

Autotuner Procedure

The purpose of the autotuner is to give satisfactory controller parameters for a completely unknown process. To do this, the autotuner has to go through the different steps shown in Figure 4.1, where each step contains actions and decisions to be performed.

The first step is the *Experiment*, where it has to be decided what type of experiment should be done, and how it should be designed. It also has to be decided what experiment parameters should be used, and what data should be extracted from the experiment. In this work the experiment is the asymmetric relay feedback experiment, with steps described further in Section 4.1.

The *Model* step includes decisions on what model structure to use. It should also contain a method to obtain the desired model parameters. In this work, the estimated model structure depends on the value of the normalized time delay τ and is discussed further in Section 4.2.

Next, a *Controller* should be designed. This step contains decisions about what controller type should be used and how to find its parameters. These choices are described and discussed in Section 4.3.

The final step is the *Evaluation* of the results. Here it is decided if the performance of the obtained controller is satisfactory, or if something should be changed or remade in the previous steps. This is mainly a task for the operator. Some discussion about the evaluation step is made in Section 4.4.

This chapter shows that the autotuner procedure contains a number of different sequences and decisions. It is therefore suitable to implement the autotuner in a sequential control language. An implementation in the sequential control language Grafchart was presented in [Theorin and Berner, 2015]. The implementation clearly illustrates the different sequences by connected steps and transitions, and the obtained controller results were satisfactory.
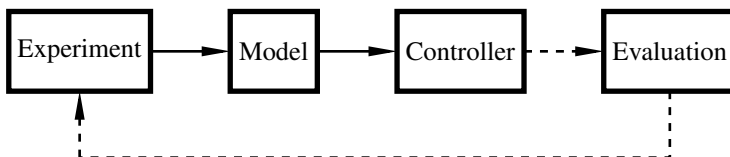
Figure 4.1 Steps to be designed and performed in an automatic tuning procedure. The dashed lines show the steps that involves the operator.

4.1 Relay Feedback Experiment

The relay feedback experiment uses the asymmetric relay function described in Section 3.1. The experiment starts when the system is at steady-state. This has to be ensured by the operator before pressing the start button on the autotuner. The different sequences of the relay experiment are described in Figure 4.2. In the initialization step most parameters are set. The default values used for the parameters are listed in Section C. The initialization step also sets up buffers to store the experiment data.

Next, the noise is measured for a specified amount of time. This step is described further in Section 5.1. When the noise level is known, an appropriate hysteresis level is set. If the noise level is high, either some parameters or deviation limits may need to be changed, and the operator will be warned about this. If the signal is too noisy, the operator is advised to filter the noise before performing the experiment.

When the noise level is measured, the relay feedback phase starts by ramping up the relay amplitudes. From the ramp-up, starting values of the amplitudes are obtained, as is the sign of the process gain. The ramp-up procedure is described further in Section 5.2.

Subsequently comes the actual experiment. Here the oscillations from the relay feedback are created. In each sample, the control signal is set according to (3.1). If the relay switches, logic for checking and updating the amplitudes are performed, see Section 5.2, and it is also checked whether the oscillations have converged to its limit cycle or not (Section 5.1). These actions are repeated in each sample until the experiment satisfies the convergence limit ε, described in Section 5.1. Then, the data needed from the experiment are retrieved and the autotuner moves on to the next step, to estimate a model of the process.

4.2 Model Design

As stated previously, the aim with this autotuner is to get a low-order model describing the process. Different model types of interest were listed in Section 2.2. The choice of model structure is in this autotuner based on the normalized time delay, τ. If τ is small, the process can be considered to be an integrating process, which implies that an ITD or IFOTD model should be estimated. In Figure 4.3 the performance and robustness for controllers based on an FOTD model of the process
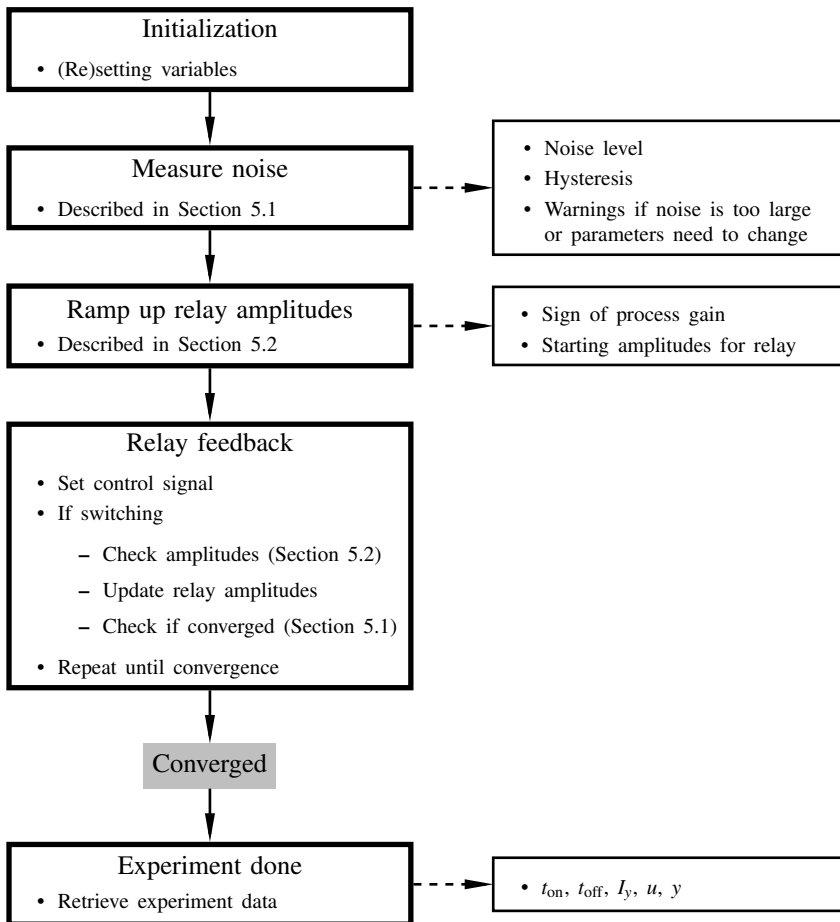
Figure 4.2 The different sequences of the relay feedback experiment. Also shown are the variables and parameters obtained in that sequence.
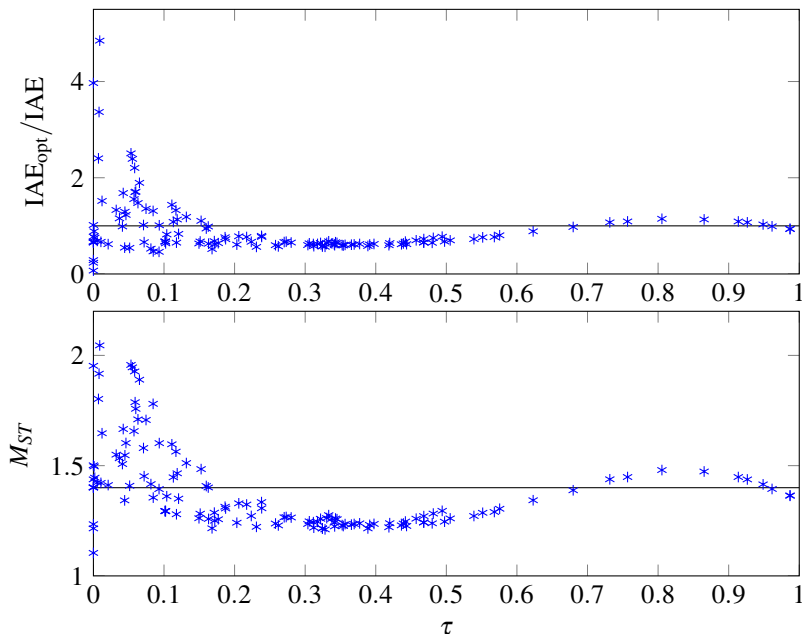
Figure 4.3 Comparison of performance and robustness of PI controllers obtained from the autotuner, and optimal PI controllers, for all processes in the test batch (Section B). The controllers from the autotuner used the AMIGO rule on the ITD or FOTD models obtained in the relay experiment. The optimal controllers were obtained by minimizing IAE with the constraint that $M_{ST} \leq 1.4$. The upper plot shows the ratio between the IAE values for the two different controllers. Equal performance is shown by a black line at the ratio 1. The lower plot shows the obtained robustness for the estimated controller, with the level $M_{ST} = 1.4$ shown as a black line.

are shown. The figure shows that for high values of τ, the results from the controllers based on an FOTD model, are close to the optimal ones. This implies that for processes with large τ, FOTD models are adequate, and hence that is what is estimated. If τ is smaller, higher-order models can give significantly better results, therefore we may consider estimating an SOTD model for these processes. It should be noted that the AMIGO rules used in this figure are not derived to minimize IAE, thus another tuning rule could improve the results. The trends for high and low values of τ would, however, be the same. The decision path of the model design is shown in Figure 4.4. The limits α and β can be varied a little, but in this thesis the values used have been $\alpha = 0.1$ and $\beta = 0.6$.

The FOTD model and ITD model can be obtained from the equations in Section 3.3. If an IFOTD model or SOTD model is wanted, the parameter estimation method described in Section 3.4 is used. Either the ITD or the FOTD model are then
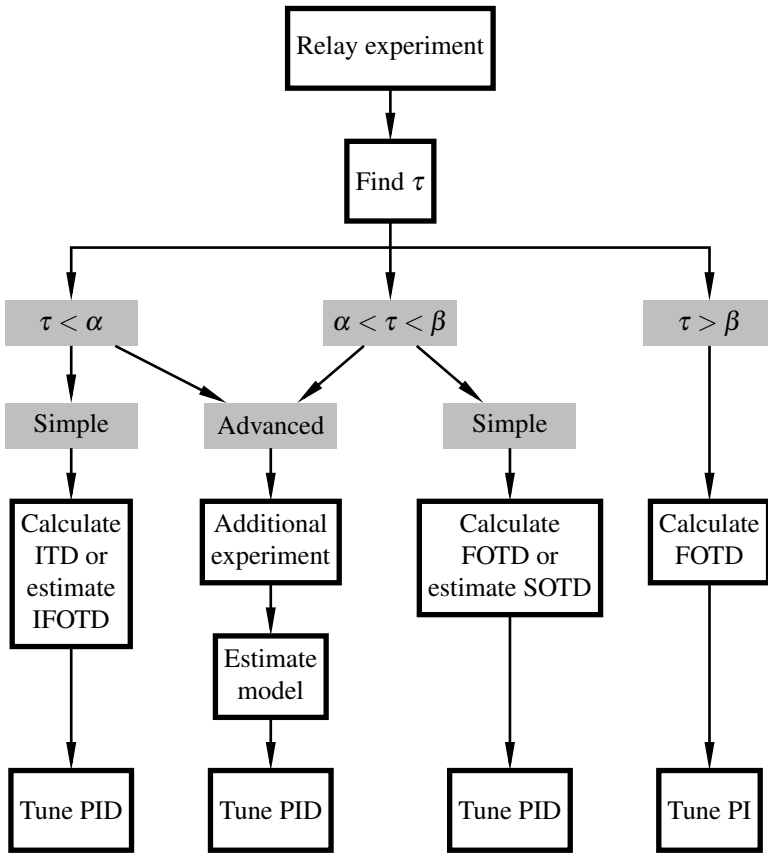
Figure 4.4 The proposed autotuner procedure.

used to get initial parameters for the algorithm. If it is crucial that we get a really good model we might consider estimating even higher order models. However, that implies that we may need better excitation and also another way to get good enough initial parameters. This is illustrated in the advanced branch in Figure 4.4. Some possible choices of additional experiments are the ones listed in Section 2.6. Note that the information from the relay experiment already performed, can be used to design the additional experiment.

4.3 Controller Design

The aim of the proposed autotuner is to find good controller parameters for the PID controller described in Section 2.1. Other controller structures are outside the scope
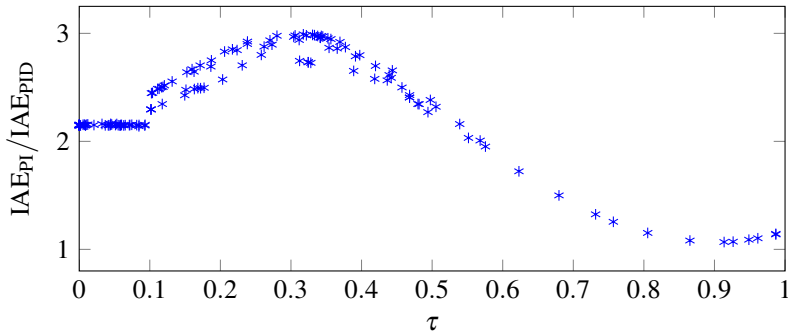
Figure 4.5 A comparison of the performance of PID and PI controllers, plotted versus the estimated normalized time delay τ. The controllers are obtained by using the AMIGO rules for the process models obtained from relay experiments on the processes in the test batch in Section B.

of this thesis. The low-order models in Section 2.2 were chosen since they all have simple existing tuning rules as the ones listed in Section 2.4. In the simple version of the autotuner we will therefore use one of them, namely the AMIGO tuning rule. This is not an obvious choice and one of the other methods could just as well have been used. In fact, what really should be done is to find a new set of simple tuning rules adjusted for the models obtained from this specific relay experiment. This is important since all tuning rules are connected to a modeling procedure, and models obtained differently may not give as good controllers as the models for which the tuning rule was derived. Since this is not done yet, this version of the autotuner will stick to the AMIGO rules. If the advanced branch is used to find higher-order models, AMIGO rules are no longer available, and instead some kind of optimization tool would be needed to find the controller parameters.

In [Åström and Hägglund, 2006], it was shown that the derivative part of the controller was beneficial for small values of τ, but not so much if τ is large. Figure 4.5 compares the performance of PI and PID controllers. The comparison is done for the test batch in Section B and shows the expected results. For large values of τ almost nothing is gained by introducing the derivative part, while for smaller τ the performance is 2-3 times higher. Notable is that the AMIGO rules, by its design, give the same benefit of the derivative part for all ITD models. Other tuning rules would give a different appearance of the curve in the low τ region. In Figure 4.4, the limit for when the derivative part of the PID controller should be used is set to $\tau < \beta$. The choice of $\beta \approx 0.6$ seems reasonable, this is the same value of β as in the model design.

4.4 Evaluation

When the autotuner has finished its experiment and obtained its final controller parameters, an evaluation of whether or not the result is satisfactory has to be performed. In this thesis the results from the autotuner are evaluated in a number of different ways. The accuracy of the model parameters obtained for the test batch, is checked for different choices of the experiment parameters in Section 5.1. The accuracy of the obtained models, as well as the robustness and performance of the obtained controllers, are shown and discussed for three chosen processes in Chapter 6. The effects of measurement noise, load disturbances, low resolution in converters and starting of experiment before steady-state is reached, are all discussed in Section 5.3-5.6. However, these results and discussions only cover a limited number of processes and situations, and no matter how much tests we would do, the evaluation will still mainly be a task for the operator. Some questions, like

- Is the resulting controller performance good enough?

- Did something go wrong during the experiment that affected the results?

- Is a more advanced model needed?

- Is another controller structure needed?

are not answered by the autotuner. These questions, the operator will need to answer him- or herself.

5

Practical Considerations

When the autotuner is used in an industrial setting, the conditions may vary a lot from the ideal simulation environment where the development has been done. In this chapter some of the practical issues for the autotuner will be presented and discussed. The first section goes through the relay parameters. How to choose the parameters may differ depending on prior knowledge about the process and possible disturbances. The parameter choices may also depend on whether the autotuner is to be used in a practical industry application, or in a large simulation environment where the autotuning facility can also be useful. The proceeding sections in the chapter discuss how the results from the autotuner are affected by practical issues like noise, load disturbances and low resolution in converters.

5.1 Parameter Choices

The relay experiment contains a lot of parameters that has to be chosen. Default values for all parameters are listed in Section C, and some of the parameter choices are explained and discussed in further detail in this section.

Noise level and hysteresis

As a first step of the autotuning procedure we measure the noise level of the signal. This is done during a specified time interval when the maximum and minimum values of the process output, y_{max} and y_{min}, are stored. The noise level, n_0, is then calculated as $n_0 = (y_{max} - y_{min})/2$. The hysteresis is then chosen to be about 2-3 times the noise level. The reference value y_0 is set during the noise measurements by taking the average of the measured y-values. If the noise level is too large the signals need to be filtered before starting the relay experiment, otherwise the output amplitudes required for the experiment will be too large. In the noise-free simulation environment the hysteresis could be chosen arbitrarily. In this thesis the hysteresis $h = 0.1$ has been used as a default value.
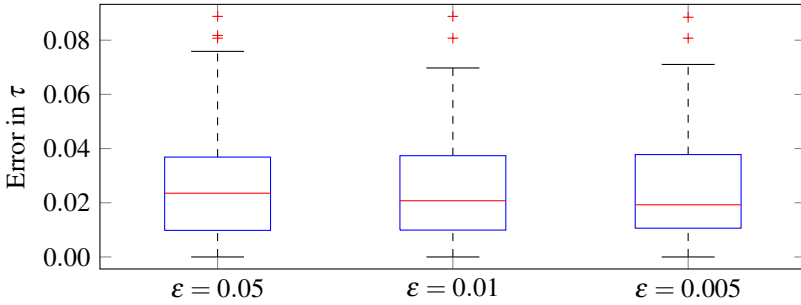
Figure 5.1 Boxplots of the absolute errors in the estimation of τ, for the three different convergence limits $\varepsilon = [0.05, 0.01, 0.005]$. On each box, the central mark is the median, the edges of the box are the 25th and 75th percentiles, the whiskers extend to the most extreme data points the algorithm does not consider to be outliers, and the outliers are plotted individually.

Convergence of limit cycles

If an FOTD system under asymmetric relay feedback has a limit cycle, it will converge to it after the first switch of the relay, see [Lin et al., 2004]. However, for other processes or if noise is added, it is not certain that the limit cycle will be reached that fast. One issue to consider in the relay experiment is therefore to decide when convergence to the limit cycle has been achieved. One method is to compare the time one period take, t_p, with the time the previous period took, t_p^*. If the difference between the period times is smaller than a certain threshold ε, i.e.,

$$\left| \frac{t_p - t_p^*}{t_p^*} \right| \leq \varepsilon \tag{5.1}$$

the system is considered to have reached the limit cycle. Another method would be to look at the oscillation amplitudes instead of the period times, but that approach was not chosen in this thesis.

To investigate the effect of ε, the processes in the test batch (Section B) was simulated with the different values $\varepsilon = [0.005, 0.01, 0.05]$. To make the situation a little more realistic, band-limited white noise with a measured noise level of $n_0 = 0.12$ was added to the process output. The resulting accuracy of τ and K_p, for the different choices of ε, are shown in Figure 5.1 and Figure 5.2. The figures show that the accuracy of τ is more or less identical for all three values, but that the estimation of K_p is improved for smaller convergence limits. Since the experiment should ideally be short, a comparison of the convergence times was performed for the entire batch. It turned out that the mean convergence time for $\varepsilon = 0.05$ was 0.31 periods shorter than for $\varepsilon = 0.1$, while the convergence time for $\varepsilon = 0.005$ was in mean 0.52 periods longer than for $\varepsilon = 0.01$. At most, $\varepsilon = 0.05$ gave a 2.5 periods shorter convergence time, while $\varepsilon = 0.005$ made one process take 9 periods more to
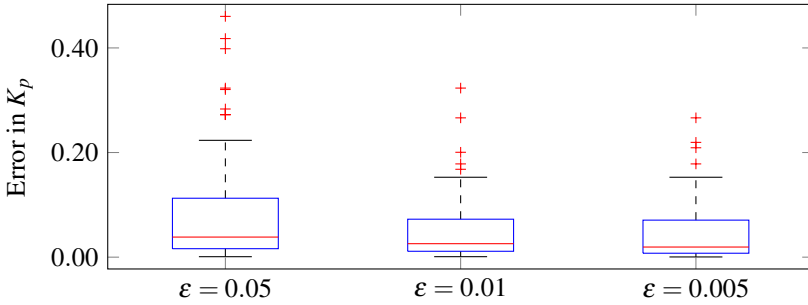
Figure 5.2 Boxplots of the absolute errors in the estimation of K_p, shown for the three different convergence limits $\varepsilon = [0.05, 0.01, 0.005]$. All processes, classified as non-integrating, from the test batch are included.

converge than it did with $\varepsilon = 0.01$. Since the accuracy was more or less the same for $\varepsilon = 0.01$ and $\varepsilon = 0.005$ there is no need to use the lower value, since that increases the experiment time. Increasing the limit to $\varepsilon = 0.05$ make the experiment a little shorter, but the obtained values of K_p are also somewhat worse. Considering the results, the default value chosen for this thesis is $\varepsilon = 0.01$.

Relay amplitudes

The question of how to choose the relay amplitudes is subject to some different aspects. It is necessary that $|K_p \min(d_1, d_2)| > h$ for the output to reach outside the hysteresis band and create oscillations. Some margin to this limit, which could be stated as

$$\min(d_1, d_2) \geq \frac{\mu h}{|K_p|} \tag{5.2}$$

where $\mu > 1$ is a constant, is required to get good results. In Figure 5.3 the accuracy of the estimated τ is shown for some different values of μ. The plot shows that the results improve a lot up to $\mu = 3$, are slightly better for $\mu = 5$ and after that stay more or less the same.

Since K_p is not known beforehand, the relay amplitudes can not be set according to the constraint (5.2) directly. Instead we consider the smallest peak deviation of the process output, y_{spd}, which is constrained to $y_{spd} \leq K_p \min(d_1, d_2)$. By putting a lower limit $y_{mindev} = \mu h$ on the peak deviation we can guarantee that (5.2) is satisfied since

$$|K_p| \min(d_1, d_2) \geq y_{spd} \geq y_{mindev} = \mu h. \tag{5.3}$$

How the lower limit is accomplished in practice is described by the amplitude adjustment in Section 5.2. Note that since there are multiple inequalities in (5.3) you may not need to put as high value of μ as in (5.2) to get good results. With the default parameters listed in Section C, the maximum error of τ is 0.08 for the test batch, and there such a small value as $y_{mindev} = 2h$ was used.
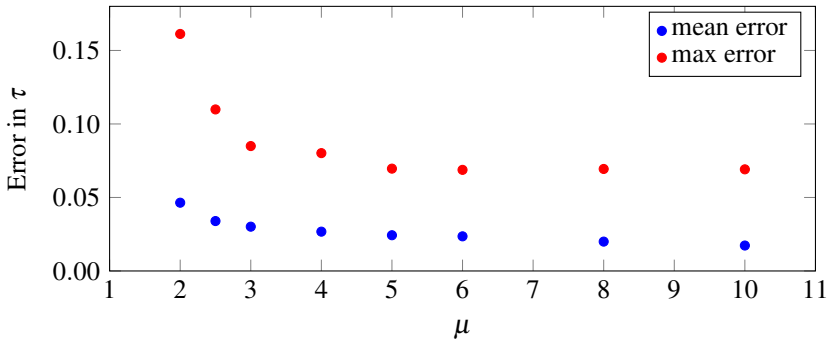
Figure 5.3 Mean and maximum errors of the τ-estimations, shown for some different values of μ. During these simulations, the relay amplitudes were fix, with the small amplitude $\min(d_1, d_2) = \mu h / |K_p|$ and the large amplitude a factor $\gamma = 2$ larger.
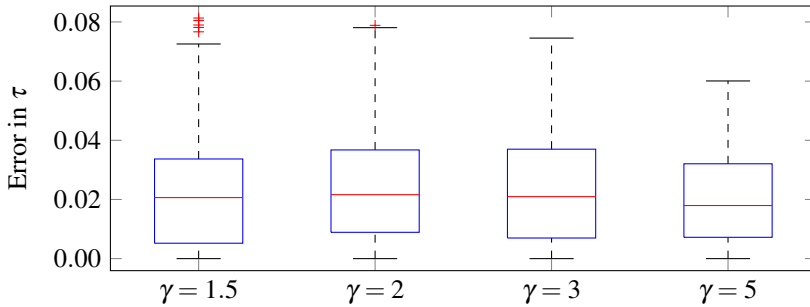

Figure 5.4 Boxplots of the absolute errors of τ, shown for different values of the asymmetry level γ.

The asymmetry level γ, i.e., the ratio between d_1 and d_2 is also something to consider. Figure 5.4 and Figure 5.5 show the results of the estimates of τ and K_p for different values of γ. For the estimates of τ the results from the entire test batch is plotted. For the estimates of K_p only the processes that were estimated as FOTD processes, and hence have a value of K_p, are shown. The results indicate that as high asymmetry as possible should be chosen to get good estimates of K_p, but that the estimates of τ does not depend that much on γ.

By forcing y_{spd} to be large, and using a high γ, the results shown are more accurate. If the autotuner is to be used in simulation environments you could therefore use high values. However, if you are to use the autotuner on real processes in an industrial setup you will have upper constraints as well. You will have limitations on the deviations of both the process output, y_{maxdev}, and the control signal, u_{maxdev}. These constraints may force you to use lower values of γ and/or y_{mindev} than you
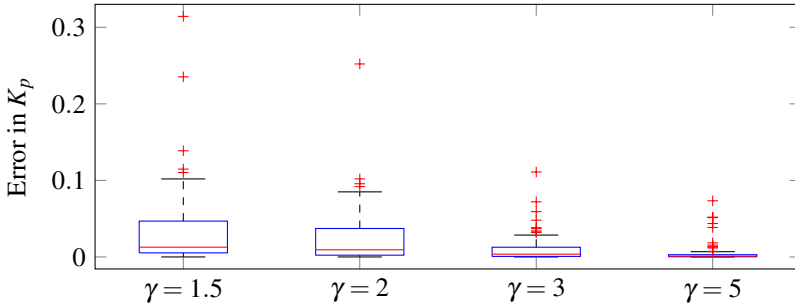
Figure 5.5 Boxplots of the absolute errors of K_p, shown for different values of the asymmetry level γ. All processes, classified as non-integrating, from the test batch are included.

would prefer to. How the upper limits enter into the process of choosing the relay amplitudes is explained in Section 5.2.

Sampling times

The question of how to choose the sampling time for the relay experiment is not easily answered, since you have no information about the process when designing the experiment. In the simulations in this thesis a fixed sample time of $t_s = 0.01$ s has been used as default value. The test batch in Section B consists of processes with very different time constants, so for some processes that sampling time is unnecessarily small, while for others it is too large. An indication of this can be seen by looking at the results of K_p and τ for the two different sampling times, $t_s = 0.01$ s and $t_s = 0.001$ s, shown in Figure 5.6 and Figure 5.7.

The results for τ in Figure 5.6 are more or less the same, so estimation of τ seems to be quite insensitive to the choice of sampling time. Looking at the results for K_p in Figure 5.7 two things can be noticed. The first observation is that the estimation of K_p deteriorates with lower values of τ. This could be explained by the fact that the integral of the control signal given in (3.9) goes towards zero as τ decreases, and a small difference in its measurement gives a greater impact on the result. The other observation is that the worst estimates are much improved when the sampling time is decreased. The processes that get a bad value of K_p with the default parameters are all very fast processes, with time periods in the order of 1s, that behaves much better when the sampling time is reduced to 0.001 s. For most of the processes, however, the result is more or less the same, since the default sampling time was sufficient. For those processes the main difference is that the time it takes to simulate is much longer and the amount of data storage needed is increased a factor 10. One of the very slow processes (number 42 in the test batch) could not be simulated with $t_s = 0.001$ s since it would not have time to converge before the data storage ran out of space. That process is therefore not included in
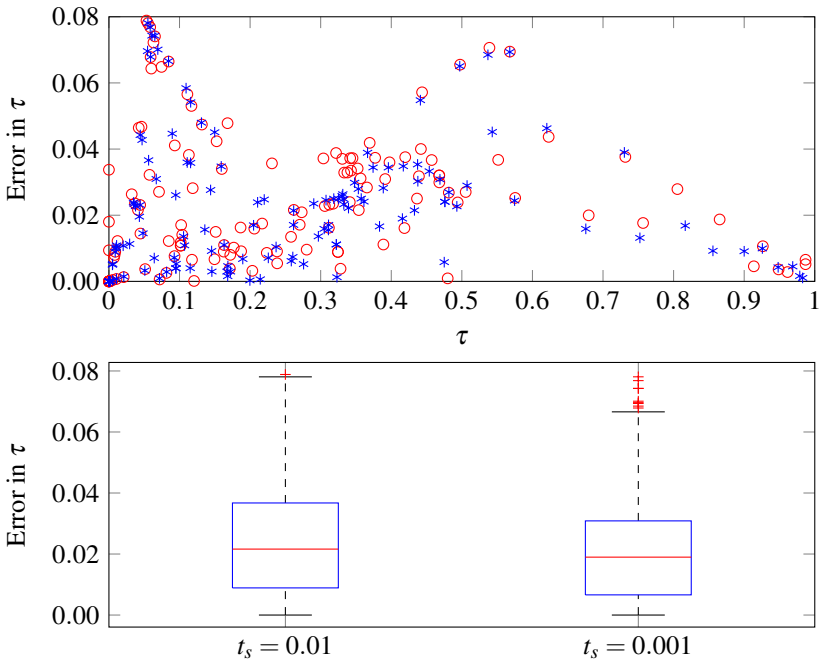
Figure 5.6   Estimates of τ for two different sampling times. The upper plot shows the absolute error between the estimated τ, and the true value τ_s, as a function of the estimated τ. The red circles show the results for $t_s = 0.01$ s, and the blue stars show the results for $t_s = 0.001$ s. The lower plot shows the corresponding boxplots of the absolute errors.

the figure for $t_s = 0.001$ s. For this reason a shorter sampling time is not always preferred.

One way to solve the problem of a way too large or too small sampling time is to adjust the sampling time after the first half-period when it is known approximately the speed of the process.

5.2   Startup and Amplitude Adjustments

Since the process gain may not be known in advance, a strategy to find adequate relay amplitudes has to be implemented in the autotuner. The startup procedure in this thesis is inspired by earlier versions of industrial autotuners. The control signal is increased exponentially during a couple of seconds until one of two things happen. Either the control signal reaches its maximum allowed value, u_{maxdev}, or the process output reaches the hysteresis limit. When the hysteresis limit is reached, the
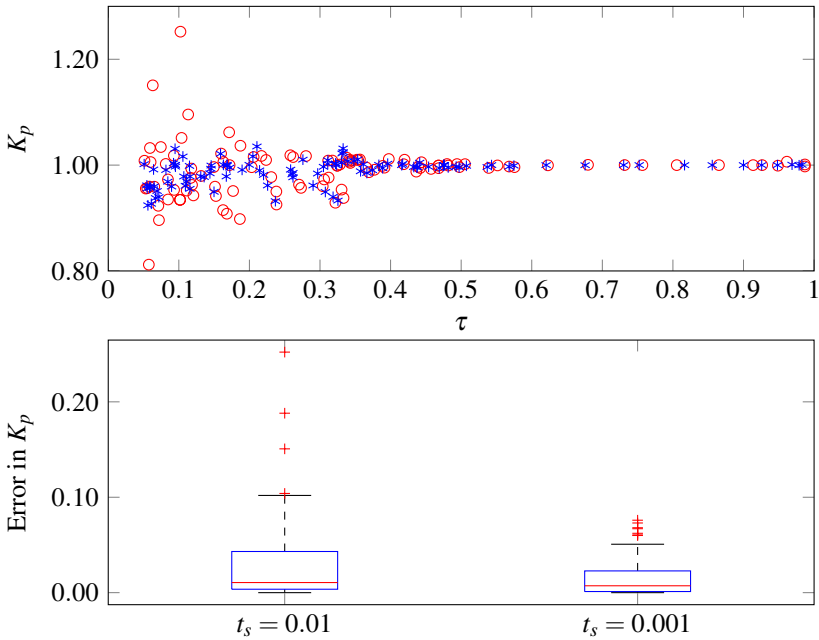
Figure 5.7 Estimates of the process gain K_p for two different sampling times. The upper plot shows the estimates as a function of the estimated τ. The red circles show the results for $t_s = 0.01$ s and the blue stars show the results for $t_s = 0.001$ s. The lower plot shows boxplots of the absolute error of K_p. In both the upper and lower plot the only processes shown are the ones with an estimated $\tau > 0.05$.

sign of the process gain is determined based on which limit that is broken, and the initial relay amplitudes are set according to the current level of the control signal. In the case where the control signal reaches its maximum value it stays at that level until the process output reaches the hysteresis limit. Then the sign of the process gain and the relay amplitudes are set as in the first case. If either the sign of the process gain, or its approximate amplitude, is known in advance this information can be used to set the initial relay amplitudes.

During the experiment the relay amplitudes are adjusted to get the oscillation in the desired amplitude interval. The lower limit on the small peak deviation, y_{spd}, was explained and motivated in Section 5.1. The upper limit y_{maxdev} on the large peak deviation, y_{lpd}, can either be a default value or specified by the user. The desired amplitude interval is shown in Figure 5.8. The amplitude adjustment is described in Algorithm 1, but has some additional logic to make sure that the relay amplitudes never reach above u_{maxdev}.
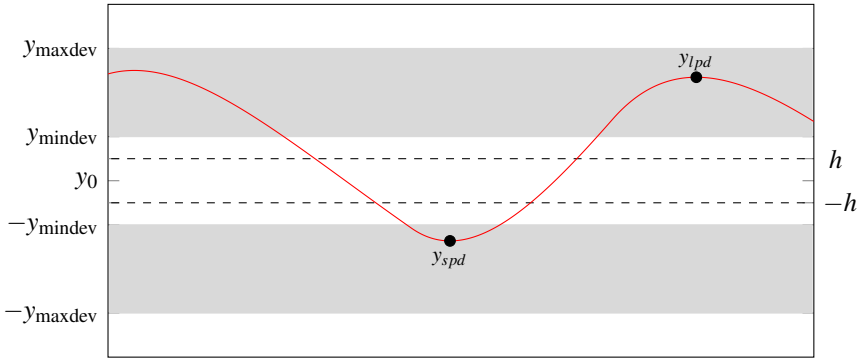
Figure 5.8 Oscillation restrictions for the process output y. The peak values y_{lpd} and y_{spd} should both stay within the gray-marked areas. To get good results it is necessary that y_{mindev} is a bit larger than the hysteresis level h. The limit y_{maxdev} is due to the fact that the process should not be disturbed too much, this limit may be set by the operator.

> **if** *Switching AND not changed in last switch* **then**
> > **if** $\max(y_{dev}) > y_{maxdev}$ **then**
> > > $y_{lpd}^* = (y_{\mathrm{maxdev}} + y_{\mathrm{mindev}}\gamma)/2$
> > > ratio $= y_{lpd}^* / \max(y_{\mathrm{dev}})$
> >
> > **else if** $\max(y_{dev}) < y_{mindev}$ **then**
> > > $y_{spd}^* = (y_{\mathrm{mindev}} + y_{\mathrm{maxdev}}/\gamma)/2$
> > > ratio $= y_{spd}^* / \max(y_{\mathrm{dev}})$
> >
> > **else**
> > > ratio $= 1$
> >
> > **end**
> > $d_1 = d_1 \cdot$ ratio
> > $d_2 = d_2 \cdot$ ratio
>
> **end**

Algorithm 1: Amplitude adjustments. Here y_{dev} stands for the process value's deviation from the setpoint, y_{spd}^* and y_{lpd}^* stands for the wanted small peak deviation and large peak deviation respectively.

To exemplify, consider a situation when the lower limit is set to $y_{\mathrm{mindev}} = 2h$, the upper limit is set to $y_{\mathrm{maxdev}} = 6h$ and $\gamma = 2$. If the relay would have been symmetric we would aim for the peak values, y_{spd}^* and y_{lpd}^* to reach $4h$, but in an asymmetric relay the asymmetry level γ needs to be taken into consideration as in Algorithm 1. The desired amplitudes in this case are

$$y_{spd}^* = (y_{\mathrm{mindev}} + y_{\mathrm{maxdev}}/\gamma)/2 = (2h + 6h/\gamma)/2 = 2.5h,$$
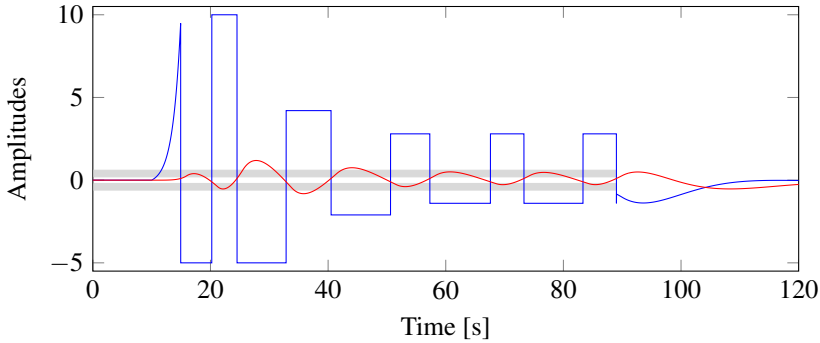$$y_{lpd}^* = (y_{\mathrm{maxdev}} + y_{\mathrm{mindev}}\gamma)/2 = (6h + 2h\gamma)/2 = 5h.$$

Figure 5.9 An example of the startup and amplitude adjustments. Here $u_{\mathrm{maxdev}} = 10$, $\gamma = 2$, $y_{\mathrm{mindev}} = 2h$, $y_{\mathrm{maxdev}} = 6h$. The gray area shows the allowed areas for the peak values y_{spd} and y_{lpd}.

The obtained ratio adjusts the measured amplitudes towards the desired values. An example of the startup and amplitude adjustments is illustrated in Figure 5.9.

It follows from the equations in Algorithm 1 that the value of γ is restricted to

$$\gamma \leq \frac{y_{\mathrm{maxdev}}}{y_{\mathrm{mindev}}}, \tag{5.4}$$

otherwise both limits can not be satisfied at the same time. Either the experiment will then not converge at all, or the converged limit cycles will not satisfy the limitations. An example of when γ is too large in comparison to the upper limit y_{maxdev} is shown in Figure 5.10. As can be seen the experiment will never converge with these parameter settings and program logic warning for this situation has to be a part of the autotuner implementation.

5.3 Measurement Noise

To check the experiment's robustness to measurement noise, the test batch was simulated for some different noise levels. To introduce noise to the simulations, band-limited white noise was connected to the process output. As was mentioned in Section 5.1 the relay experiment starts by measuring the noise level, and a suitable hysteresis level for the experiment is then chosen. The noise level for the different simulations in the batch was the same since the same seed was used by the noise block in all simulations. The accuracy of K_p and τ for different noise levels is shown in Figure 5.11, and the corresponding boxplots are shown in Figure 5.12 and Figure 5.13 respectively.

Figure 5.11 shows that the processes with low τ give the worst estimates. This emphasizes that it could be worthwhile to put some extra effort in modeling these
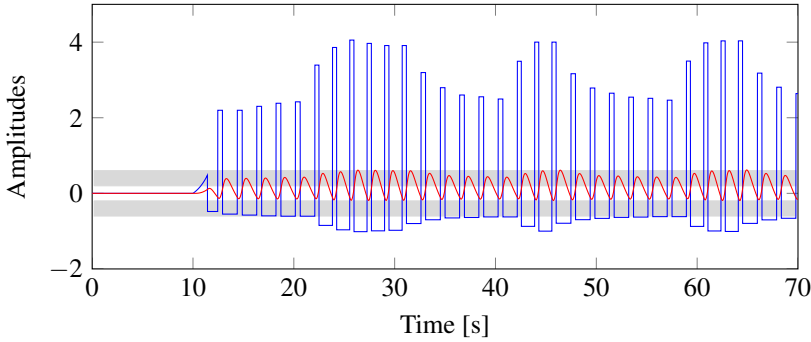
Figure 5.10   An example of the problems obtained if γ is larger than $y_{\mathrm{maxdev}}/y_{\mathrm{mindev}}$. The gray areas shows the allowed areas for the peak values y_{spd} and y_{lpd}. The algorithm will never find relay amplitudes that satisfies both limits and will keep changing the amplitude.

processes. The figures for K_p only contain the processes with an estimated $\tau > 0.1$, but Figure 5.11 shows that some of the processes get such a large error in τ that they are wrongly classified to have $\tau > 0.1$ in the most noisy simulation. These are the same processes that deviates a lot in their estimates of K_p. The results for the different noise levels do, however, not differ that much if the few miss-classified outliers are disregarded.

To further illustrate the effect of a noisy experiment, we take a closer look at Figure 5.14 where one of the processes from the test batch, namely

$$P(s) = \frac{1}{(s+1)^5}, \tag{5.5}$$

has been simulated with and without noise. The noise level in the upper plot was measured to $n_0 = 0.38$ and the estimates were $K_p = 0.95$ and $\tau = 0.45$, which can be compared to the noise-free estimates $K_p = 1.01$ and $\tau = 0.44$. Hence, the introduction of noise did not deteriorate the accuracy of the estimates in this example. Worth noting is that in the upper plot the hysteresis level is $h = 2n_0 = 0.76$ instead of the default value $h = 0.1$ used in the noise-free case. Since the default values of both the upper and lower limit on the deviations of the process signal are set proportional to the hysteresis level, the amplitude is both forced and allowed to be much larger when noise is added. In reality the upper limit on the deviation may not be allowed to be so large. This gives that it may be necessary to filter the signal before performing the relay experiment if the noise level is too large.
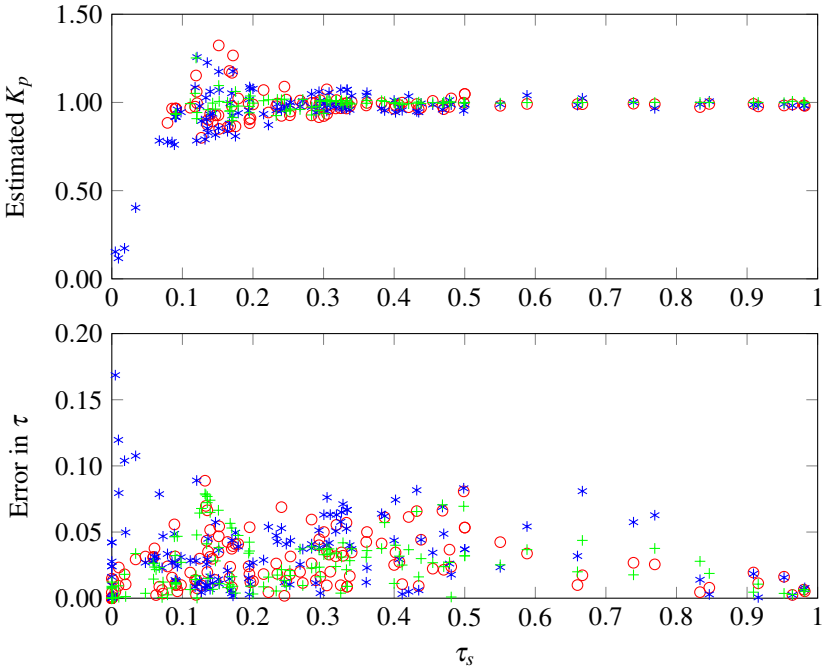
Figure 5.11 Results from relay experiment for the different noise levels $n_0 = 0$ in green plus signs, $n_0 = 0.12$ in red circles, and $n_0 = 0.38$ in blue stars. The upper plot shows the estimated values of K_p as a function of τ_s. The lower plot shows the absolute error between the estimates of τ and the true values τ_s, as a function of τ_s. In the plot for K_p, only the processes classified as non-integrating are included.
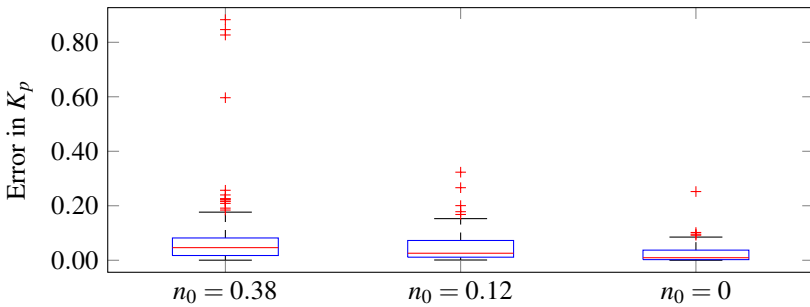

Figure 5.12 Boxplots of the absolute errors of K_p, shown for different noise levels. All processes, classified as non-integrating, from the test batch are included.
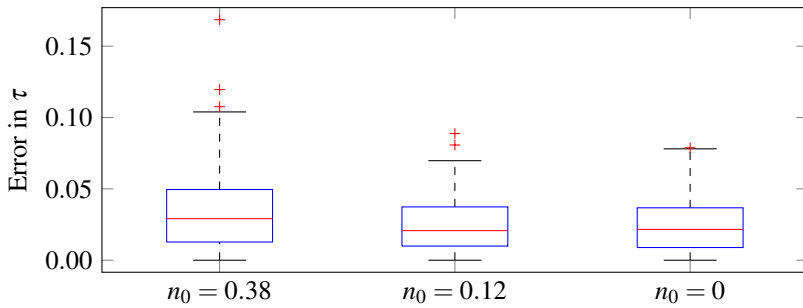
Figure 5.13   Boxplots of the absolute errors of τ, shown for different noise levels.
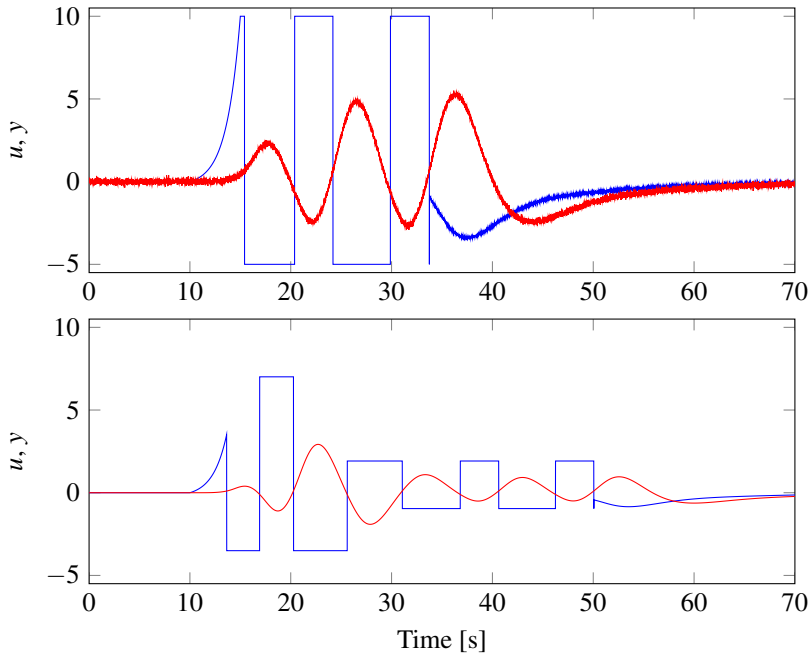

Figure 5.14   Relay experiment for the process $P(s) = 1/(s+1)^5$. The upper plot shows the signals when noise is added to the process output, while the lower plot shows the corresponding noise-free simulation. The blue line shows the relay output, the red line shows the process output.
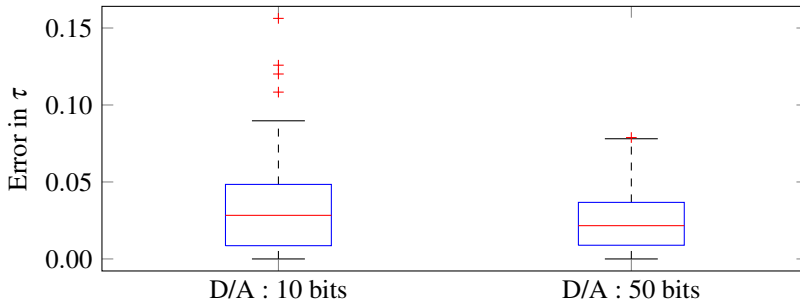
51

Figure 5.15 Boxplots of the absolute errors of τ, shown for two different resolutions of the D/A converter.

5.4 Effects of Quantization

The resolution of A/D and D/A converters may affect the autotuner performance. The largest issue is the resolution in the D/A converter, since it will change the asymmetry level of the relay. A small change in relay asymmetry can give large deviations in the calculations of τ and K_p. To see the effect of quantization the test batch was simulated with a D/A converter that had a resolution of 10 bits on the control interval [0,100]. The errors in τ are shown in Figure 5.15, where the results are compared to the default setup that tries to mimic infinite resolution by using a 50 bit D/A converter.

The results are worse for the less resolved D/A converter. The reason to the difference can be understood by looking at the example shown in Figure 5.16. This example shows the worst case achieved in the test batch simulation. The process is

$$P(s) = \frac{e^{-0.7s}}{s(0.3s+1)}, \tag{5.6}$$

so it is an integrating process that should have $\tau = 0$, but due to the quantization the estimated value is $\tau = 0.16$. The oscillations in the figure looks fine, but still the resulting model is bad. The relay amplitude before the D/A converter is shown in blue while the actual control signal that enters the system is the turquoise line. The levels in this example are $d_1 = 0.88$, $d_1^q = 0.88$, $d_2 = 0.44$, $d_2^q = 0.49$, where the q denotes that it is the quantized level. This gives that $\gamma = 2$ while $\gamma^q = 1.8$. So we think that we have an asymmetry level of 2 but in reality it is 1.8. In Figure 5.17 we see the implications of this on the estimated τ. Since the actual asymmetry level is $\gamma^q = 1.8$, the integrating system gets the half period ratio $\rho = \gamma^q = 1.8$. Assuming the curve where $\gamma = 2$ gives $\tau = 0.16$ when the true τ-value is 0.

The problem with quantization can be resolved by having a higher resolution of the converters, or by knowing the control signal levels that are actually sent to the process. If we had known, in the example above, that the true value was $\gamma = 1.8$ and used that in the calculations there would have been no problem.
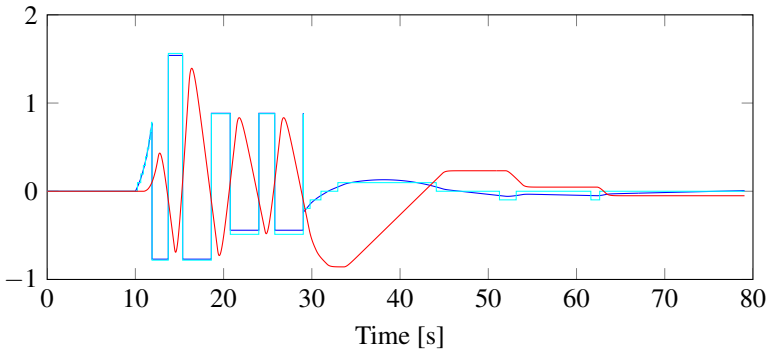
Figure 5.16   The effect of quantization in the D/A converter. The calculated control signal u is shown in blue, while the output from the 10 bit D/A converter, u^q, is shown in turqoise. The process output y is shown in red. The final relay amplitudes are $d_1 = 0.88$, $d_1^q = 0.88$, $d_2 = 0.44$, $d_2^q = 0.49$.
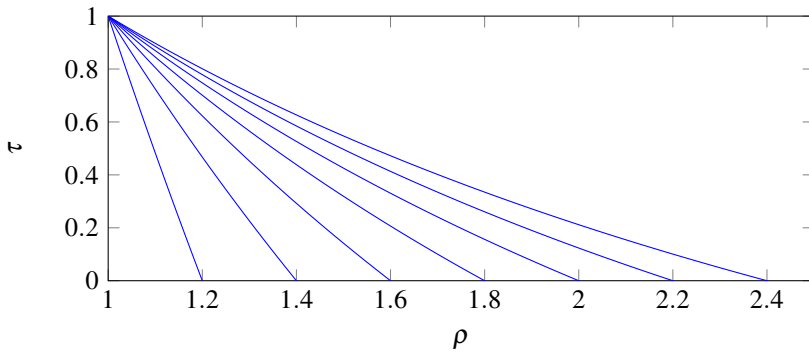

Figure 5.17   The τ curves corresponding to (3.6), plotted for different values of the asymmetry level γ.

5.5   Load Disturbances

Static load disturbances that enter the system during the relay experiment can create problems. If the load disturbance is large, it may stop the process from oscillating and no result is achieved. This will be obvious to the operator and the experiment can be restarted, hopefully without any disturbances. If the load disturbance is small, we get a problem similiar to the quantized D/A converter described in Section 5.4. Bad parameter estimates are obtained, since the desired relay signal is not what actually enters the process. If this is not noticed in the validation phase, bad controller parameters may be used on the process. It is therefore desirable to be able to detect if a load disturbance is present. With a symmetric relay, load disturbances are easily detected since the oscillations will become asymmetric. The magnitude of the disturbance could be determined and the relay experiment could then either be restarted with a bias to compensate for the load, or calculations could be modified to take account of the disturbance. Some different approaches to handle load disturbances for symmetric relay feedback are described in [Hang et al., 1993], [Shen et al., 1996a], [Park et al., 1997] and [Sung and Lee, 2006].

When an asymmetric relay is used, the detection of static load disturbances is more difficult, since there is no way to determine whether the asymmetry in the oscillations comes from the relay or from a disturbance. In [Kaya and Atherton, 2001a] a method to find the parameters of a stable or unstable FOTD or SOTD model with an asymmetric relay and a static load disturbance is presented. The method requires knowledge of the static gain of the process in order to calculate the magnitude of the load. The same methodology is used to estimate the parameters of an IFOTD model in [Kaya, 2006]. Hence, a small static load disturbance is not a big problem, if either the magnitude of the load disturbance or the process gain is known. Usually that is not the case though, which makes a short experiment time of the relay experiment even more important, since that decreases the risk of having a load disturbance entering during the experiment.

5.6   Start in Non-Steady State

As stated earlier, it is assumed that the process is at the steady-state level (u_0, y_0) when the experiment is started. In this section it is investigated how the results are affected if this assumption is violated. To make the investigations a step change in the reference signal was conducted, with a reasonably well-tuned controller. The relay experiment was started when the process output y had almost reached its reference value. The error between the reference value and y at the starting point of the experiment is denoted y_e. After the relay experiment was done a load disturbance was added to see the achieved controller performance. The controller in this case was a PI controller tuned with the AMIGO rules for the obtained FOTD model. The results of the experiments are shown in two different figures. In Figure 5.18 the relay

experiment starts immediately when the error is small enough. In Figure 5.19 the experiment starts with measurement of the noise level, as described in Section 5.1.

As can be seen from Figure 5.18 the resulting controller parameters are good for the first two cases where the starting error was $y_e = 0.001$ and $y_e = 0.01$ respectively, slightly worse for $y_e = 0.1$ and really bad for $y_e = 0.5$. In Figure 5.19 the results for $y_e = 0.5$ is much better. In this case the process reaches a steady-state during the noise measurement phase. The reason why the result is still slightly worse than for the upper plots is that the reference value used in the relay experiment is an average of the *y*-values measured during the entire noise measurement phase and hence slightly different from the steady-state level obtained.

To conclude the results for start in non-steady state, it is clear that you want the system to have reached its equilibrium before starting the experiment. However, if the system is almost at steady-state the results are still reasonable. From the lower plots in Figure 5.18 and Figure 5.19 where $y_e = 0.5$ it is obvious that the process is not in steady-state and that the experiment should not yet be started, still we may get reasonable results if the process is fast enough to reach steady-state during the noise measurement phase.
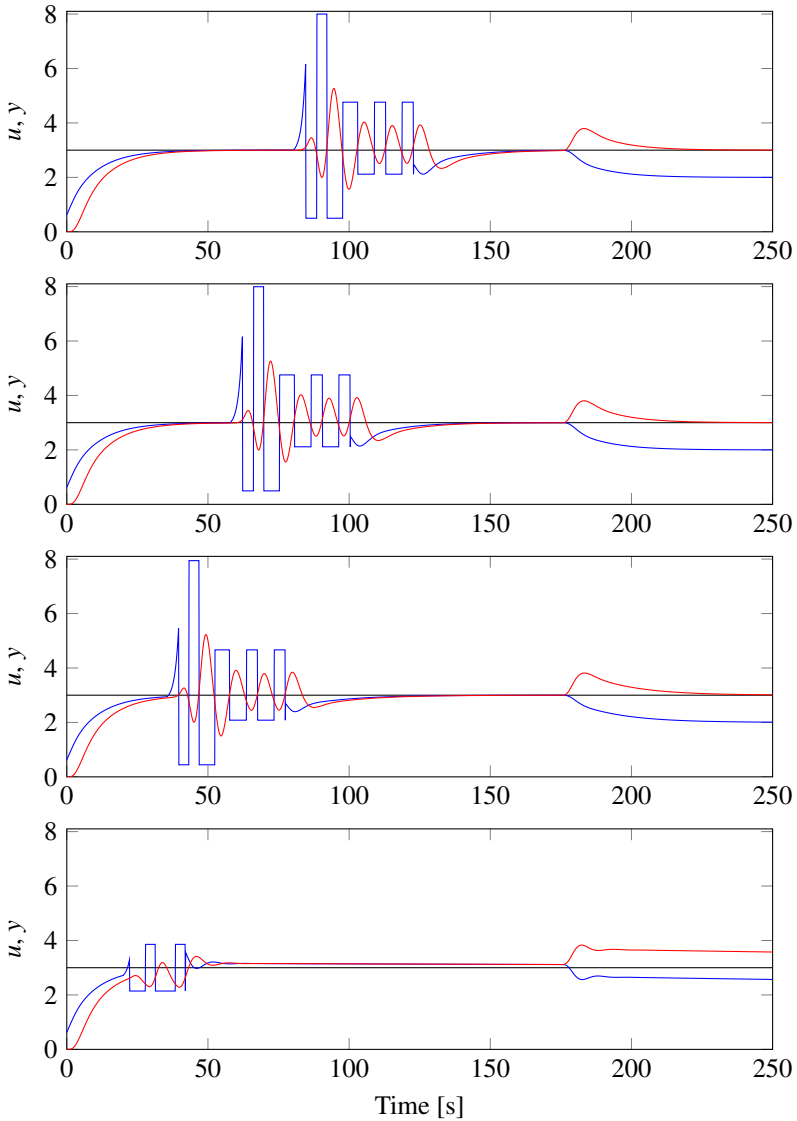
Figure 5.18 Results when the relay experiment is started at a non-steady state. The error at the experiment start is $y_e = [0.001, 0.01, 0.1, 0.5]$, listed from the top figure to the bottom one. The resulting PI controller parameters achieved from the experiment are $K = [0.26, 0.26, 0.24, 0.75]$ and $T_i = [3.38, 3.38, 4.05, 171.15]$, listed from top to bottom. At time 175 s a step load disturbance is added and the control performance can be seen.

Figure 5.19 Results when the relay experiment is started at a non-steady state. In this figure the experiment starts with measuring the noise level for 10 s before the relay starts. The error at the experiment start is $y_e = [0.001, 0.01, 0.1, 0.5]$, listed from the top figure to the bottom one. The resulting PI controller parameters achieved from the experiment are $K = [0.26, 0.26, 0.27, 0.24]$ and $T_i = [3.38, 3.38, 4.03, 4.40]$, listed from top to bottom. At time 175 s a step load disturbance is added and the control performance can be seen.
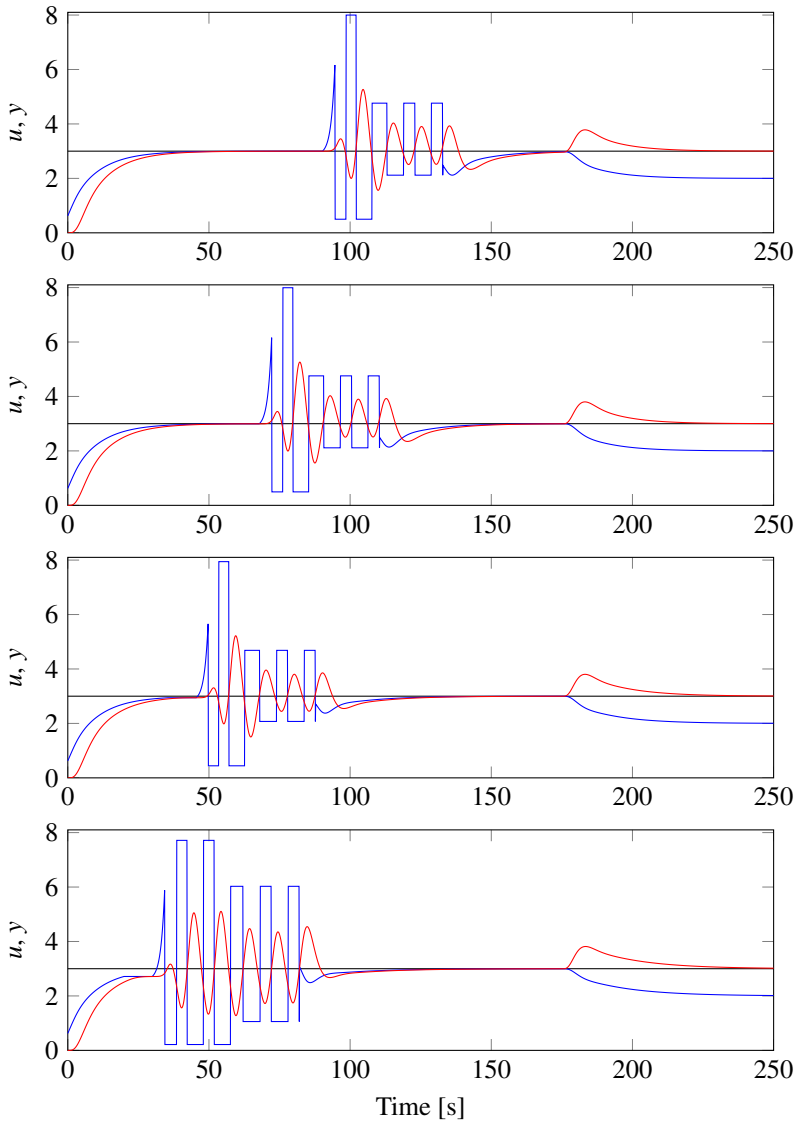
6

Examples

In this chapter, three processes are investigated more in detail, to evaluate the accuracy of the described autotuner procedure. The three processes are

$$P_1(s) = \frac{1}{(s+1)(0.1s+1)(0.01s+1)(0.001s+1)},$$

$$P_2(s) = \frac{1}{(s+1)^4}, \tag{6.1}$$

$$P_3(s) = \frac{1}{(0.05s+1)^2}e^{-s},$$

where P_1 is lag dominated, P_2 balanced, and P_3 delay dominated. All simulations in this chapter have been performed in Matlab/Simulink with the default parameters listed in Section C, and using the startup and adaptive relay amplitudes described in Section 5.2. The simulation model contains one block with the PID controller, and another block containing all the code for the relay experiment. The PID controller is active until the experiment is started, and is reconnected as soon as the new controller parameters are calculated. The simulations have been performed in discrete time with a fix sampling time, and the models and the PID controller described in Section 2.1 are therefore discretized as well. However, all parameters are derived for the continuous case.

For each of the processes one model is obtained from the relay experiment, and the system identification method described in Section 3.4 has been used to get an additional model. The models are compared with each other and with the true process by the Vinnicombe norm. They are also compared by a modified version of the Vinnicombe norm, where only the frequency range of interest for PID controllers are considered. Controllers based on the AMIGO rule are tuned for all approximate models. The obtained controllers are compared to optimal controllers for the true process with respect to performance and robustness.
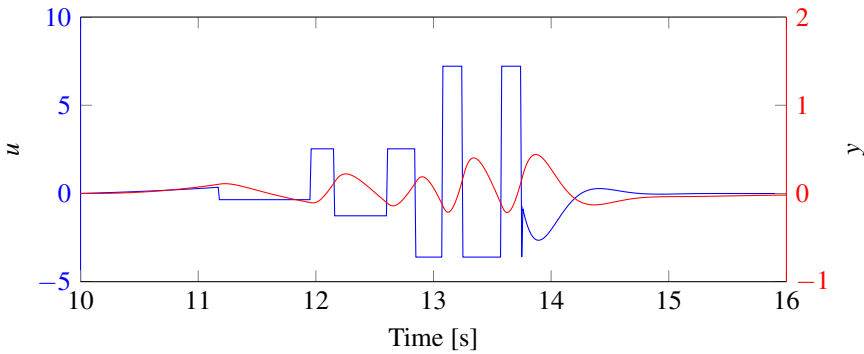
Figure 6.1 Signals from the relay experiment for P_1. The blue line shows the relay output u, and the red line shows the process output y. Note the different scales on the y axes.

6.1 The Lag Dominant Process P_1

The relay experiment for process P_1 is shown in Figure 6.1. The first 10 seconds of the experiment the noise level is measured, this is omitted in the figure. When the noise level is established, the relay amplitude is ramped up until the process output leaves the hysteresis band. Since this is a fast process this happens quite rapidly, when the relay amplitude is still small. In the first half-periods, the oscillation amplitude is lower than desired, and the relay amplitudes adapt to a larger value during the experiment. Once the appropriate amplitudes are achieved, the experiment continues for four half-periods until the convergence limit is satisfied at time 13.75 s. Subsequently the calculations to get the model and controller parameters are performed. A PI controller is connected as soon as its new parameters are calculated, and immediately starts to bring the process output back to the reference value.

The normalized time delay for this process is estimated to $\tau = 0.044$ which could be compared to the τ obtained from a step response that is $\tau_s = 0.067$. Since τ is small an ITD process is estimated in the autotuner. The system identification method is then used to find an IFOTD model with start values from the ITD model. The model parameters as well as their Vinnicombe norms are listed in Table 6.1. The resulting models are also shown in Figure 6.2 where they have been simulated with the experiment data as input. The figure shows that the IFOTD model gives a somewhat better fit than the ITD model, but none of them fits perfectly.

Table 6.1 Resulting model parameters and Vinnicombe norms for P_1.

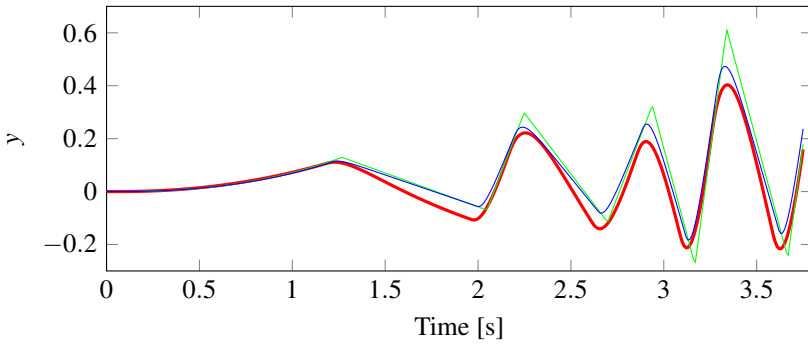| Model	k_v	T	L	δ_v	δ_v^*
ITD relay	0.727		0.088	0.707	0.087
IFOTD est	0.681	0.038	0.038	0.707	0.090

Figure 6.2 Simulated model output when the experiment data is used as input. The true experiment output is shown in red, the output from the ITD model is shown in green, and the output from the IFOTD model is shown in blue.

Table 6.1 shows that the Vinnicombe norms are bad for both models, $\delta_v = 0.707$, which is due to the fact that the models both have an integrator, while the real process does not. This make the models differ significantly at low frequencies. If the modified Vinnicombe norm is considered instead, the results are much better with $\delta_v^* \approx 0.09$ for both models.

PI and PID controllers for the ITD model as well as a PID controller for the IFOTD model, are tuned using the AMIGO rules. The resulting controller parameters, and robustness and performance measures M_{ST} and IAE, are listed in Table 6.2. The table also contains the same values for optimal PI and PID controllers for comparison. The optimal controllers are obtained by minimization of IAE with the constraint that $M_{ST} \leq 1.4$. Step responses to a load disturbance are shown in Figure 6.3 for all the controllers. The results verify the statements made for low values of τ. The derivative part is beneficial, the PID controllers perform much better than the PI controllers, and better modeling can increase the performance significantly. The PID controller tuned for the ITD model is a factor 100 worse in performance than the optimal PID controller. However, even the simple models obtained from this experiment give low values of IAE, and both the PID controllers for the simple models are performing better than the optimal PI controller. So the results are not bad, they

Table 6.2 Controller parameters for P_1.

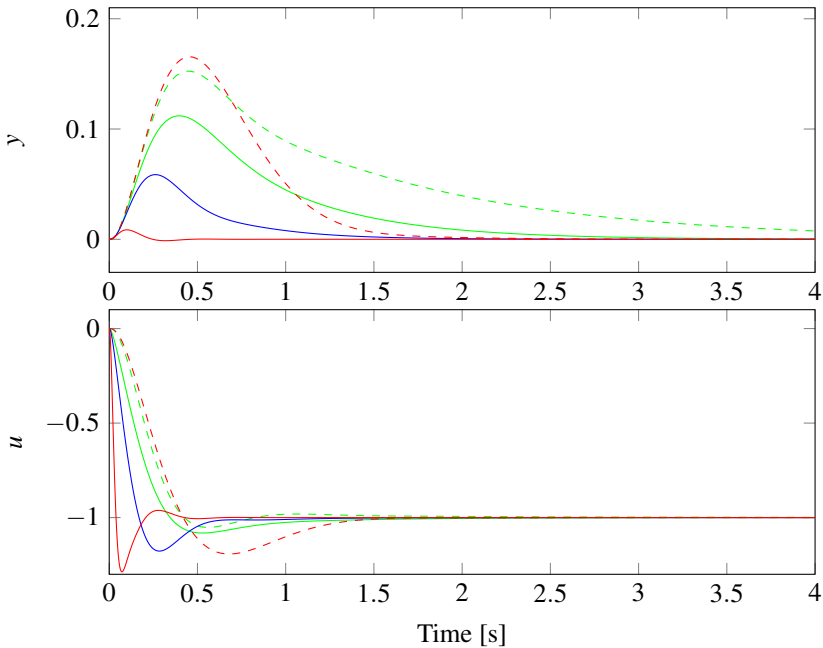| Controller	K	T_i	T_d	M_{ST}	IAE
ITD PI	5.478	1.178		1.34	0.215
ITD PID	7.043	0.703	0.044	1.15	0.100
IFOTD PID	15.28	0.469	0.043	1.23	0.031
Optimal PI	4.200	0.494		1.40	0.118
Optimal PID	89.50	0.086	0.051	1.40	0.001

Figure 6.3   Responses from a step load disturbance, for the closed loop systems containing the true process P_1 and the different controllers listed in Table 6.2. The upper plot shows the process output, the lower plot shows the control signal. The dashed lines correspond to PI controllers, while the solid lines show PID controllers. The green lines show the controllers based on the ITD model, the blue lines show the controller based on the IFOTD model, the red lines show the optimal controllers.

could just be made even better by more advanced modeling and tuning. Notable are also the gains of the PID controllers, especially the optimal one, that may prove to be too high for noisy applications.

6.2   The Balanced Process P_2

The signals from the relay experiment for P_2 are shown in Figure 6.4. The plot is similar to the one for P_1, but this process is a bit slower which gives a longer total experiment time. Notice that the amplitude was considered too high during the third half-period, so the relay amplitude was lowered to get the desired oscillation amplitudes.

The estimated τ for this process is $\tau = 0.37$ which can be compared to its true value $\tau_s = 0.33$. The obtained value of τ implies that an FOTD model is estimated from the experiment, but that an SOTD model estimated with the system identifica-
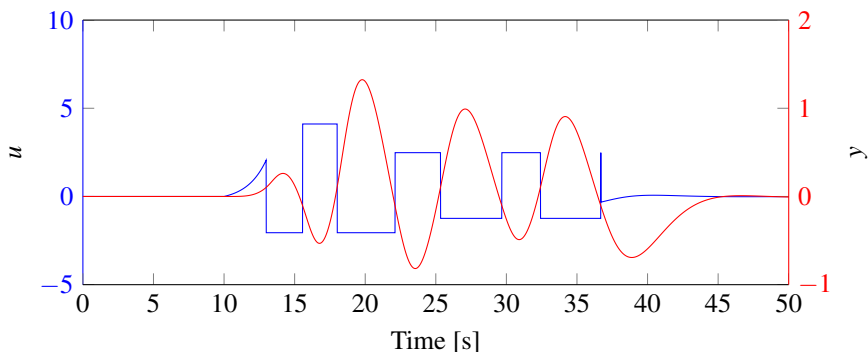
Figure 6.4 Signals from the relay experiment for P_2. The blue line shows the relay output u, and the red line shows the process output y. Note the different scales on the y axes.
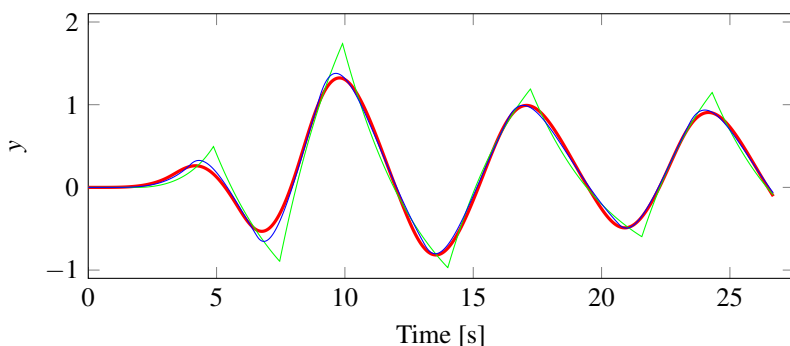

Figure 6.5 Simulated model output when the experiment data is used as input. The true experiment output is shown in red, the output from the FOTD model is shown in green, and the output from the estimated SOTD model is shown in blue.

tion method might give better results. The obtained models are shown in Table 6.3 along with their Vinnicombe norms. The models are also simulated against the experiment data in Figure 6.5. As can be seen from both the table and the figure, the SOTD model is a much better approximation than the FOTD model. This is due to the fact that an FOTD model can not represent the multiple identical poles in P_2 in a satisfactory way.

A PI controller and a PID controller are tuned for the FOTD model, and a PID controller is tuned for the SOTD model, using the AMIGO rules. The controller parameters and the obtained values of IAE and M_{ST} are listed in Table 6.4. Optimal controllers are listed as well for comparison. In Figure 6.6 the performance of the obtained controllers are shown for a step change in load disturbance. As can be seen the benefit of the derivative part is obvious for this process as well.
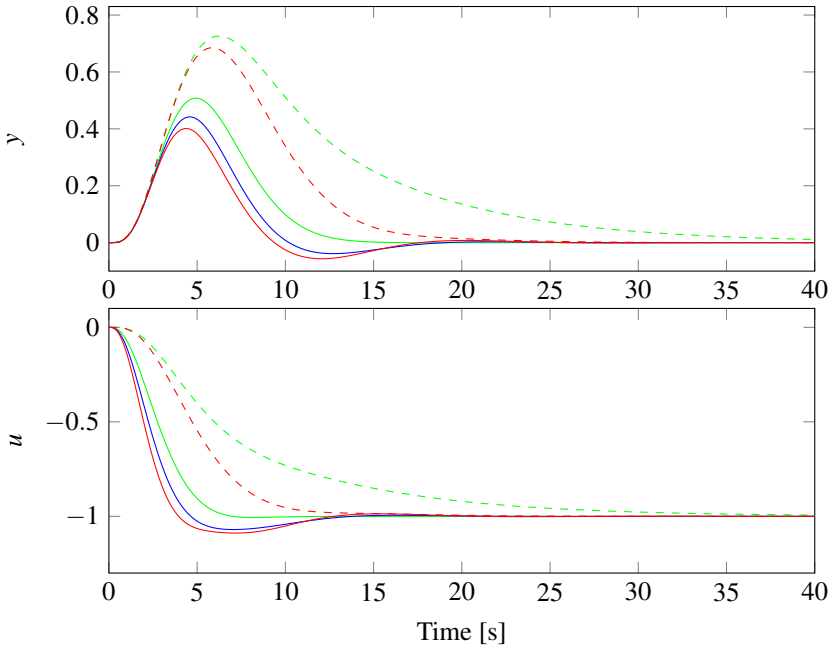
Figure 6.6   Step responses from a load disturbance, shown for the closed loop systems containing the true process P_2 and the different controllers listed in Table 6.4. The upper plot shows the process output, and the lower plot shows the control signal. The dashed lines correspond to the PI controllers, while the solid lines show the PID controllers. The green lines show the controllers based on the FOTD model, the blue lines show the controller based on the SOTD model, and the red lines show the optimal controllers.

Table 6.3   Resulting model parameters and Vinnicombe norms for P_2.

Model	K_p	T_1	T_2	L	δ_v	δ_v^*
FOTD relay	0.987	3.225		1.889	0.124	0.124
SOTD est	1.054	1.762	1.762	1.004	0.057	0.057

Table 6.4   Controller parameters for P_2.

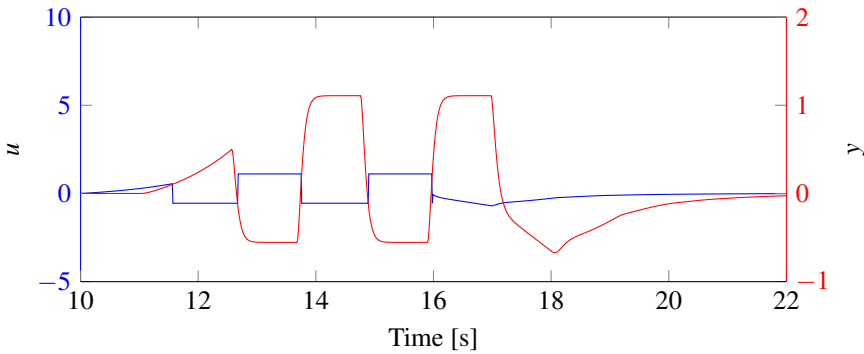| Controller	K	T_i	T_d	M_{ST}	IAE
FOTD PI	0.355	3.015		1.24	8.487
FOTD PID	0.981	2.849	0.803	1.35	2.906
SOTD PID	1.191	2.354	1.111	1.37	2.348
Optimal PI	0.432	2.250		1.39	5.208
Optimal PID	1.331	2.110	1.340	1.40	2.134

Figure 6.7   Signals from the relay experiment for P_3. The blue line shows the relay output u, and the red line shows the process output y. Note the different scales on the y axes.

6.3   The Delay Dominant Process P_3

The relay experiment for the delay dominant process P_3 is shown in Figure 6.7. The figure shows that the process is fast, the total experiment time, including 10 s of noise measurements before the relay oscillations are started, is 16 s. The time delay of 1 s is apparent from the oscillations, and since the dynamics are very fast the process reaches its steady state fast after each relay switch. Notice also that the half-periods t_{on} and t_{off} are almost the same, which matches the limit equations for $\tau \approx 1$ in (A.27). The normalized time delay for the process is estimated to $\tau = 0.93$, indicating that an FOTD model probably describes the process sufficiently. However, an SOTD model is estimated by the parameter estimation method as well as a comparison. The resulting models are shown in Table 6.5 and in Figure 6.8.
 The table shows that the estimated SOTD model is in fact the true process. The FOTD model, obtained directly from the simple equations on the relay data, is not far from the true process in either a fit aspect or when the Vinnicombe norm for the interesting controller region is considered.

Controllers are, just like before, tuned by the AMIGO rules for the obtained FOTD and SOTD models, and compared with optimal controllers in Table 6.6 and Figure 6.9. All the controllers show similar performance and robustness. This implies that there is no need for either the derivative part, or more accurate modeling than the FOTD model.

Table 6.5   Resulting model parameters and Vinnicombe norms for P_3.

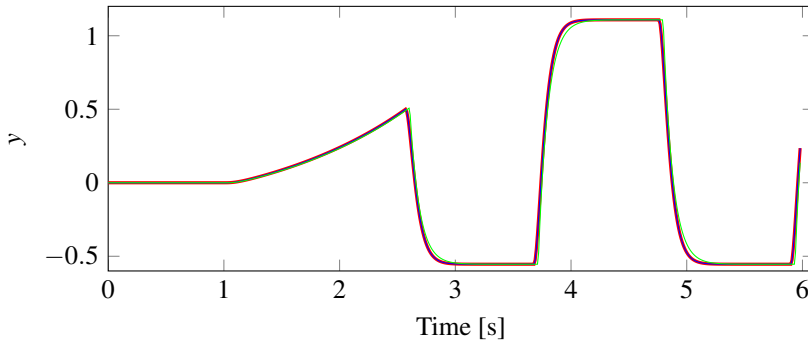| Model	K_p	T_1	T_2	L	δ_v	δ_v^*
FOTD relay	1.000	0.082		1.036	0.168	0.036
SOTD est	1.000	0.050	0.050	1.000	0	0

Figure 6.8 Simulated model output when the experiment data is used as input. The true experiment output is shown in red, the output from the FOTD model is shown in green, and the output from the estimated SOTD model is shown in blue.

Table 6.6 Controller parameters for P_3.

Controller	K	T_i	T_d	M_{ST}	IAE
FOTD PI	0.172	0.373		1.44	2.158
FOTD PID	0.236	0.477	0.108	1.41	2.014
SOTD PID	0.218	0.452	0.129	1.40	2.069
Optimal PI	0.164	0.371		1.40	2.313
Optimal PID	0.201	0.400	0.140	1.40	1.988

6.4 Discussion

The more detailed study of the three processes performed in this chapter strengthens the proposition that the normalized time delay is useful in the tuning procedure. It is clear that the derivative part is most useful for processes with low values of τ. Even though the obtained controllers from the simple version of the autotuner show satisfactory results, it is clear from the examples that better modeling, and also better tuning could be very useful for processes with a small normalized time delay.

The Vinnicombe norm gives a good measure of how close the estimated model is to the true process. However, a modified version of the measure gave a more appropriate comparison for our purposes. In Figure 6.10 the distances between the estimated and the true models are shown as a function of ω for each of the example processes. Looking at the upper plot, where the models of P_1 are shown, it is obvious that the Vinnicombe norm will be decided by the low frequencies. This since the estimated model has an integrator while the true process does not. The thicker lines show the frequencies where the true process phase is between $-90°$ and $-270°$. This covers the interesting frequency region for PID control. It is from this frequency interval that the modified Vinnicombe norm δ_v^* is calculated. For P_1
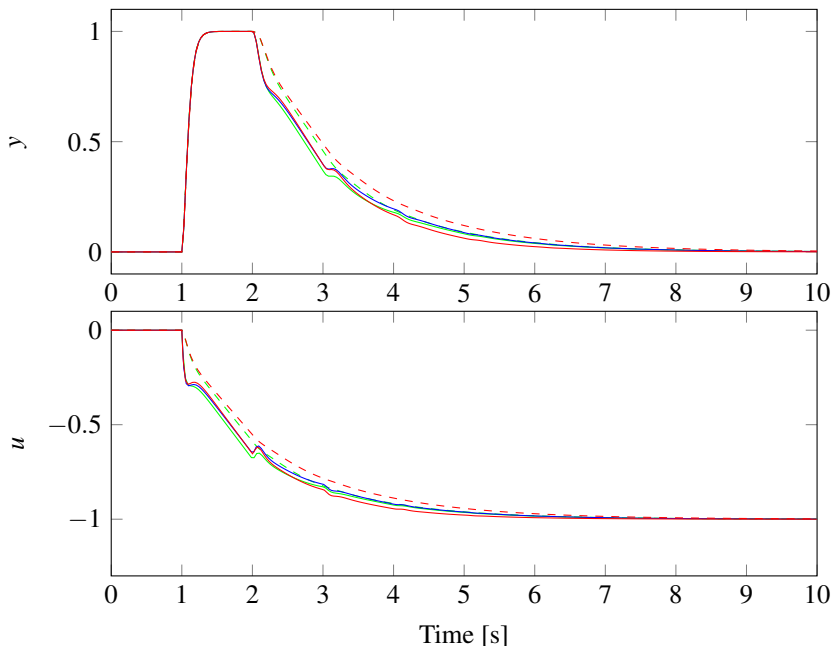
Figure 6.9 Step responses from a load disturbance, shown for the closed loop systems containing the true process P_3 and the five different controllers listed in Table 6.6. The upper plot shows the process output, and the lower plot shows the control signal. The dashed lines correspond to the PI controllers, while the solid lines show the PID controllers. The green lines show the controllers based on the FOTD model, the blue lines show the controller based on the SOTD model, and the red lines show the optimal controllers.

and P_3 the modification gives smaller values of the norm, while for P_2 the greatest distance between the models are actually in the interesting region and hence stays the same. Since the estimated SOTD model for P_3 coincides with the true process, its Vinnicombe norm is zero and does not show in the figure.

In the examples the tuning method used for the simple models is the AMIGO method, the obtained controllers are then compared to controllers that optimizes IAE. This is not a completely fair comparison since AMIGO is designed for another performance measure, namely the integral error (IE). Neither are the AMIGO rules designed for the models obtained from the relay experiment, but instead from models obtained by step response experiments. With a tuning rule designed from the obtained models, and with the same performance measure in mind, the results would probably be better.
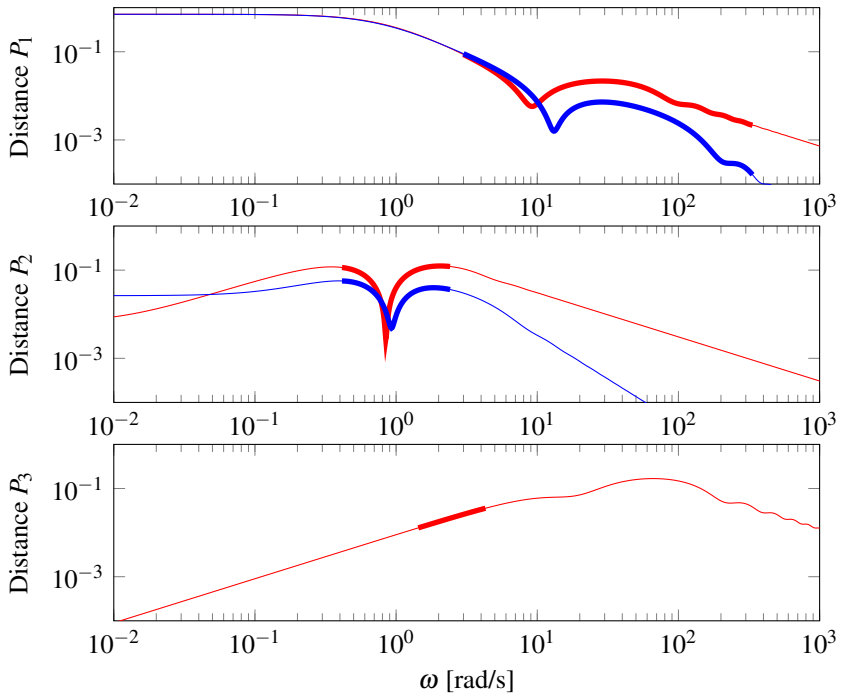
Figure 6.10 The distance measure of the Vinnicombe norm, as a function of the frequency ω. The red lines show the models estimated from the relay experiment, the blue lines show the models from the parameter estimation. The thick lines correspond to the frequency region used for the modified Vinnicombe norm.

7

Experimental Results

The autotuner was implemented and tested on an air handling unit provided by Schneider Electric Buildings AB in Malmö. The autotuner experiments were performed on two subsystems of the air handling unit. One subsystem that controls the pressure in the supply air duct, and another subsystem that controls the air temperature in the same duct. The implementation is described in Section 7.1. The system and subsystems are described in more detail in Section 7.2. The results from the experiments on the two subsystems are given in Section 7.3 and Section 7.4 respectively. Some remarks and discussion are presented in Section 7.5

7.1 Integration of the Autotuner in an Industrial System

To be able to test the autotuner on the air handling unit, it first had to be implemented in Schneider Electric's software StruxureWare Building Operation. The implementation of the autotuner procedure was made as a script program in the Building Operation server. During the implementation phase, the inputs and outputs from the script program were connected to a test system implemented as a function block program in the same server. This provided the possibility of code development and testing by simulations. The autotuner implementation includes all the sequences of the relay feedback experiment that were described in Section 4.1. The implementation uses the simple version of the autotuner, where the experiment data is used to find an FOTD model or an ITD model, and parameters for a PI/PID controller are tuned by the AMIGO rules.

   An implementation of the PID controller already existed, so no work was put on that part. The obtained controller parameters were manually entered into the PID controller during operation. To use the implemented autotuner on the air handling unit, the inputs and outputs from the script program, as well as the controller, were connected to the physical inputs and outputs instead of the simulation model.
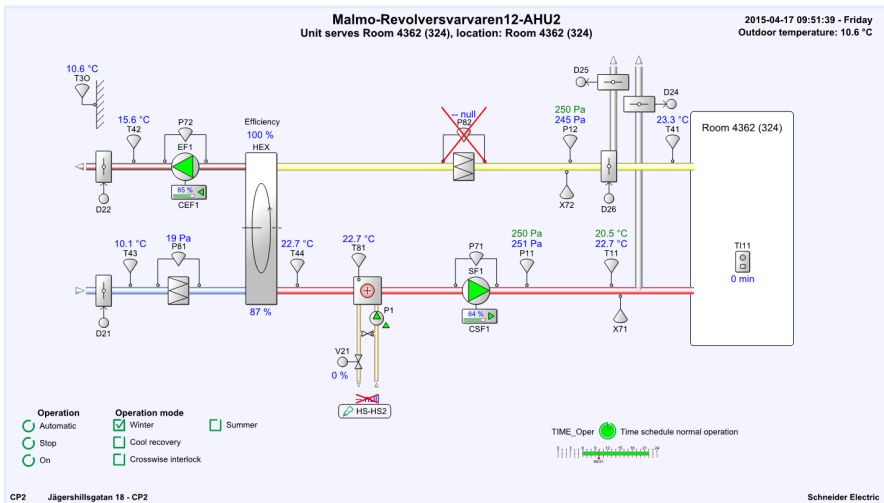
Figure 7.1   Schematics of the air handling unit.

7.2   System Description

The schematics of the air handling unit is shown in Figure 7.1, and pictures of the system are shown in Figure 7.2 and Figure 7.3. Outside air enters the system through the duct with a blue arrow sign, to the left in Figure 7.2. This corresponds to the lower left duct in the schematics. The air temperature in that duct is measured by sensor T43. The air then enters a box consisting of filters, the rotational heat exchanger and the fans shown in Figure 7.3. The heated air is then led on through the duct with the red arrow sign to the right in Figure 7.2, and enters the room from the white outlet vents in the roof. The air temperature and pressure in that duct are measured by sensors T11 and P11. If the temperature T11 is not sufficiently high when the heat exchanger runs at full speed, there is an additional pumping system circulating hot water to heat up the air. This system contains the pump P1 and valve V21, but was not used during the experiments.

The exhaust air follows a similar path, but in the opposite direction. The air is taken from the room through an intake in the roof. It then flows through the upper duct that enters the box in Figure 7.2 from the right. The temperature and the pressure of the exhaust air are measured by the sensors T41 and P12. The exhaust air flows through the heat exchanger where its temperature is used to heat up the inlet air, and the exhaust air then leaves the building through the duct with the brown arrow sign in Figure 7.2.

69

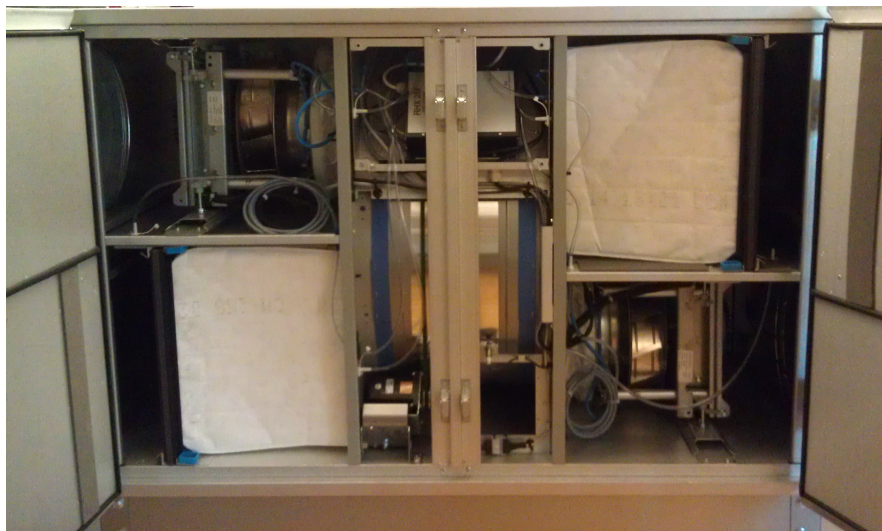**Figure 7.2** The air handling unit.

Figure 7.3 The heat exchanger, filters and fans. The supply air passes in the lower part, while the exhaust air passes in the upper part. The rotational heat exchanger in the middle transfers heat from the exhaust air to the supply air.
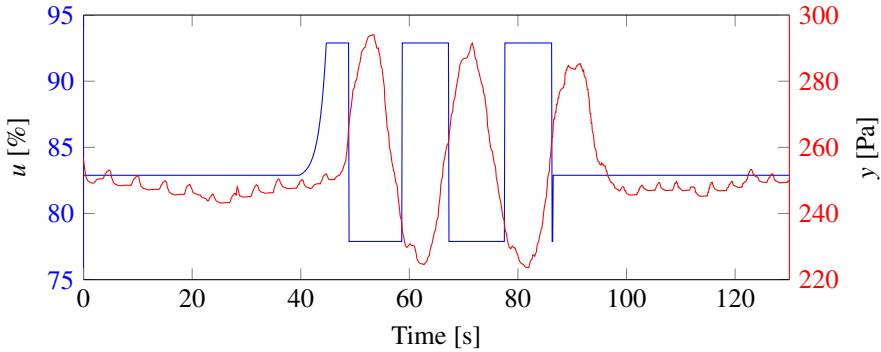
Figure 7.4 Experiment data from the pressure control loop. The normalized fan speed is shown in blue, the pressure measurement in red. Note the different scales and units on the axes.

7.3 Pressure Control

The pressure control loop consists of the supply air fan SF1, positioned inside the box, and the pressure sensor P11, positioned in the duct a few meters away from the box. The control signal is normalized to a percentage of the full speed of SF1, while the pressure is measured in the physical entity Pascal. The reference value of the pressure was set to 250 Pa. Two relay experiments were performed on the system. The calculated model and controller parameters for the two experiments are listed in Table 7.1. The second experiment is also shown graphically in Figure 7.4. The experiment started with 40 s measurement of the noise. The figure shows that the signal is rather noisy, in this experiment the noise level was measured to $n_0 = 6.51$ Pa. According to this noise level, the hysteresis value was set to $h = 2n_0 = 13.02$. The sample time used during the experiment was $t_s = 0.1$ s. The asymmetry level, convergence limit, maximum and minimum deviations and the maximum relay deviation, were set according to the default values in Section C.

In Table 7.1 it is seen that the results are quite similar between the experiments, with the largest difference in the estimates of K_p. Since τ was large, a PI controller was tuned. The second experiment gave a somewhat more aggressive controller tuning than the first one.

The controller parameters from the second experiment were used to investigate the obtained controller performance. The controller used a sampling time of 1 s, and

Table 7.1 Parameters from the pressure control experiment.

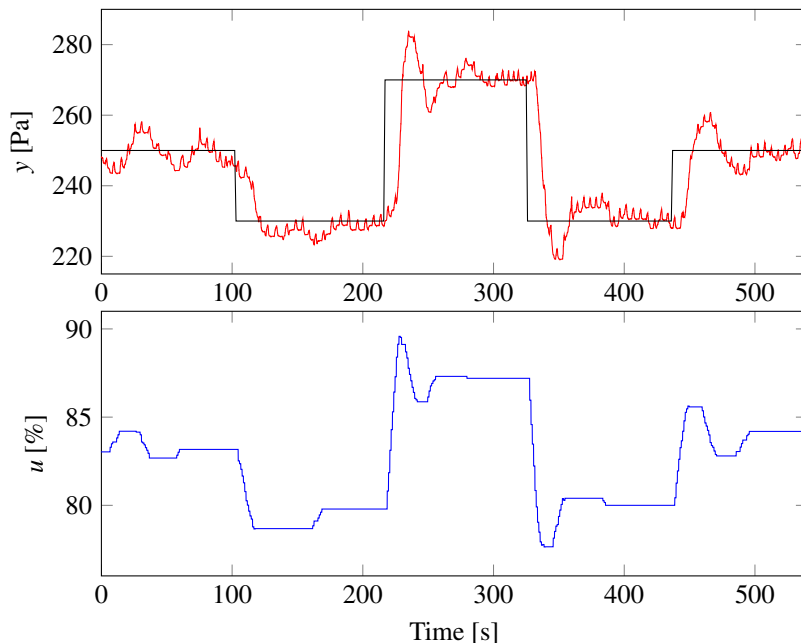|	τ	K_p	T	L	K	T_i
Exp. 1	0.71	3.56	2.50	6.07	0.059	3.23
Exp. 2	0.77	2.29	1.92	6.31	0.088	2.92

Figure 7.5 Response to setpoint changes for the system with the controller tuned from the experiment. The upper plot shows the measured pressure in red, and the setpoint in black. The lower plot shows the control signal.

had a dead zone of 5 Pa. Results from step changes in the reference value are shown in Figure 7.5. The step response results are satisfactory. There is an overshoot, but that can be reduced by filtering the setpoint. The use of the dead zone is clearly visible from the long periods of constant control signal and process output deviations from the setpoint.

By manually adjusting a damper, step load disturbances of unknown sizes were added, the response to these are shown in Figure 7.6. This also shows satisfactory results. The effect of the load disturbances are removed completely in approximately 20-25 s with rather small overshoots.

7.4 Temperature Control

In the temperature control experiment the supply air temperature T11 is controlled by the rotational heat exchanger HEX. The control signal is normalized to a percentage of the full rotation speed of the heat exchanger, and the temperature is measured in °C. The temperature of the inlet air was measured by T43 and varied a lot depending on the outdoor temperature. The temperature control experiments were
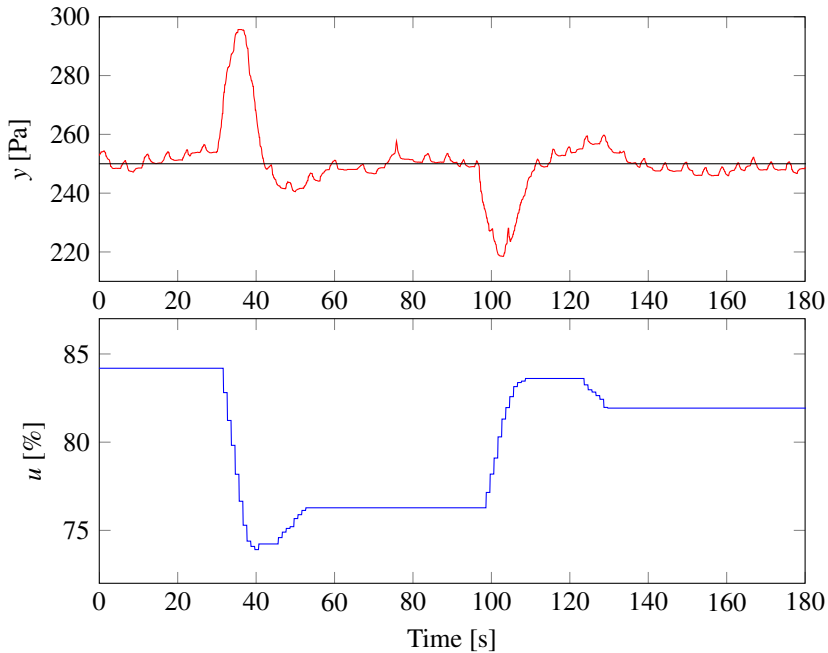
Figure 7.6 Response to load disturbances. The upper plot shows the measured pressure in red, and the setpoint in black. The lower plot shows the control signal.

made between 10.30 AM and 1.30 PM on a sunny spring day. During this time period the inlet air temperature measured by T43 varied according to Figure 7.7. This temperature deviation is treated as a load disturbance and caused some trouble during the experiments.

One relay experiment is shown in Figure 7.8. The noise was measured for 40 s and the measured noise level was $n_0 = 0.045°C$. This gave a hysteresis level of $h = 0.09$. The same default values as for the pressure control experiment were used, except for a change in the convergence limit from $\varepsilon = 0.01$ to $\varepsilon = 0.05$. The reason for this change is that the varying load from the inlet air temperature makes it even more crucial to keep the experiment time short. However, the higher steady state level after the experiment shows that the drift in temperature has still influenced the result. The measurements from the experiment gave that $\tau = 0.20$ and the estimated FOTD model was

$$P(s) = \frac{0.035}{35.8s + 1}e^{-8.8s}. \tag{7.1}$$

From this model the parameters for a PI controller were calculated to $K = 26.6$ and $T_i = 29.2$. The PI controller was then connected and some setpoint changes were performed to check the obtained controller performance. During the step tests the
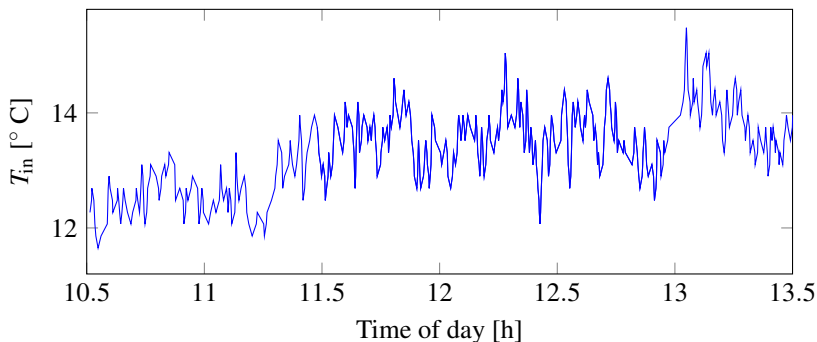
Figure 7.7 The temperature of the inlet air, T_{in}, measured by sensor T43.
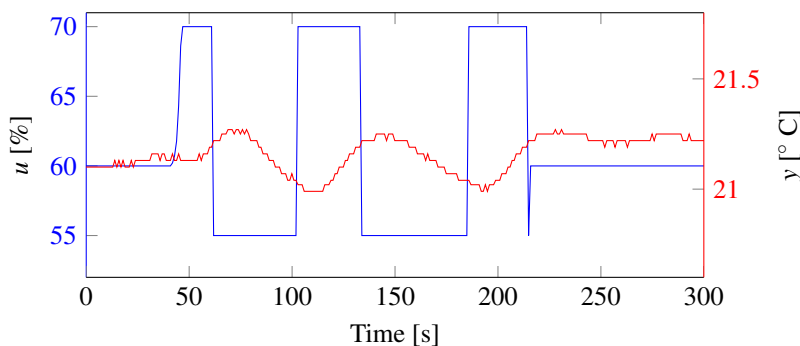

Figure 7.8 A relay experiment performed on the temperature control loop. The blue line shows the normalized speed of the heat exchanger, and the red line shows the supply air temperature. Note the different scales and units on the axes.

controller had a sampling time of 1 s and the dead zone was 0.025°C. The results from the step tests are shown in Figure 7.9. The reason why the setpoint is not constant during the steps is that it is continuously updated by an equation related to the outdoor temperature. The results from the step responses show that the obtained controller performs reasonable, even though the experiment was disturbed by the load that may have deteriorated the accuracy of the estimated model.

As an attempt to decrease the effect of the load disturbance, another relay experiment was done with higher allowance on the relay amplitudes, $u_{maxdev} = 20$. The data from this experiment is shown in Figure 7.10. This experiment showed another problem with the process. The heat exchanger is not linear, so increasing the speed from 60% to 80% does not double the effect compared to the previous increase from 60% to 70%. This nonlinearity is also seen in the step responses in Figure 7.9. The
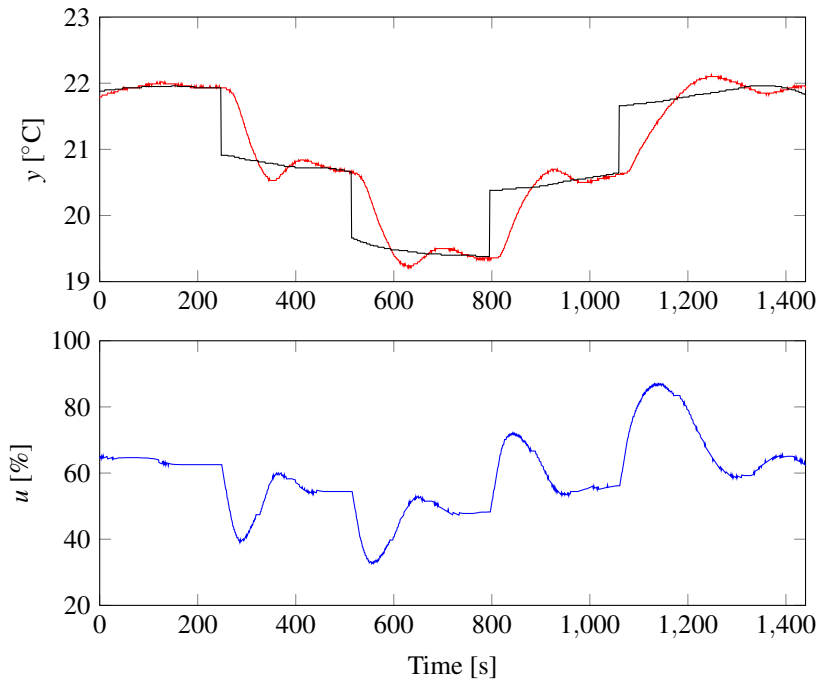
Figure 7.9   Response to setpoint changes. The upper plot shows the supply air temperature in red, and the setpoint in black. The lower plot shows the control signal.
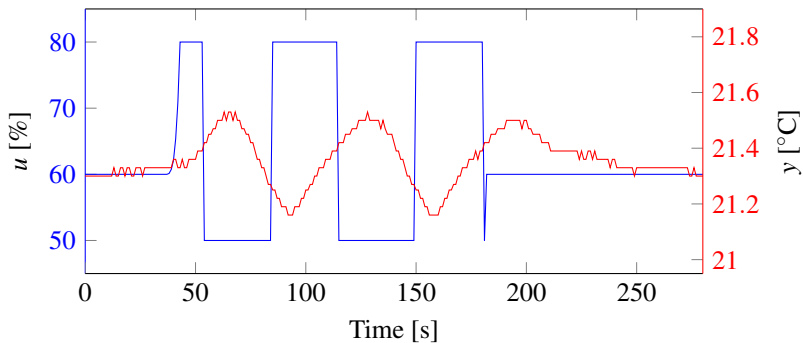

Figure 7.10   A relay experiment that suffers from the nonlinearities in the system. The blue line shows the normalized speed of the heat exchanger, and the red line shows the supply air temperature.

last step change, which is in the higher control signal range, demands much more control signal than the previous three steps, even though the steps are of the same size. In the relay experiment this nonlinearity changes the "true asymmetry level". In the experiment we used the asymmetry level $\gamma = 2$, but since the step of 20 units up does not have the double effect of 10 units down, the true value of γ is less than 2. This decrease in true asymmetry level is reflected in the half-period ratio ρ, and as can be seen in Figure 7.10, the half-periods t_{on} and t_{off} are more or less equal in this experiment. Using this data, assuming $\gamma = 2$, would give a value of τ close to 1 as for a delay dominant process. Since our previous experiments had shown that this wasn't correct, the model was disregarded.

7.5   Discussion

The experiments on the air handling unit gave satisfactory results for the pressure control loop. For the temperature control loop some problems occured due to load disturbances and nonlinearities in the system. The obtained controller did, however, show good results for the setpoint changes. The experiments also gave some new insights. One such insight is that the use of default values is not possible unless all signals are normalized. As an example consider the hysteresis value, which default value used in the simulations chapter was $h = 0.1$. In these experiments the scales of the physical entities were very different and in the pressure control the noise level was measured to $n_0 = 6.51$ Pa, while in the temperature control the noise level was $n_0 = 0.045°$C, where parts of it probably were subject to drift in inlet air temperature rather than noise. To use the same hysteresis level for both these experiments would not be sensible.

Another remark is the use of dead zones, instead of low-pass filters, to handle noise in many industrial controllers. The dead zone and controller need to be adjusted to each other. In the temperature control loop the initial controller had a dead zone of $0.5°$C, this was way too high for the rather fast PI controller obtained from the experiment and the system started to oscillate due to the dead zone. The dead zone was then adjusted to the typical noise levels, and the same controller showed no sign of oscillatory behavior in the experiment with the setpoint changes.

The temperature control experiments indicate that it would be valuable to add gain scheduling to deal with the nonlinearity in the heat exchanger. The temperature experiments also show the importance of evaluation from the operator. If the experiment with the higher relay amplitudes would have been performed without any evaluation, completely wrong model parameters would have been obtained. Now the nonlinearity in the system was discovered by the step responses, and other relay experiments gave very different model parameters. Of course the aim of the auto-tuner is that the results will be good at the first attempt, but with unknown load disturbances or nonlinearities that may not be the case, and some kind of test of the obtained controller performance is therefore necessary.

8

Conclusions and Future Work

In this thesis an autotuner for PID controllers based on an asymmetric relay feedback experiment has been proposed. Explanation and motivation to the different sequences in the automatic tuning procedure have been presented.

Some of the choices made by the autotuner are based on the process classification made by the normalized time delay, these choices are presented and motivated. A method of finding the normalized time delay from the asymmetric relay experiment was presented and evaluated. The results showed that classification could be done robustly from the obtained estimates.

How the experiment parameters should be chosen was evaluated, and the effect of insecurities and disturbances was discussed and evaluated. It showed that load disturbances as well as quantization in converters are problematic. So are nonlinearities in the actuators. These three issues have in common that they change the true asymmetry level of the relay function. If that is not known or discovered during the experiment, the resulting calculations will be inaccurate. This emphasizes the importance of evaluation of the obtained controller performance.

The proposed autotuner was implemented and tested both in a simulation environment and on an industrial process. A simple version of the autotuner was used, where FOTD or ITD models were obtained from the relay experiment. In the simulations, the experiment data was also used to estimate SOTD or IFOTD models from a parameter estimation method. For all the obtained models a PI or PID controller was tuned from the AMIGO rules. The evaluation gave satisfactory results, but also showed some possible improvements. One such improvement is to use a tuning method designed for the used experiment. Another is that better modeling and tuning for processes with small normalized time delays could increase the performance significantly. An advanced version of the autotuner could be developed to accomplish this.

How the advanced autotuner should be designed is subject to future research. A chirp signal, designed for the wanted frequency range, is an alternative for the possi-

ble additional experiment. Other model structures could be considered and a method to choose among these should then be part of the autotuner procedure. How the controller should be tuned in an advanced version is also subject to further research, but it would probably be optimization based if the computational power allows it. An advanced autotuner should ideally also design noise filtering, setpoint weighting and other aspects of the controller design. The possibility to use the models obtained from an advanced version of the autotuner in the design of model predictive controllers could be investigated.

Future research on the simple version of the autotuner should contain deriving its own simple tuning method. There are also process structures not present in the test batch that could be evaluated, for example unstable systems and systems with slow process zeros. A method to discover if the true asymmetry level differs from the supposed one would be of great value.

Another interesting subject for future research would be to extend the autotuner to multivariable systems.

Bibliography

Åström, K. J. and T. Hägglund (1984). "Automatic tuning of simple regulators with specifications on phase and amplitude margins". *Automatica* **20**:5, pp. 645–651.

Åström, K. J. and T. Hägglund (2006). *Advanced PID Control*. eng. ISA - The Instrumentation, Systems, and Automation Society; Research Triangle Park, NC 27709. ISBN: 978-1-55617-942-6.

Åström, K. J. and R. M. Murray (2008). *Feedback systems*. Princeton University Press, Princeton, NJ.

Desborough, L. and R. Miller (2002). "Increasing customer value of industrial control performance monitoring-Honeywell's experience". In: *AIChE Symp. Ser.* Pp. 169–189.

Friman, M. and K. V. Waller (1997). "A Two-Channel Relay for Autotuning". *Ind. Eng. Chem. Res.* **36**:7, pp. 2662–2671. DOI: 10.1021/ie970013u.

Garpinger, O. and T. Hägglund (2008). "A Software Tool for Robust PID Design". In: *Proc. 17th IFAC World Congr. Seoul, Korea*.

Hang, C., K. Åström, and W. Ho (1993). "Relay auto-tuning in the presence of static load disturbance". *Automatica* **29**:2, pp. 563–564. DOI: 10.1016/0005-1098(93)90159-Q.

Kaya, I. and D. Atherton (2001a). "Exact parameter estimation from. relay auto-tuning under static load disturbances". In: *Proc. 2001 Am. Control Conf. (Cat. No.01CH37148)*. Vol. 5. IEEE, pp. 3274–3279. DOI: 10.1109/ACC.2001.946427.

Kaya, I. and D. Atherton (2001b). "Parameter estimation from relay autotuning with asymmetric limit cycle data". *J. Process Control* **11**:4, pp. 429–439. DOI: 10.1016/S0959-1524(99)00073-6.

Kaya, I. (2006). "Parameter Estimation for Integrating Processes Using Relay Feedback Control under Static Load Disturbances". *Ind. Eng. Chem. Res.* **45**:13, pp. 4726–4731. DOI: 10.1021/ie060270b.

Khalil, H. K. (2000). *Nonlinear systems*. 3rd ed. Prentice Hall New Jersey.

Li, W., E. Eskinat, and W. L. Luyben (1991). "An improved autotune identification method". *Ind. Eng. Chem. Res.* **30**:7, pp. 1530–1541. DOI: 10.1021/ie00055a019.

Lin, C., Q.-G. Wang, and T. H. Lee (2004). "Relay Feedback: A Complete Analysis for First-Order Systems". *Ind. Eng. Chem. Res.* **43**:26, pp. 8400–8402. DOI: 10.1021/ie034043a.

Liu, T. and F. Gao (2008). "Alternative Identification Algorithms for Obtaining a First-Order Stable/Unstable Process Model from a Single Relay Feedback Test". *Ind. Eng. Chem. Res.* **47**:4, pp. 1140–1149. DOI: 10.1021/ie070856d.

Liu, T., Q.-G. Wang, and H.-P. Huang (2013). "A tutorial review on process identification from step or relay feedback test". *J. Process Control* **23**:10, pp. 1597–1623. DOI: 10.1016/j.jprocont.2013.08.003.

Ljung, L. (1999). *System Identification - Theory For the User*. 2nd ed. PTR Prentice Hall, Upper Saddle River, N.J.

Luyben, W. L. (1987). "Derivation of transfer functions for highly nonlinear distillation columns". *Ind. Eng. Chem. Res.* **26**:12, pp. 2490–2495. DOI: 10.1021/ie00072a017.

Luyben, W. L. (2001). "Getting More Information from Relay-Feedback Tests". *Ind. Eng. Chem. Res.* **40**:20, pp. 4391–4402. DOI: 10.1021/ie010142h.

Park, J. H., S. W. Sung, and I.-B. Lee (1997). "Improved relay auto-tuning with static load disturbance". *Automatica* **33**:4, pp. 711–715. DOI: 10.1016/S0005-1098(96)00174-4.

Sell, N. J. (1995). *Process control fundamentals for the pulp and paper industry*. Technical Association of the Pulp & Paper Industry.

Shen, S.-H., J.-S. Wu, and C.-C. Yu (1996a). "Autotune Identification under Load Disturbance". *Ind. Eng. Chem. Res.* **35**:5, pp. 1642–1651. DOI: 10.1021/ie950480g.

Shen, S.-H., J.-S. Wu, and C.-C. Yu (1996b). "Use of biased-relay feedback for system identification". *AIChE J.* **42**:4, pp. 1174–1180. DOI: 10.1002/aic.690420431.

Skogestad, S. (2003). "Simple analytic rules for model reduction and PID controller tuning". *J. Process Control* **13**:4, pp. 291–309. DOI: 10.1016/S0959-1524(02)00062-8.

Skogestad, S. (2006). "Tuning for smooth PID control with acceptable disturbance rejection". *Ind. & Eng. Chem. Res.* **45**:23, pp. 7817–7822.

Sung, S. W. and J. Lee (2006). "Relay feedback method under large static disturbances". *Automatica* **42**:2, pp. 353–356. DOI: 10.1016/j.automatica.2005.10.001.

Theorin, A. and J. Berner (2015). "Implementation of an Asymmetric Relay Auto-tuner in a Sequential Control Language". In: *IEEE Int. Conf. Autom. Sci. Eng.* Submitted.

Vinnicombe, G. (2001). *Uncertainty and Feedback: H∞ loop-shaping and the ν-gap metric*. Imperial Collage Press.

Wang, Q.-G., C.-C. Hang, and Q. Bi (1997a). "Process frequency response estimation from relay feedback". *Control Eng. Pract.* **5**:9, pp. 1293–1302. DOI: 10.1016/S0967-0661(97)84368-7.

Wang, Q.-G., C.-C. Hang, and B. Zou (1997b). "Low-Order Modeling from Relay Feedback". *Ind. Eng. Chem. Res.* **36**:2, pp. 375–381. DOI: 10.1021/ie960412+.

Ziegler, J. and N. Nichols (1942). "Optimum Settings for Automatic Controllers". *trans. ASME* **64**:11.

A

Derivation of Equations

A.1 FOTD Model

An FOTD model on the form

$$P(s) = \frac{K_p}{1+sT}e^{-sL} \tag{A.1}$$

can be written on state space form as

$$\begin{aligned}\dot{x}(t) &= ax(t) + bu(t-L)\\ y(t) &= x(t)\end{aligned} \tag{A.2}$$

with $a = -1/T$ and $b = K_p/T$. The solution to this is

$$y(t) = e^{a(t-t_0)}y(t_0) + \int_{t_0}^{t} be^{a(t-\theta)}u(\theta - L)d\theta. \tag{A.3}$$

In [Lin et al., 2004] an analysis of the existence and stability of limit cycles for FOTD systems under relay feedback has been performed. Assuming that these conditions are satisfied, the equations for an FOTD system can be derived in the following manner.

Let t_1, t_2 and t_3 denote three consecutive time points where the relay switches. Without any loss of generality we can assume that before time t_1 the control signal from the relay was u_{on} and then at time t_1 the relay switched to u_{off}. According to the definitions in Section 3.1 this implies that

$$\begin{aligned}y(t_1) &= y_0 + h\\ y(t_2) &= y_0 - h\\ y(t_3) &= y_0 + h\end{aligned} \tag{A.4}$$

and that

$$\begin{aligned}t_{\text{off}} &= t_2 - t_1\\ t_{\text{on}} &= t_3 - t_2.\end{aligned} \tag{A.5}$$

In the interval $[t_1, t_1 + L]$ the control signal is constant, $u(t - L) = u_{on}$. This gives that

$$
\begin{aligned}
y(t_1 + L) &= e^{a(t_1 + L - t_1)} y(t_1) + \int_{t_1}^{t_1 + L} b e^{a(t_1 + L - \theta)} u(\theta - L) d\theta \\
&= e^{aL} y(t_1) + b u_{on} \int_{t_1}^{t_1 + L} e^{a(t_1 + L - \theta)} d\theta \\
&= e^{aL} y(t_1) + \frac{b u_{on}}{-a} \left(1 - e^{aL}\right) \\
&= e^{aL} \left(y(t_1) + \frac{b u_{on}}{a}\right) - \frac{b u_{on}}{a}.
\end{aligned}
\tag{A.6}
$$

In the interval $[t_1 + L, t_2]$ we have $u(t - L) = u_{off}$. This gives

$$
\begin{aligned}
y(t_2) &= e^{a(t_2 - (t_1 + L))} y(t_1 + L) - \frac{b u_{off}}{a} \left(1 - e^{a(t_2 - (t_1 + L))}\right) \\
&= e^{a((t_2 - t_1) - L)} \left(y(t_1 + L) + \frac{b u_{off}}{a}\right) - \frac{b u_{off}}{a} \\
&= e^{a(t_2 - t_1)} e^{-aL} \left(\left[e^{aL} \left(y(t_1) + \frac{b u_{on}}{a}\right) - \frac{b u_{on}}{a} \right] + \frac{b u_{off}}{a} \right) - \frac{b u_{off}}{a} \\
&= e^{a(t_2 - t_1)} \left(\left(y(t_1) + \frac{b u_{on}}{a}\right) + e^{-aL} \frac{b}{a} (u_{off} - u_{on}) \right) - \frac{b u_{off}}{a}.
\end{aligned}
\tag{A.7}
$$

The same arguments give $y(t_2 + L)$ and $y(t_3)$. By transforming from (a, b) to (K_p, L) the equations can be summarized by

$$
y(t_1) = y_0 + h
\tag{A.8}
$$

$$
y(t_1 + L) = e^{-L/T} \left(y(t_1) - K_p u_{on}\right) + K_p u_{on}
\tag{A.9}
$$

$$
y(t_2 + L) = e^{-L/T} \left(y(t_2) - K_p u_{off}\right) + K_p u_{off}
\tag{A.10}
$$

$$
y(t_2) = e^{-(t_2 - t_1)/T} \left(y(t_1) - K_p u_{on} - K_p e^{L/T} (u_{off} - u_{on})\right) + K_p u_{off}
\tag{A.11}
$$

$$
y(t_3) = e^{-(t_3 - t_2)/T} \left(y(t_2) - K_p u_{off} - K_p e^{L/T} (u_{on} - u_{off})\right) + K_p u_{on}
\tag{A.12}
$$

The values of y at the switching points are given by (A.4), and the time intervals t_{on} and t_{off} are given by (A.5). The equations (A.11) and (A.12) can then be reformulated to give the time intervals t_{on} and t_{off} as a function of the process parameters K_p, T and L.

$$
t_{off} = t_2 - t_1 = T \ln \left(\frac{y(t_1) - K_p u_{on} - K_p e^{L/T} (u_{off} - u_{on})}{y(t_2) - K_p u_{off}} \right)
\tag{A.13}
$$

$$
t_{on} = t_3 - t_2 = T \ln \left(\frac{y(t_2) - K_p u_{off} + K_p e^{L/T} (u_{off} - u_{on})}{y(t_3) - K_p u_{on}} \right)
\tag{A.14}
$$

Inserting the y-values from (A.4), the definitions for u_{on}, u_{off}, and using that $y_0 = K_p u_0$ in the stationary equilibrium point, the equations (A.13) and (A.14) can be rewritten as

$$t_{\text{off}} = T \ln \left(\frac{h/|K_p| - d_1 + e^{L/T}(d_1 + d_2)}{d_2 - h/|K_p|} \right) \tag{A.15}$$

$$t_{\text{on}} = T \ln \left(\frac{h/|K_p| - d_2 + e^{L/T}(d_1 + d_2)}{d_1 - h/|K_p|} \right). \tag{A.16}$$

Limits for the hysteresis-free equations

To get some insight we investigate the limits of t_{on} and t_{off} as the ratio $L/T \to 0$ and $L/T \to \infty$ respectively. To simplify the calculations the hysteresis-free situation, $h = 0$, is considered. The equations for t_{on} and t_{off} then reduces to

$$t_{\text{on}} = T \ln \left(\frac{-d_2 + e^{L/T}(d_1 + d_2)}{d_1} \right),$$
$$t_{\text{off}} = T \ln \left(\frac{-d_1 + e^{L/T}(d_1 + d_2)}{d_2} \right). \tag{A.17}$$

Let's consider the limit as $L/T \to 0$. If we would enter $L/T = 0$ directly into the equations we would get $T \ln(1)$. Since $L/T \to 0$ implies that T may be infinite, and $\ln(1) = 0$, this doesn't tell us much. Instead we use that $e^{L/T} \approx 1 + L/T$ for small L/T. The equations can then be rewritten to

$$t_{\text{on}} \approx T \ln \left(\frac{-d_2 + (1 + L/T)(d_1 + d_2)}{d_1} \right),$$
$$t_{\text{off}} \approx T \ln \left(\frac{-d_1 + (1 + L/T)(d_1 + d_2)}{d_2} \right). \tag{A.18}$$

Simplifying these equations gives

$$t_{\text{on}} \approx T \ln \left(1 + L/T \left(\frac{d_2}{d_1} + 1 \right) \right),$$
$$t_{\text{off}} \approx T \ln \left(1 + L/T \left(\frac{d_1}{d_2} + 1 \right) \right). \tag{A.19}$$

Since $\ln(1 + x) \approx x$ for small x, this gives

$$t_{\text{on}} \approx L \left(\frac{d_2}{d_1} + 1 \right),$$
$$t_{\text{off}} \approx L \left(\frac{d_1}{d_2} + 1 \right). \tag{A.20}$$

To conclude, as $L/T \to 0$, the time intervals t_{on} and t_{off} and their ratio are

$$t_{\text{on}} = L\left(\frac{d_2}{d_1} + 1\right), \qquad L/T \to 0$$

$$t_{\text{off}} = L\left(\frac{d_1}{d_2} + 1\right), \qquad L/T \to 0 \qquad \text{(A.21)}$$

$$\frac{t_{\text{off}}}{t_{\text{on}}} = \frac{d_2}{d_1}, \qquad L/T \to 0.$$

Now consider the other limit, when $L/T \to \infty$. The equations for t_{on} and t_{off} are still

$$t_{\text{on}} = T \ln\left(\frac{-d_2 + e^{L/T}(d_1 + d_2)}{d_1}\right),$$

$$\qquad \text{(A.22)}$$

$$t_{\text{off}} = T \ln\left(\frac{-d_1 + e^{L/T}(d_1 + d_2)}{d_2}\right).$$

Rewriting these equations as

$$t_{\text{on}} = T \ln\left(e^{L/T}\frac{-d_2 e^{-L/T} + d_1 + d_2}{d_1}\right),$$

$$\qquad \text{(A.23)}$$

$$t_{\text{off}} = T \ln\left(e^{L/T}\frac{-d_1 e^{-L/T} + d_1 + d_2}{d_2}\right),$$

and using that $\ln(xy) = \ln(x) + \ln(y)$ we get

$$t_{\text{on}} = T\left(L/T + \ln\left(\frac{-d_2 e^{-L/T} + d_1 + d_2}{d_1}\right)\right),$$

$$\qquad \text{(A.24)}$$

$$t_{\text{off}} = T\left(L/T + \ln\left(\frac{-d_1 e^{-L/T} + d_1 + d_2}{d_2}\right)\right).$$

Simplifying these equations and inserting $L/T \to \infty$ in the exponentials, we get

$$t_{\text{on}} = L + T \ln\left(\frac{d_1 + d_2}{d_1}\right),$$

$$\qquad \text{(A.25)}$$

$$t_{\text{off}} = L + T \ln\left(\frac{d_1 + d_2}{d_2}\right).$$

Since $L/T \to \infty$ implies that $L \gg T$, the equations simplify to

$$t_{\text{on}} \approx L,$$

$$\qquad \text{(A.26)}$$

$$t_{\text{off}} \approx L.$$

In conclusion, in the limit $L/T \to \infty$ we get

$$
\begin{aligned}
t_{\mathrm{on}} &= L, & L/T \to \infty, \\
t_{\mathrm{off}} &= L, & L/T \to \infty, \\
\frac{t_{\mathrm{off}}}{t_{\mathrm{on}}} &= 1, & L/T \to \infty.
\end{aligned}
\tag{A.27}
$$

A.2   ITD Model

As in the derivation of the FOTD model parameters in Section A.1 we let t_1, t_2 and t_3 denote three consecutive time points where the relay switches. We also assume that before time t_1 the control signal from the relay was u_{on} and then at time t_1 the relay switched to u_{off}, so (A.4) and (A.5) are still valid.

An integrating process with the transfer function

$$
P(s) = \frac{k_v}{s} e^{-sL}
\tag{A.28}
$$

can be written as the differential equation

$$
\dot{y}(t) = k_v u(t - L).
\tag{A.29}
$$

From this equation it is clear that the only way to have a stationary point (u_0, y_0) is if the control signal $u_0 = 0$. The equation also implies that the shape of y is triangular, since $u(t)$ is piecewise constant. This gives the following equations

$$
\begin{aligned}
y(t_1) &= y_0 + h, & \text{(A.30)} \\
y(t_1 + L) &= y(t_1) + k_v L u_{\mathrm{on}}, & \text{(A.31)} \\
y(t_2) &= y(t_1 + L) + k_v(t_2 - t_1 - L)u_{\mathrm{off}} = y_0 - h, & \text{(A.32)} \\
y(t_2 + L) &= y(t_2) + k_v L u_{\mathrm{off}}, & \text{(A.33)} \\
y(t_3) &= y(t_2 + L) + k_v(t_3 - t_2 - L)u_{\mathrm{on}} = y_0 + h. & \text{(A.34)}
\end{aligned}
$$

The equations for t_{on} and t_{off} are then given by

$$
t_{\mathrm{off}} = t_2 - t_1 = \frac{k_v L(u_{\mathrm{off}} - u_{\mathrm{on}}) - 2h}{k_v u_{\mathrm{off}}},
\tag{A.35}
$$

$$
t_{\mathrm{on}} = t_3 - t_2 = \frac{k_v L(u_{\mathrm{on}} - u_{\mathrm{off}}) + 2h}{k_v u_{\mathrm{on}}},
\tag{A.36}
$$

which gives the ratio

$$
\frac{t_{\mathrm{off}}}{t_{\mathrm{on}}} = -\frac{u_{\mathrm{on}}}{u_{\mathrm{off}}}.
\tag{A.37}
$$

The process parameters k_v and L can not be determined from this equation. To get an additional equation we investigate the integral I_y.
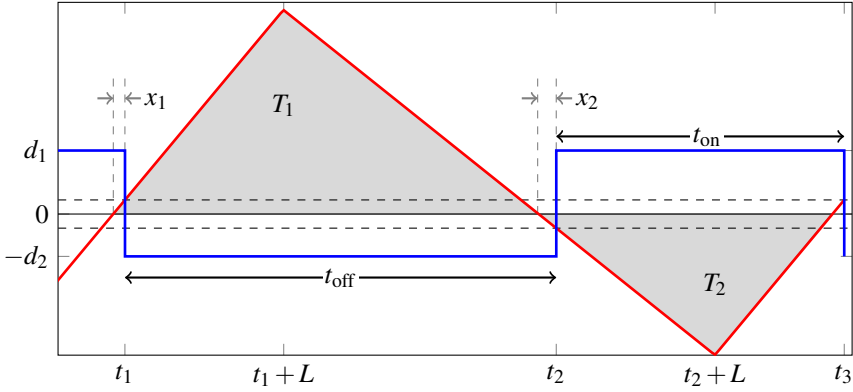
Figure A.1   Output from a relay experiment with an ITD process. The blue line shows the relay output u while the red line shows the process output y. The time intervals t_{on} and t_{off} as well as the time intervals x_1 and x_2 are denoted in the figure. The dashed black lines show the hysteresis. The gray area shows the process output integral I_y, consisting of the two triangles T_1 and T_2.

Denoting $A_u := |y(t_1 + L) - y_0|$ and $A_d := |y(t_2 + L) - y_0|$, the integral of the process output, I_y, can be calculated as the difference of the two triangles shown in Figure A.1. The triangles are given by

$$T_1 = \frac{A_u(t_{off} + x_1 - x_2)}{2}, \tag{A.38}$$

$$T_2 = \frac{A_d(t_{on} + x_2 - x_1)}{2}. \tag{A.39}$$

The time intervals x_1 and x_2 are given from the similarity of the triangles. They can be expressed either as

$$x_1 = \frac{Lh}{A_u - h}, \qquad x_2 = \frac{Lh}{A_d - h}, \tag{A.40}$$

or as

$$x_1 = h\left(\frac{t_{on} - L}{A_d + h}\right), \qquad x_2 = h\left(\frac{t_{off} - L}{A_u + h}\right). \tag{A.41}$$

From (A.40) and (A.41) expressions for the products between the top amplitudes and x_1 and x_2 are given by

$$\begin{aligned} A_u x_1 &= h(L + x_1) & A_u x_2 &= h(t_{off} - L - x_2) \\ A_d x_1 &= h(t_{on} - L - x_1) & A_d x_2 &= h(L + x_2). \end{aligned} \tag{A.42}$$

87

Appendix A. Derivation of Equations

Inserting these expressions in the equations for T_1 and T_2 gives

$$
\begin{aligned}
T_1 &= \frac{1}{2}\left(A_u t_{\text{off}} + h(L + x_1) - h(t_{\text{off}} - L - x_2)\right) \\
&= \frac{1}{2}\left(t_{\text{off}}(A_u - h) + 2hL + hx_1 + hx_2\right)
\end{aligned}
\tag{A.43}
$$

$$
\begin{aligned}
T_2 &= \frac{1}{2}\left(A_d t_{\text{on}} + h(L + x_2) - h(t_{\text{on}} - L - x_1)\right) \\
&= \frac{1}{2}\left(t_{\text{on}}(A_d - h) + 2hL + hx_2 + hx_1\right)
\end{aligned}
\tag{A.44}
$$

This gives the integral

$$
I_y = T_1 - T_2 = \frac{1}{2}\left(t_{\text{off}}(A_u - h) - t_{\text{on}}(A_d - h)\right).
\tag{A.45}
$$

Since A_u and A_d are given by

$$
A_u = |y(t_1 + L) - y_0| = h + k_v L u_{\text{on}},
\tag{A.46}
$$
$$
A_d = |y(t_2 + L) - y_0| = h - k_v L u_{\text{off}},
\tag{A.47}
$$

equation (A.45) can be rewritten as

$$
I_y = \frac{1}{2}\left(t_{\text{off}}k_v L u_{\text{on}} - t_{\text{on}}k_v L u_{\text{off}}\right) = \frac{k_v L}{2}\left(t_{\text{off}}u_{\text{on}} - t_{\text{on}}u_{\text{off}}\right).
\tag{A.48}
$$

This could easily have been found if the hysteresis would have been zero, but is now proven also when there is hysteresis.

Using (A.48) and either of (A.35) or (A.36) equations for k_v and L can be found. From (A.36) we get

$$
L = \frac{u_{\text{on}}t_{\text{on}} - 2h/k_v}{u_{\text{on}} - u_{\text{off}}},
\tag{A.49}
$$

and from (A.48) we get

$$
k_v = \frac{2I_y}{L\left(t_{\text{off}}u_{\text{on}} - t_{\text{on}}u_{\text{off}}\right)}.
\tag{A.50}
$$

Inserting (A.49) in (A.50) gives

$$
\begin{aligned}
k_v &= \frac{2I_y(u_{\text{on}} - u_{\text{off}})}{(u_{\text{on}}t_{\text{on}} - 2h/k_v)(t_{\text{off}}u_{\text{on}} + t_{\text{on}}u_{\text{off}})} \Rightarrow \\
k_v u_{\text{on}}t_{\text{on}} - 2h &= \frac{2I_y(u_{\text{on}} - u_{\text{off}})}{t_{\text{off}}u_{\text{on}} + t_{\text{on}}u_{\text{off}}} \Rightarrow \\
k_v &= \frac{2I_y(u_{\text{on}} - u_{\text{off}})}{(t_{\text{off}}u_{\text{on}} + t_{\text{on}}u_{\text{off}})u_{\text{on}}t_{\text{on}}} + \frac{2h}{u_{\text{on}}t_{\text{on}}} \Rightarrow \\
k_v &= \frac{2I_y(u_{\text{on}} - u_{\text{off}})}{t_{\text{off}}u_{\text{on}}^2 t_{\text{on}} + t_{\text{on}}^2 u_{\text{off}}u_{\text{on}}} + \frac{2h}{u_{\text{on}}t_{\text{on}}} \Rightarrow \\
\left[t_{\text{on}} \right. &\left. = -\frac{u_{\text{off}}t_{\text{off}}}{u_{\text{on}}}(\text{A.37}) \right] \Rightarrow \\
k_v &= \frac{2I_y(u_{\text{on}} - u_{\text{off}})}{t_{\text{off}}u_{\text{on}}^2 t_{\text{on}} - t_{\text{off}}u_{\text{off}}^2 t_{\text{on}}} + \frac{2h}{u_{\text{on}}t_{\text{on}}} \Rightarrow \\
k_v &= \frac{2I_y(u_{\text{on}} - u_{\text{off}})}{t_{\text{off}}t_{\text{on}}(u_{\text{on}}^2 - u_{\text{off}}^2)} + \frac{2h}{u_{\text{on}}t_{\text{on}}} \Rightarrow \\
k_v &= \frac{2I_y}{t_{\text{off}}t_{\text{on}}(u_{\text{on}} + u_{\text{off}})} + \frac{2h}{u_{\text{on}}t_{\text{on}}}.
\end{aligned}
\tag{A.51}
$$

To conclude, the parameters for the ITD model can be found by measuring the time intervals t_{on} and t_{off} as well as the integral I_y. The parameters are then given by

$$
k_v = \frac{2I_y}{t_{\text{on}}t_{\text{off}}(u_{\text{on}} + u_{\text{off}})} + \frac{2h}{u_{\text{on}}t_{\text{on}}},
\tag{A.52}
$$

$$
L = \frac{u_{\text{on}}t_{\text{on}} - 2h/k_v}{u_{\text{on}} - u_{\text{off}}}.
\tag{A.53}
$$

B

The Test Batch

The test batch used is the one described in [Åström and Hägglund, 2006]. The processes in the batch are representative for many of the processes encountered in the process industry. The batch contains both integrating, lag-dominant and delay-dominant processes. In total the batch consists of 134 processes, divided into nine different process types. All the processes included in the batch are listed below.

$$P_1(s) = \frac{e^{-s}}{1+sT},$$
$$T = 0.02,\ 0.05,\ 0.1,\ 0.2,\ 0.3,\ 0.5,\ 0.7,\ 1,\ 1.3,\ 1.5,\ 2,$$
$$4,\ 6,\ 8,\ 10,\ 20,\ 50,\ 100,\ 200,\ 500,\ 1000$$

$$P_2(s) = \frac{e^{-s}}{(1+sT)^2},$$
$$T = 0.01,\ 0.02,\ 0.05,\ 0.1,\ 0.2,\ 0.3,\ 0.5,\ 0.7,\ 1,\ 1.3,\ 1.5,$$
$$2,\ 4,\ 6,\ 8,\ 10,\ 20,\ 50,\ 100,\ 200,\ 500$$

$$P_3(s) = \frac{1}{(s+1)(1+sT)^2},$$
$$T = 0.005,\ 0.01,\ 0.02,\ 0.05,\ 0.1,\ 0.2,\ 0.5,\ 2,\ 5,\ 10$$

$$P_4(s) = \frac{1}{(s+1)^n},$$
$$n = 3,\ 4,\ 5,\ 6,\ 7,\ 8$$

$$P_5(s) = \frac{1}{(1+s)(1+\alpha s)(1+\alpha^2 s)(1+\alpha^3 s)}$$
$$\alpha = 0.1,\ 0.2,\ 0.3,\ 0.4,\ 0.5,\ 0.6,\ 0.7,\ 0.8,\ 0.9$$

$$P_6(s) = \frac{1}{s(1+sT_1)} e^{-sL_1}, \quad T_1 + L_1 = 1$$
$$L_1 = 0.01, 0.02, 0.05, 0.1, 0.2, 0.3, 0.5, 0.7, 0.9, 1.0$$

$$P_7(s) = \frac{1}{(1+sT)(1+sT_1)} e^{-sL_1}, \quad T_1 + L_1 = 1$$
$$T = 1, 2, 5, 10 \quad L_1 = 0.01, 0.02, 0.05, 0.1, 0.3, 0.5, 0.7, 0.9, 1.0$$

$$P_8(s) = \frac{1 - \alpha s}{(s+1)^3},$$
$$\alpha = 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1$$

$$P_9(s) = \frac{1}{(s+1)((sT)^2 + 1.4sT + 1)},$$
$$T = 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0.$$

C

Default Parameters

This section lists the parameters used for the relay experiment if nothing else is stated. Most of the parameters are discussed and explained in Chapter 5. Some of the parameters are also discussed in the definitions of the asymmetric relay function in Section 3.1. Some parameters may though need some further explanation and comments. The *Large process value* simply decides whether it is desired to have a larger process deviation up or down. The *Ramp up time* is the time it takes for the relay amplitude to ramp up to the maximum amplitude during the startup. The resolution of the A/D and D/A converters are set to a very high value as default to be able to ignore their existence, they will never be so high in reality.

Explanation	Notation	Value
Asymmetry level	γ	2
Bits in D/A converter		50
Bits in A/D converter		50
Convergence limit	ε	0.01
Hysteresis	h	$2n_0$
Hysteresis noise-free	h	0.1
Large process value		Up
Maximal control signal deviation	u_{maxdev}	10
Maximal process deviation	y_{maxdev}	$12h$
Minimal process deviation	y_{mindev}	$2h$
Noise level	n_0	0
Noise measurement time [s]		10
Ramp up time [s]		5
Sample time [s]	t_s	0.01