Automatic Controller Tuning using Relay-based Model Identification

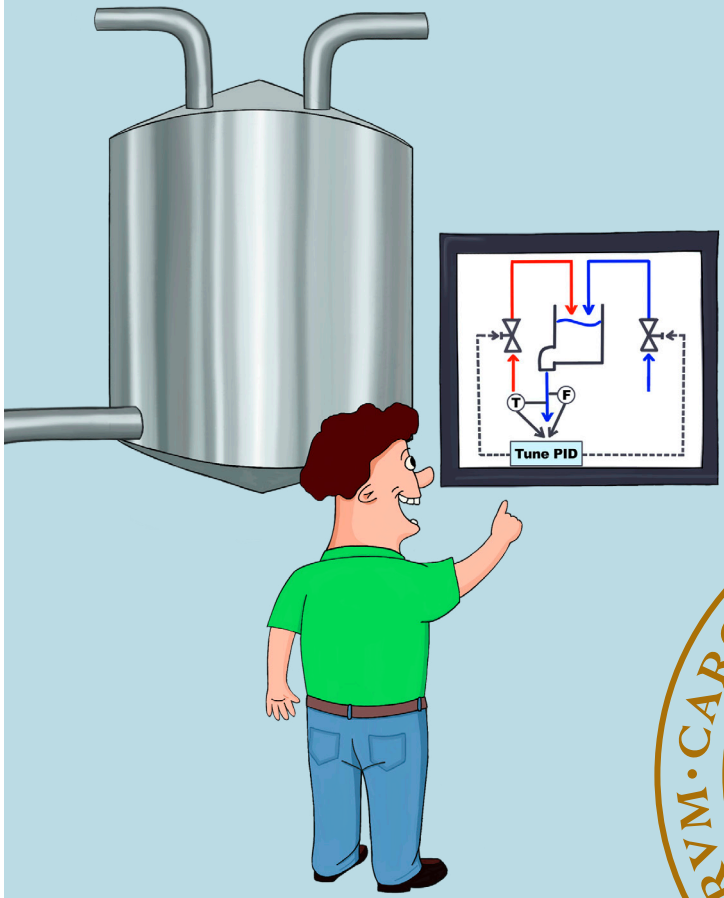
Berner, Josefin

2017

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

Automatic Controller Tuning using Relay-based Model Identification

JOSEFIN BERNER
DEPARTMENT OF AUTOMATIC CONTROL | LUND UNIVERSITY



Automatic Controller Tuning using Relay-based Model Identification

Josefin Berner

LUND
UNIVERSITY

Department of Automatic Control

PhD Thesis TFRT-1118
ISBN 978-91-7753-446-4 (print)
ISBN 978-91-7753-447-1 (web)
ISSN 0280–5316

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

Printed in Sweden by MediaTryck.
Lund 2017

Abstract

Proportional-integral-derivative (PID) controllers are very common in the process industry. In a regular factory there may be hundreds or thousands of them in use. Each of these controllers needs to be tuned, and even though the PID controller is simple, tuning the controllers still requires several hours of work and adequate knowledge in order to achieve a desired performance. Because of that, many of the operating PID controllers today are poorly tuned or even running in manual mode. Methods for tuning the controllers in an automated fashion are therefore highly beneficial, and the relay autotuner, that was introduced on the market in the 1980's, has been listed as one of the great success stories of control.

The technology development since the 1980's, both concerning PID control and available computing power, gives opportunities for improvements of the autotuner. In this thesis three new autotuners are presented. They are all based on asymmetric relay feedback tests, providing process excitation at the frequency intervals relevant for PID control. One of the proposed autotuners is similar to the classic relay autotuner, but provides low-order models from which the controllers are tuned by simple formulas. The second autotuner uses the data from a very short relay test as input to an optimization method. This method provides more accurate model estimations, but to the cost of more computing. The controller is then tuned by another optimization method, using the estimated model as input. The principle of the third autotuner is similar to the second one, but it is used to tune multivariable PID controllers for interacting processes. In this case a relay feedback experiment is performed on all loops simultaneously, and the data is used to identify the process transfer function matrix. All of the proposed autotuners strive to be user-friendly and practically applicable.

Evaluation of the three autotuning strategies are done both through simulation examples and on experimental processes. The developed autotuners are also compared to commercially available ones, and the study shows that an upgrade of the industry standard to the newly available autotuners will yield a significant performance improvement.

Acknowledgments

First of all I want to thank everyone at the Department of Automatic Control for providing a really inspiring and delightful working environment. My supervisors Tore Hägglund, Karl Johan Åström and Kristian Soltesz are worth a lot of thanks. It is every PhD student's dream to have a supervisor that is truly interested in her/his research, enthusiastic and available for questions and discussions. I do not know what I have done to deserve not just one, but three such supervisors. You have always encouraged me, no matter if I wanted to do something different in a method or paper, wanted to write a fairytale about my research, or wanted to leave work for a couple of months to go to the other side of the world and help out with the local kids. Your support in these, sometimes rather odd, ideas have meant a lot to me.

Martin, Olof and Andreas, thanks for head-hunting me to the best office! You made the first three years of my PhD studies a really fun time, and I still think you finished your theses way too early. Luckily for me, Marcus TA, Giulia, Morten and Marcus G, stepped in, filled up the empty spots and continued to make it fun to come to the office every morning (or lunch), thanks a lot for that!

A lot of thanks to the administrative staff, Eva, Ingrid, Mika, Cecilia and Monika. Not only for your amazing work and help with all kind of work-related things, but mainly for the joy and good spirit you all bring to this department. It would not be the same without you and I hope you know how appreciated you are. The same goes for the technical staff, Anders x 2, Pontus and Leif. Thanks for always helping out when a computer stops working, a lab process is broken or a latex document won't compile.

Tove Sörnmo, thanks for illustrating my fictive control engineer Kontroll-Kalle. Even though I hope that some people will read and enjoy at least parts of this thesis, which improved by appreciated help from Tommi and Gustav with comments and corrections, I do believe that Kontroll-Kalle will continue to be my most downloaded, used and appreciated publication.

Richard, thanks for trying to invade the fortress of Carcassonne with me! Carolina, thanks for excellent cooperation in the last years of Flickor på Teknis. Fredrik bänk-Bagge, thanks for introducing the concept of Bänktorsdag. Gautham, thanks for bringing food from the Indian lady and for organizing a lot of boardgame nights.

Michelle, thanks for introducing me to the massage-chair at Actic. Thanks Gustav for keeping us all up-to-date with the latest gossip. Thanks to Christian for organizing betting each major football championship, and to Mariette for doing the same for Sommar i P1.

Even though I really like my workplace, not everything in life is connected to work. All my life, sports has occupied most of my spare time, and here in Lund I've been lucky enough to find two of the absolute best clubs and teams I have ever been a part of, my football team in Hallands Nations FF and my handball team in Botulfsplatsens BK. We are not winning every game, far from it, but the team spirit, respect and joy that we share make me proud of calling you my team.

Linnea and Madeleine, thanks for being amazing! The two of you have made my life so much better for so many years now, and I'm looking forward to doing many more fun/crazy/important things with you in the future.

Teresia, thanks for not only being a good friend and an inspiring person, but also for reminding me time after time how lucky I am to do my PhD at such a nice and well-functioning department. I wish that would be the situation for everyone! I also want to thank you and Ellinor for our very irregular TV-series nights every couple of months when all three of us happen to have a free evening at the same time. Maybe it will be easier now?

And to my other friends, I will not mention you all, I'm already failing to keep this short, but take some pages of this thesis and consider them as your contribution, because I would not have made this without you.

Apart from my lovely friends I am also fortunate enough to have an amazing family. All the way from my grandparents who all inspired me a lot, to my nieces and nephews who let me bring out my inner child in a legitimate way. Mum, dad, Anna, Karin and Ellen, as my family you have been a constant source of inspiration (and frustration) for me. I am happy to have grown up in your company, you have all been a part of shaping me into the person I am today, you are always there if I need you, and I love you all!

And to Tommi, kämpa på himmelsblå!

Financial support

The following are acknowledged for financial support: The Swedish Research Council through the LCCC Linnaeus Center, and the ELLIIT Excellence Center.

Contents

Nomenclature

Notation	Description
$C(s)$	Controller transfer function
γ	Asymmetry level of the relay
d	Load disturbance
d_1	Positive relay amplitude
d_2	Negative relay amplitude
e	Control error
$F(s)$	Filter transfer function
h	Hysteresis of the relay
K	Proportional gain of PID controller
k_c	Critical gain
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	Measurement noise
$P(s)$	Process (model) transfer function
ρ	Half-period ratio
r	Reference value, setpoint
T	Time constant of process
T_d	Derivative time of PID controller
T_f	Filter time constant
T_i	Integral time of PID controller
τ	Normalized time delay
u	Control signal, relay output
ω_c	Critical frequency
y	Process output

Abbreviation	**Description**
DFA	Describing function approximation
ECA	An industrial autotuner from ABB, full name ABB ECA600
FFT	Fast Fourier Transform
FOTD	First-order time-delayed (model)
IAE	Integrated absolute error
IE	Integrated error
ITD	Integrating time-delayed (model)
IFOTD	Integrating plus first-order time-delayed (model)
MIMO	Multiple-input multiple-output (system)
MPC	Model predictive control
NOMAD	Noise-robust optimization-based modeling and design (autotuner)
PID	Proportional integral derivative (controller)
SISO	Single-input single-output (system)
SOTD	Second-order time-delayed (model)
TITO	Two-input two-output (system)
τ-tuner	An autotuning procedure using τ

1

Introduction

1.1 Motivation

PID control has been the backbone of the process industry for many decades. Despite its success, many control loops are still run in manual mode or perform poorly due to bad tuning [Desborough and Miller, 2002]. Since the introduction of the relay autotuner [Åström and Hägglund, 1984] in the 1980's the situation has improved, but still the number of poorly performing controllers is significant. The relay autotuner has the benefits of being fast, simple and not requiring any a priori information. However, the early autotuner was restricted to use simple design rules due to computational limitations at the time. With those computational limitations long gone, and some increased insight gained in controller tuning, the time has come to revisit the relay autotuner to see if it can be improved.

In many applications the process variables interact with each other. Hence there is also a desire to investigate if the relay autotuner could be extended to handle multivariable systems. Some previous research has been addressing this problem, but to the author's knowledge the feature is not provided in commercial control systems today.

1.2 Aim of Thesis

The work in this thesis explores the possibilities of developing improved PID autotuners, using the last decades' advances in controller tuning and computing power. The aim is to maintain the benefits of the classic relay autotuner, such as it being simple to use, fast, and automatically exciting the process in a frequency interval relevant for PID control. The work also aims to have a strong focus on practical usage of the autotuner; developed procedures should not only be applicable for academic simulation examples.

1.3 Contents and Contributions of the Thesis

The main contributions of this thesis are three new versions of relay autotuners. The autotuning procedures are developed, described and evaluated in the papers. The evaluations have been made through simulations, tests on laboratory equipment and by industrial experiments. The thesis contains four chapters and six papers. This section gives an overview of the chapters and describes the contributions of each paper as well as what role the author had in the different papers.

Chapter 1 - Introduction

In this introductory chapter the scope of the thesis is presented. A motivation to the research problem, the contributions of the thesis, and delimitations of the work are presented.

Chapter 2 - Background

The background chapter provides definitions and descriptions of some of the relevant concepts for this thesis. This chapter is completely based on previous knowledge and is included to give a common ground to the readers. It also gives the necessary background to put the findings of the thesis into a historical context. Parts of the background chapter are reused from the background chapter in the author's licentiate thesis [Berner, 2015].

Chapter 3 - Three versions of the autotuner

This chapter describes the work presented in this thesis. It relates the results in the different papers to each other and to work done by others. This chapter motivates the need of the different autotuners proposed in the thesis, and summarizes their functionality, benefits and drawbacks. The chapter also gives examples of when the proposed autotuners have been used in different settings, and discusses some of the obtained results.

Chapter 4 - Future work

Here some additional ideas, which have not yet been investigated, are listed as possibilities for future work.

Paper I

Berner, J., T. Hägglund, and K. J. Åström (2016). "Improved Relay Autotuning using Normalized Time Delay". In: *2016 American Control Conference (ACC)*. IEEE, pp. 1869–1875.

In this paper it is shown that the normalized time delay has an important role in the model and controller selections made in the relay autotuner. A simple way of finding the normalized time delay from an asymmetric relay experiment is provided.

The proposed autotuner obtains first-order models with time delay of the process. These models are obtained from analytic equations including a few properties measured from the experiment. The used properties are robust to noise. The model and controller selections, based on the estimated normalized time delay, are evaluated through simulations. The paper also shows some experimental results from an industrial testing facility at Schneider Electric Buildings AB in Malmö, Sweden.

Most of the ideas in the paper were obtained from discussions between all authors. The idea to use the normalized time delay came from K. J. Åström. The relation between the normalized time delay and the half-period ratio was found by J. Berner, with some assistance from former colleague M. Hast. All simulations were performed by J. Berner. The manuscript was mainly written by J. Berner with input and comments from the co-authors. The industrial experiments were performed by J. Berner and T. Hägglund in cooperation with M. Grundelius at Schneider Electric Buildings AB. The implementation of the autotuner in Schneider's software was made by J. Berner.

Paper II

Berner, J., T. Hägglund, and K. J. Åström (2016). "Asymmetric relay autotuning–Practical features for industrial use". *Control Engineering Practice* **54**, pp. 231–245.

This paper investigates practical aspects of the autotuner in Paper I. It provides a strategy to fully automate the parameter choices, steps and selections performed by the autotuner. Features including a startup procedure and adaptive relay amplitudes are proposed and described. The paper also shows experimental results from an industrial testing facility at Schneider Electric Buildings AB in Malmö, Sweden.

Suggestions on how to solve some practical issues were made based on previous experience by T. Hägglund and K. J. Åström, and all simulations and investigations were made by J. Berner. The industrial experiments were performed by J. Berner and T. Hägglund in cooperation with M. Grundelius at Schneider Electric Buildings AB. The implementation of the autotuner in Schneider's software was made by J. Berner. The manuscript was written by J. Berner with input and comments from the co-authors.

Paper III

Berner, J. and K. Soltesz (2017). "Short and robust experiments in relay autotuners". In: *2017 IEEE Conference on Emerging Technologies and Factory Automation (ETFA)*. Limassol, Cyprus.

This paper proposes a more advanced version of the autotuner. It uses shorter experiments and allows the experiment to start with non-stationary initial states of the dynamics to be identified. First-order and second-order models with time delay are obtained and an existing selection method of choosing between them is used.

The short experiments were used already in [Soltesz et al., 2016a], but were modified here by J. Berner to include the automated startup procedure. The major part of the paper, which is the identification method, builds on previous work by [Åström and Bohlin, 1966]. The identification was implemented in [Soltesz et al., 2016a], but was extended in this paper to include the estimation of the initial states. It was modified, implemented and described in the manuscript by K. Soltesz with some remarks and changes by J. Berner. The initialization of the identification method, as well as the model selections were implemented by J. Berner. The simulation study and the experiment section were performed and written by J. Berner with input from K. Soltesz. The remaining parts of the manuscript were written in cooperation between the authors.

Paper IV

Berner, J., K. Soltesz, T. Hägglund, and K. J. Åström (2017). "An experimental comparison of PID autotuners". Manuscript submitted to journal.

This paper compares industry standard autotuners with the two novel tuning strategies proposed in Paper I–II and Paper III. In total, four autotuners are evaluated on three laboratory processes with different characteristics. The identification method in Paper III is extended with an existing controller design method, and both our proposed procedures are extended with a controller filter in order to get complete autotuners. The results from the comparison show that the industrial standard could definitely be improved by incorporating the recent research advances.

The idea to perform this study came from J. Berner. The selection of suitable autotuners and processes was made by all authors. Experiment setups and designs were made in collaboration between the authors. Most experiments were carried out by J. Berner, sometimes in company of one or more of the co-authors. The manuscript was written by J. Berner with input and comments from the co-authors.

Paper V

Berner, J., K. Soltesz, T. Hägglund, and K. J. Åström (2017). "Autotuner identification of TITO systems using a single relay feedback experiment". In: *2017 IFAC World Congress*. Toulouse, France.

This paper provides an extension of the relay autotuner to multivariable (two-input two-output) systems. First-order time-delayed models for all subsystems are obtained from one single experiment where all loops are closed simultaneously. The method does not make any assumptions on the coupling level of the system and does not need to wait for convergence of limit cycles, which makes the procedure both generally applicable and fast. The method is evaluated in simulations.

The identification method was jointly developed by K. Soltesz and J. Berner. Decisions on the relay experiment design, as well as the simulations, were performed by J. Berner after discussions with the other authors. The identification section of

the manuscript was written in cooperation between K. Soltesz and J. Berner. The rest of the manuscript was written by J. Berner with input and comments from the co-authors.

Paper VI

Berner, J., K. Soltesz, K. J. Åström, and T. Hägglund (2017). "Practical evaluation of a novel multivariable relay autotuner with short and efficient excitation". In: *2017 IEEE Conference on Control Technology and Applications (CCTA)*. Kohala Coast, Hawaii.

In this paper the multivariable autotuner is extended with the possibility to find second-order models for the subsystems. The experiment is modified to do an amplitude change of the relay after a few switches to increase low-frequency excitation and improve the model quality. The experiment design and identification method are combined with an existing, but slightly modified, multivariable PID tuning algorithm to provide a complete autotuner. The autotuner is evaluated in simulations as well as on a laboratory quadruple tank process.

The modification of the experiment was an idea from discussions between all authors. The implementation of the more advanced identification method was made by J. Berner and K. Soltesz. The simulations and quadruple tank experiments were performed by J. Berner. The manuscript was written by J. Berner with input and comments from the co-authors.

Additional Publications

The following publications by the author are not included in the thesis

Berner, J. (2015). *Automatic Tuning of PID Controllers based on Asymmetric Relay Feedback*. Licentiate Thesis ISRN LUTFD2/TFRT--3267--SE. Dept. Automatic Control, Lund University, Sweden.

Berner, J., K. J. Åström, and T. Hägglund (2014). "Towards a New Generation of Relay Autotuners". *IFAC Proceedings Volumes* **47**:3, pp. 11288–11293.

Theorin, A. and J. Berner (2015). "Implementation of an Asymmetric Relay Autotuner in a Sequential Control Language". In: *2015 IEEE International Conference on Automation Science and Engineering*, pp. 874–879. DOI: 10.1109/CoASE.2015.7294191.

1.4 Delimitations

The autotuners developed in this thesis mainly focus on experiment design and model identification. Different controller tuning methods and filter designs have been applied in order to get the complete autotuning functionality, but not much effort has been put into selection, development or evaluation of tuning methods.

The autotuners proposed in this thesis mainly target the process industry, and typical process types for that setting. Fast, highly oscillating processes, like the ones commonly encountered in robotics, have not been evaluated. Also, all processes that have been used in simulations and experiments are either stable or integrating.

As is claimed in Section 1.2, the thesis work aims to maintain the benefits of the relay autotuner. The relay autotuner has been the basis of the work, and alternative autotuning methods have not been investigated.

2

Background

Automatic tuning of a controller requires some different steps to be performed. An experiment is made to retrieve process data, which can then be used to find some kind of model or description of the process. The estimated model is then used by the tuning method to find appropriate controller parameters. This thesis mainly focuses on how the experiment should be designed in order to obtain the necessary process data, and how to use that data to get a good model description of the process. However, this chapter gives some necessary background to all of the required steps.

In Section 2.1 the process control environment, that the autotuner is aimed for, is described. The controller structure that is used most in process industry, and throughout this thesis, is the PID controller. The PID controller is described in Section 2.2. Different model structures are described in Section 2.3, and methods of tuning the PID parameters from these models are given in Section 2.4. The automatic tuning procedure is described in Section 2.5, where it is also motivated why the autotuner is needed. The chosen autotuning strategy is the relay autotuner, which is described in Section 2.6. The chapter also contains explanations of some important concepts, and alternatives and motivations to the selected procedures.

2.1 Process Control

In process industries raw materials are physically or chemically transformed, or material and energy streams interact and transform each other [Craig et al., 2011]. Areas in process industry include for instance chemical industries, pulp and paper, minerals and metals, heating, ventilation and air conditioning (HVAC), pharmaceuticals, petrochemical/refining, and power generation. Almost all control loops in the process industry can be classified as either flow, pressure, liquid level, product quality or temperature control [Shinskey, 1996]. The goal of process control is to bring about and maintain the conditions of a process at desired or optimal values [Craig et al., 2011]. In order to keep the process condition at the desired value, feedback control loops are used. A typical feedback control loop is shown in Figure 2.1. The controller, $C(s)$, compares the reference value, r, to the measured process value,

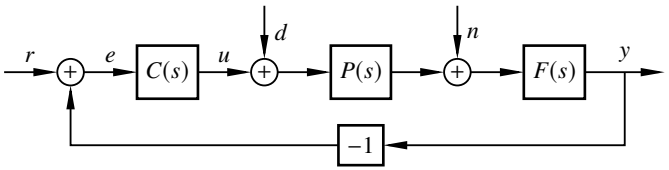


Figure 2.1 An illustration of a typical feedback control loop. The process, modeled by the transfer function $P(s)$, is controlled by the controller $C(s)$ to keep the measurement value y close to the reference value r. Disturbances d may affect the control signal u and noise n may affect the measurement value. A low-pass filter $F(s)$ can be included to reduce the influence of measurement noise. Note that the arrows in the block schedule are denoting the information flow and not necessarily a physical flow.

y, and based on the error, e, between them calculates a control signal, u, that is applied to the process, $P(s)$. Disturbances may occur at several places in the feedback loop. The ones that are considered here are a load disturbance, d, affecting the process input, and noise, n, affecting the measurement value. The influence of the measurement noise can be decreased by including a suitable low-pass filter $F(s)$ in the loop.

The controller objectives are, as stated earlier, to bring the process to the desired condition and to keep it there. This could also be stated as being able to handle changes in setpoint (or reference value), as well as staying at the desired setpoint, even in the presence of disturbances. Usually in a process control setting, the setpoints are infrequently changed, and the main focus of process control should therefore be to decrease the influence of load disturbances [Shinskey, 2002].

The absolute majority of controllers used in process industry are PID controllers [Desborough and Miller, 2002], and this thesis will therefore focus on that controller structure. The PID controller will be further described and motivated in Section 2.2. Over the last few decades more advanced control solutions, such as model predictive control (MPC), have started to become a standard in several industries and are routinely offered by many vendors [Craig et al., 2011]. The MPC controllers are usually used on a higher level, to control economical objectives for the factory. Some examples on economical motivations for improvements in process control, from [Craig et al., 2011], are that it can increase the throughput, reduce the fuel consumption and emission levels, and reduce the quality variability. When MPC is implemented, its manipulated variables are typically the setpoints of existing PID controllers [Desborough and Miller, 2002]. Hence the introduction of MPC does not decrease the importance of PID control. In fact, much of the improvement accredited to MPC in the process industry actually comes from the improved tuning of the basic loops [Åström and Hägglund, 2001].

2.2 PID Control

The PID controller is by far the most common controller type in industry [Desborough and Miller, 2002]. In an investigation made in the MCC Mizushima plant in Japan in 2010 it was found that the ratio of applications of PID control, conventional advanced controller structures like e.g. feedforward control, and modern advanced control like model predictive control (MPC), was 100:10:1 [Kano and Ogawa, 2010], and numbers like 90-97% are commonly used to describe the dominance of PID controllers [Åström and Hägglund, 2001; Ender, 1993; Desborough and Miller, 2002]. In a recent survey of industry impact, all the respondents considered PID control to have *"High multi-industry impact: Substantial benefits in each of several industry sectors; adoption by many companies in different sectors; standard practice in industry."* from which the authors draw the conclusion that *"90 years after its invention (or discovery), we still have nothing that compares with PID!"* [Samad, 2017]. Some reasons for the large dominance of the PID controller is that it is easy to understand, with only three parameters to tune, but at the same time complex enough to yield sufficient control performance for most applications. Another reason is that it is pre-programmed in every commercial control system, hence not requiring extensive implementation time [Desborough and Miller, 2002].

The PID controller contains three parts, the *proportional* (P) part, the *integral* (I) part and the *derivative* (D) part. The control signal is calculated as a combination of these parts. The proportional part looks at the current error between reference value and measurement value. The integral part looks at the cumulative error up until the current time. The derivative part looks at the change of the error, i.e., if the error is increasing or decreasing and how fast. The PID controller can be expressed in different formats, depending on how one wants to specify the parameters. In this thesis the parallel form will be used, in which the control signal is calculated as

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

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

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

In order to be practically useful, the PID controller has to be combined with a low-pass filter. The reason for this is that the derivative part has very high gain for high frequencies and hence is largely affected by measurement noise. Some common choices of filters are the first-order filter

$$F(s) = \frac{1}{1 + sT_f}, \tag{2.3}$$

or a second-order filter

$$F(s) = \frac{1}{1 + sT_f + (sT_f)^2/2}, \tag{2.4}$$

where T_f is the filter time constant. Second-order filters have the benefit of ensuring that the magnification decreases for higher frequencies, a property referred to as *high frequency roll-off*, while first-order filters only keep it bounded. The low-pass filter $F(s)$ could either be added to the derivative part only, in which case there is no high frequency roll-off, or to the entire controller. Filter design is an essential part of the controller design and the two should be performed simultaneously, since they affect each other. One common way of choosing the filter time constant is to make it a fraction of the derivative time constant, i.e., $T_f = T_d/N$. More elaborate procedures for how to choose T_f are presented in e.g. [Segovia et al., 2014; Soltesz et al., 2016b].

There are more parameters that are needed for a complete PID controller implementation. For instance the integral part needs to have some anti-windup scheme containing parameters, and the proportional part usually includes setpoint weighting with a parameter that should be set. These parameters will, however, not be considered in this thesis. Neither will implementation issues concerning for instance discretization, and how to switch between automatic and manual mode in a smooth way. Further information about these parameters and issues can be found in for example [Åström and Hägglund, 2006].

2.3 Modeling

For control purposes, all physical processes have to be approximated by simple models. Most controller design methods rely on a dynamic model of the process to be controlled. Often the overall dynamics of the process is modeled by a transfer function, as for instance the process model $P(s)$ in Figure 2.1. This section will list the model descriptions used for controller tuning in this thesis, and comment on some important aspects of the models needed for control.

System dimensions

When one input signal is used to control one process variable it is said to be a *single-input single-output* (SISO) system. This could for instance be the case when one pump is used to control the water level of one tank, where the control signal is the power to the pump and the process variable is the water level. If on the other hand multiple input signals are used to control multiple interacting process variables, the system is said to be *multivariable* or *multi-input multi-output* (MIMO). Instead of a scalar transfer function, the multivariable process can be described by a transfer

function matrix relating all inputs with all outputs

$$\begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} P_{11} & \dots & P_{1m} \\ \vdots & \ddots & \vdots \\ P_{n1} & \dots & P_{nm} \end{pmatrix} \begin{pmatrix} u_1 \\ \vdots \\ u_m \end{pmatrix}, \tag{2.5}$$

where $y_1 \dots y_n$ are the outputs, $u_1 \dots u_m$ the inputs, and P_{ij} the transfer function between input u_j and output y_i. Note that the number of outputs and the number of inputs do not have to be the same, it could be that $n \neq m$.

A special case of MIMO is the *two-input two-output* (TITO) system, shown in Figure 2.2, where $n = m = 2$. An example of a TITO system could be if both the level and the temperature of a water tank should be controlled using two pumps, providing hot and cold water respectively. If the interaction between the variables is not large, that is if the P-matrix in (2.5) can be made (almost) diagonal, a MIMO system can be treated as multiple SISO systems. Which controller that should be connected to which measurement, i.e., how the *pairing* should be made, can be selected by calculating the *relative gain array* (RGA), described in e.g. [Åström and Hägglund, 2006]. If the interactions, or couplings, are strong, then treating the system as separate SISO loops is not recommendable, and may even cause instability of the feedback loops. Some alternatives are then to either make a static or dynamic decoupling of the systems, or to use a multivariable controller that calculates all control signals based on all measured process values. Decoupling methods are presented in many textbooks on control, for instance [Åström and Hägglund, 2006]. In [Boyd et al., 2016] one option of a multivariable PID controller is proposed, which will be used later on in this thesis.

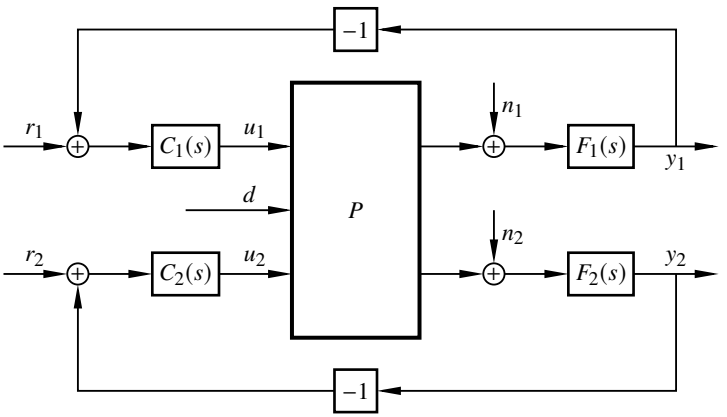


Figure 2.2 Schematics of a TITO system controlled by two separate controllers.

Model orders

The process models, or elements of the process transfer function matrix, that will be used in this thesis are integrating, first-order or second-order models with time delay. Even though the physical processes may have more complex or non-linear dynamics, many of them can be adequately described by one of these simple models, at least in a region close to a desired working point. The phase loss from the higher-order dynamics is approximated by an increase in the time delay of the model.

The integrating model with time delay (ITD) that will be used in the thesis is defined as

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

where k_v is the integrator gain, and L the time delay of the process. For first-order models with time delay (FOTD) the parametrization

$$P(s) = \frac{K_p}{1 + sT} e^{-sL}, \tag{2.7}$$

where K_p is the static gain and T the time constant of the process, will be used in some papers, and the alternative formulation

$$P(s) = \frac{b}{s + a} e^{-sL}, \tag{2.8}$$

where $b = K_p/T$ and $a = 1/T$, in others. The second order models used are the integrating plus first-order time-delayed (IFOTD) model

$$P(s) = \frac{k_v}{s(1 + sT)} e^{-sL}, \tag{2.9}$$

and the second-order time-delayed (SOTD) model

$$P(s) = \frac{K_p}{(1 + sT_1)(1 + sT_2)} e^{-sL} \tag{2.10}$$

or alternatively on the form

$$P(s) = \frac{b}{s^2 + a_1 s + a_2} e^{-sL}. \tag{2.11}$$

The parametrizations in (2.7) and (2.10) have the benefit of immediately showing the static gain and time constants of the process, but they cannot be used to describe integrating processes. Hence the need of the separate ITD and IFOTD models. The parametrizations in (2.8) and (2.11) include the integrating models, but can on the other hand not be used to describe pure time-delays. These parametrizations are more convenient to use for optimization purposes, but lack the immediate display of static gain and time constants.

Normalized Time Delay

The normalized time delay is a measure of the relative importance of the time delay, L, in comparison to the dynamics of the process, represented by a time constant T. It is defined as

$$\tau = \frac{L}{L+T}, \quad 0 \leq \tau \leq 1. \tag{2.12}$$

The concept of normalized time delay can be generalized to processes of higher order dynamics by considering the *apparent time constant* and *apparent time delay*. These are obtained from an FOTD approximation of the process, obtained from step response analysis.

Processes with a large value of τ, that is $\tau \approx 1$, are said to be *delay-dominated* since the time delay is much larger than the time constant. Processes with a small value of τ are said to be *lag-dominated* and are more influenced by their dynamics due to the comparatively large time constant T. Processes with intermediate values of τ are said to be *balanced*.

The classification of processes into lag-dominated, balanced or delay-dominated is useful when it comes to the tuning of PID controllers. It was shown in [Åström and Hägglund, 2006] that delay-dominated systems (large τ) do not benefit much from derivative action, while lag-dominated processes can gain a lot in performance by using the derivative part. It was also shown that while an FOTD model is usually sufficient to describe delay-dominated systems, lag-dominated systems can be much more appropriately described by increasing the model order. The selection of model orders and controller parameters are major ingredients in the design of an autotuner, and the knowledge of the normalized time delay is therefore highly beneficial.

The idea to use the information from the τ-value 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 to select tuning method, and to find an FOTD model from the relay experiment. Paper I proposes a simpler method to find and use this information. There are also discussions on using the τ-value for model selection for the autotuner in Paper III.

2.4 PID Tuning

Tuning the PID controller consists of selecting the parameter values K, T_i and T_d in (2.2). It could be done manually by modifying the parameter values to see if certain requirements are satisfied or not. There are several basic rules of thumb one can apply, for example, if the response to setpoint changes is too slow you may need to increase the proportional part K, if the attenuation of load disturbances is too slow you need to decrease the integral time T_i etc. More intuition about how the parameters should be changed to achieve the wanted behavior is given in any classic book on PID control, like for instance [Åström and Hägglund, 2006]. In order to reduce the amount of manual tuning and better utilize the knowledge of PID

tuning, a vast number of different tuning rules have been developed during the years. This section will present some of them, and also give some examples of common requirements for the control loop.

Requirements

Typical requirements on the control loop are related to load disturbance attenuation, and robustness to process variations and measurement noise. A measure of load disturbance attenuation is the *integrated absolute error*, IAE-value, defined as

$$\text{IAE} = \int_0^\infty |e(t)|dt, \tag{2.13}$$

where $e(t)$ is the error due to a unit step change, at $t = 0$, in the load disturbance d, entering at the process input as in Figure 2.1.

Robustness to process variations can be described by the sensitivity function

$$S(s) = \frac{1}{1 + P(s)C(s)}, \tag{2.14}$$

and complementary sensitivity function

$$T(s) = \frac{P(s)C(s)}{1 + P(s)C(s)}. \tag{2.15}$$

Restrictions on the maximum values of these sensitivities,

$$\begin{aligned} M_S &= \max_\omega |S(i\omega)|, \\ M_T &= \max_\omega |T(i\omega)|, \end{aligned} \tag{2.16}$$

are common to ensure robustness. In this thesis the combined maximum value,

$$M_{ST} = \max(M_S, M_T), \tag{2.17}$$

will be used as a robustness measure.

In addition to the requirements on IAE and M_{ST}, many other performance requirements 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. The integral error,

$$\text{IE} = \int_0^\infty e(t)dt, \tag{2.18}$$

is used for the optimization-based controller tunings in [Hast et al., 2013] and [Boyd et al., 2016], that are used in Paper IV and Paper VI. Minimizing IAE is usually preferred to minimizing IE, since IE may be small even when the load response is highly oscillatory. The IE is, however, more computationally tractable from an optimization point of view, since minimizing IE is equivalent to maximizing the integral gain of the controller [Åström and Hägglund, 2006]. In [Hast et al., 2013] the authors discuss this issue and state that if the minimization of IE is combined with requirements on robustness, it usually gives controllers with good properties. Apart from the tuning strategies in Paper IV and Paper VI, which are using IE, the performance and robustness measures in this thesis will be restricted to IAE and M_{ST}.

Tuning Methods

There are numerous methods for tuning of PID controllers, ranging from the simple classic rules proposed in [Ziegler and Nichols, 1942], to advanced solutions based on optimization. In [O'Dwyer, 2009] about 1700 different PID tuning rules are listed based on different model structures and performance requirements. It is common to compare new tuning methods for both PID controllers and other control structures to the Ziegler-Nichols rules. That is, however, not always a fair comparison, since they are known to perform poorly in many situations and many better PID tuning algorithms exist. According to [Åström and Hägglund, 2001] it is very easy to demonstrate that any controller with reasonable tuning will outperform a PID with Ziegler-Nichols tuning for most processes.

Some existing tuning rules based on an FOTD model of the process are λ-tuning [Dahlin, 1968; Sell, 1995], the SIMC [Skogestad, 2003; Skogestad, 2006] and AMIGO [Åström and Hägglund, 2006]. Different tuning rules are derived to satisfy different performance requirements, or suit different process types. The choice of tuning methods has not been the focus of this thesis, but since the controller tuning step is required for a complete autotuner, a selection of methods are provided. The selected methods are similar to each other and applicable to the models at hand.

In Paper I and Paper II the AMIGO tuning rules from [Åström and Hägglund, 2006], and the optimization-based tuning described in [Garpinger and Hägglund, 2008], are the two methods used. The AMIGO rules are based on an approximation of the optimization method MIGO, also described in [Åström and Hägglund, 2006], that minimizes the integral error (IE) with restrictions on the maximum sensitivities M_S and M_T. The AMIGO rules were derived from the test batch from [Åström and Hägglund, 2006], which is listed in Appendix B of Paper II. The AMIGO method provides tuning rules for PI controllers based on FOTD and ITD models, as well as PID controllers based on FOTD, ITD, SOTD and IFOTD models. The method from [Garpinger and Hägglund, 2008] minimizes IAE with constraints on M_{ST}. In Paper IV the tuning method described in [Hast et al., 2013], using a convex-concave method to minimize IE with constraints on M_S and M_T, is used. For the TITO-

autotuner proposed in Paper VI, a multivariable extension of the [Hast et al., 2013] method, described in [Boyd et al., 2016], is used. However, since all the proposed autotuners in this thesis provide FOTD or SOTD models, the use of other tuning methods is straightforward.

2.5 Automatic Tuning

Despite the low complexity of the PID controller and the many available tuning methods, it is still common that control loops in the process industry are not working satisfactory. Reports are stating that more than 30 % of the process controllers installed operate in manual mode [Ender, 1993], and in the study presented in [Desborough and Miller, 2002] the numbers are that 36 % of the controllers are operated in open loop (manual), 32 % are performing poor or fair, and 32 % perform acceptable or excellent. In the more recent study in [Kano and Ogawa, 2010], a focused project on improving the performance of PID control systems resulted in an increase to 90 % of the PID controllers in the participating factories operating in automatic mode. The study showed that there is a great interest in the control performance activity from the chemical and petroleum refining enterprises in Japan, which the authors take as an indication that the process control section had not realized that the operation section was dissatisfied with the control performance [Kano and Ogawa, 2010]. Hence, the need of improving the PID controller tuning in industry is still highly relevant.

In an oil refinery, chemical plant, paper mill, or another continuous process industry facility, there are typically between five hundred and five thousand regulatory controllers [Desborough and Miller, 2002]. Tuning all these by hand would require a lot of man-hours and a lot of knowledge. Only the modeling part typically has an engineering cost of $250-$1000 per SISO loop [Desborough and Miller, 2002]. Being able to find a decent controller tuning in an automated way is therefore highly beneficial, and autotuning of PID controllers is listed as one of thirty examples on "Success stories for control" in [Samad and Annaswamy, 2014]. Due to the importance of the problem, a very large variety of PID autotuners have been developed and are currently available on the market [Leva et al., 2002].

Common for what we define as autotuners is that they go through the different steps in Figure 2.3. First they perform some kind of experiment to retrieve process data. The retrieved process data is used to create a description of the process, a model, that in the next step can be used to find appropriate controller parameters. The last step depicted in Figure 2.3, the evaluation step, is usually not performed by the autotuner but rather by the user who has to make the decision if the obtained controller performance is sufficient or if any step has to be redone. The evaluation step is important since no autotuner will ever be perfect. As was stated in [Leva et al., 2002] *"[...] it must be made clear that a sufficiently skilled human with sufficient process knowledge, data and time available can outperform any autotuner in any*

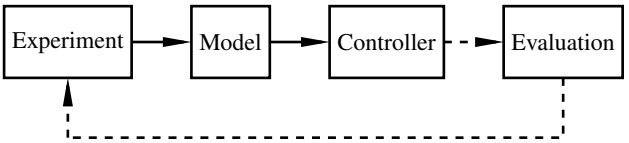


Figure 2.3 Steps to be designed and performed in an automatic tuning procedure. The dashed lines show the steps that involve the user. Figure from Paper II.

situation.". The aim of an autotuner is to give decent controller performance for a wide range of processes, and in that way significantly decrease the amount of time needed for manual tuning. It is important that an autotuner is easy to use, also for users not that familiar with control theory. It is also important that the autotuner is widely applicable, removing the need of switching between different autotuning tools depending on the type of process to be controlled. Or as it was stated in [Craig et al., 2011], *"Industry could live with 95% optimality but not with five different optimization tools in one plant."*.

Alternative procedures

The autotuners in this thesis have experiment designs based on asymmetric relay feedback, a methodology that will be described and motivated further in Section 2.6. From the experiment data one of the low-order transfer function models listed in Section 2.3 will be estimated, either by simple formulas or by optimization methods depending on which autotuner version that is used. The controller tuning methods used to find controller parameters from the estimated models also depend on the autotuner version. These methods and choices will be explained in Chapter 3.

Of course it exists other approaches to the experiments, modeling and controller design. Some alternative strategies to do the experiment and modeling part, or the system identification as it is usually referred to, will be briefly described in this section. All system identification methods start with the design of the input signal to the process. Experiments could be performed 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 $90^\circ - 180^\circ$ are of particular interest. When the experimental data has been obtained, it needs to be analyzed to find the desired model. A common way is to use some parameter estimation method to obtain process models, and then apply various testing methods like estimation error, Akaike's Information Criterion [Akaike, 1974], parameter variances, etc., to determine a proper model structure.

Another common and simple open-loop identification method is to look at a step response. By identifying the steepest slope of the step response, as well as its rise time, stationary value and apparent time delay, the first-order models given in Section 2.3 can be estimated. 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 quantities and slopes from the experiment data accurately.

All the mentioned signal types for open-loop identification have the drawback that process information is needed, in order to design the input signals to give the desired excitation. Knowledge about time scales, frequency ranges, and/or amplitudes is required. This is, however, not a problem for the relay feedback experiment since it will provide excitation in the interesting frequency range for PID control automatically. The reason for this is that the relay feedback causes the process to oscillate with the critical frequency ω_c, that is, the frequency where the phase lag of the process is 180°. More details about the relay feedback experiment will be given in Section 2.6.

The selected approaches for the autotuners in this thesis follow the lines of traditional system identification, but are guided by the fact that we want models suitable for design of PID controllers. Due to the low complexity of the PID controller, it is reasonable to restrict the model selections to the low-order model structures described in Section 2.3. 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 second 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 the desired number of model parameters. A 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 a symmetric and an

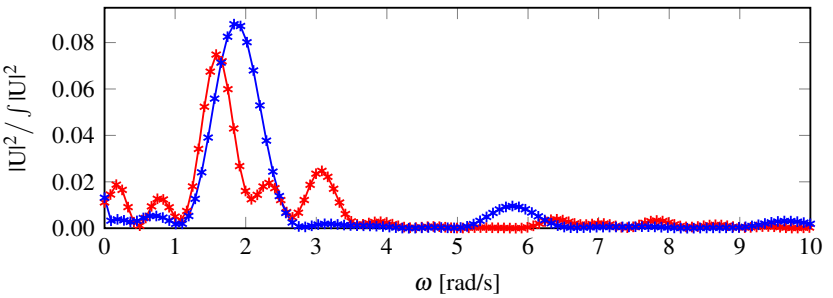


Figure 2.4 Frequency spectra for two relay experiments on the process $P(s) = e^{-sL}/(s+1)$, which has a critical frequency $\omega_c \approx 2$. The notation U is used for the Fourier transform of u. The blue curve shows the spectrum for a symmetric relay experiment. The red curve shows the spectrum for an asymmetric relay experiment, where one of the relay amplitudes were five times the size of the other.

asymmetric relay is shown in Figure 2.4. The process simulated in this figure is

$$P(s) = \frac{1}{s+1}e^{-s} \tag{2.19}$$

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. As was mentioned before, this is within the interesting frequency interval for PID control. However, the asymmetric relay has a more spread-out frequency content, hence covering more of the interval.

Model evaluation

To evaluate whether the obtained model describes the real process well or not is a difficult issue. A number of different measures can be used to compare the models. One common way is to compare the step response of the model with the one of 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]. The optimization-based autotuners in this thesis, estimate their models by minimizing the error between the measured process output and the output from the model fed with the same input data. This comparison is, as well as the step response, made in open-loop. Other common ways to evaluate the model in open loop is to look at the Nyquist and/or Bode diagrams, however, those evaluations require the diagrams of the true process to compare with.

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]. The ν-gap provides an upper bound on how much a specific stability margin would be decreased if a controller designed for P_1 is applied on process P_2 [Vinnicombe, 2001]. This property makes

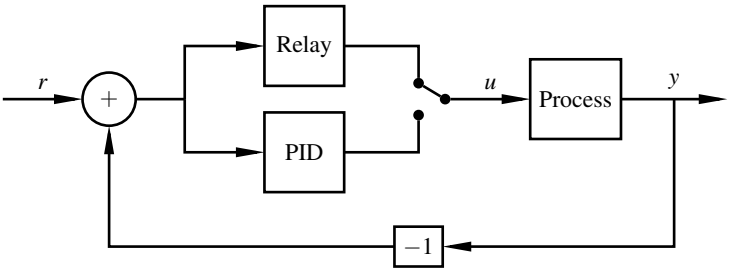


Figure 2.5 The setup for the relay feedback experiment. When the experiment starts, the PID controller is disconnected, and instead the process output y is controlled by a relay function. When the experiment is done and the PID controller parameters are tuned, the system switches back to PID control.

the Vinnicombe metric a good measure for an 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. However, as for the Bode and Nyquist evaluations, the ν-gap metric requires that you know the true process model. Since you do not have access to that when you use the autotuner, this measure would mainly be useful for evaluation in an autotuner-development phase using well-defined simulations.

2.6 The Relay Autotuner

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 automated and controlled way. By introducing a relay function in the control loop, as shown in Figure 2.5, 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. A typical output from a relay experiment is shown in Figure 2.6. 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 range automatically. In the experiment in [Åström and Hägglund, 1984] the zero-crossings and the peak amplitudes of the process output were 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 a combination of the specified amplitude and phase margin. A relay with hysteresis was introduced to deal with measurement noise. With hysteresis the obtained 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.

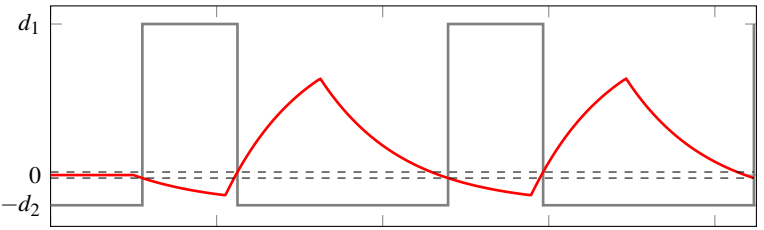


Figure 2.6 Output from a typical (asymmetric) relay feedback experiment. The relay function (gray) switches between d_1 and $-d_2$ every time the process output (red) leaves the hysteresis band (dashed black). The working point (u_0, y_0), around which the oscillations occur, has been normalized and denoted by 0 in this figure.

The relay autotuner has since its introduction been widely used in industry. Variations on the relay method have become a *de facto* standard for commercial autotuning controllers, though vendors rarely mention which technology they use [VanDoren, 2006]. Apart from that no prior information about the process is needed, some additional benefits of the relay autotuner have 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 function models 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. In [Leva, 1993], a time delay was added to adjust the obtained critical point. In [Yu, 2006], the relay function was saturated to produce more sine-like output curves and in that way improve the estimation of ω_c and k_c. In [De Keyser et al., 2012], restrictions on M_S were used instead of phase or amplitude margins providing more reliable controllers. The most common modification is, however, to find one of the low-order models described in Section 2.3 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 autotuners 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 asymmetric relay test was first presented in [Shen et al., 1996], where the asymmetry was introduced in the switching conditions of the relay. In most later versions of asymmetric relay functions, including the one used in this thesis, the asymmetry is instead introduced in the relay amplitudes, as is seen in Figure 2.6 where $d_1 \neq d_2$. For a complete definition of the relay function used in this thesis, see Paper I. The possibility to estimate the static gain from the relay experiment provides a way to get an FOTD model from the experiment, instead of a 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., 1996] used DFA, which is not recommendable when the relay is asymmetric. The reason for this is that 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 were 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 them to get $G(i\omega_i)$ for some different frequencies ω_i.

The approach for the autotuner presented in Paper I and II 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 curve-fitting features that can be used are the time period of the oscillation, the amplitudes of the oscillation, the times when the maximum amplitudes occur, 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. But if the autotuner is used in an industrial setting, measures that are easily and robustly determined even in the presence of noise are required. Because of that, the only data used from the relay experiment in those papers, 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, 2001], [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] have 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 measurement is, however, quite sensitive to noise and in the results they used an average of 10 limit-cycle oscillation periods to obtain their values when noise was added. This gives a rather long experiment time, which is not useful in practice.

Multivariable Relay Autotuning

Given the benefits of the relay autotuner it is tempting to extend it to handle interacting systems. The aim of a multivariable relay autotuner could either be to tune a number of SISO PID controllers by using a suitable pairing or decoupling strategy, or to tune a multivariable PID controller as in e.g. [Boyd et al., 2016]. There has been some interest in the extension of relay autotuning to multivariable systems, even if the literature on the subject is much more limited than for the SISO case. One study comparing different multivariable relay autotuning methods available at the time was presented in [Menani and Koivo, 2001], where benefits and drawbacks of the methods were listed together with the performance results.

One thing that differs when extending the relay feedback to multivariable systems is the concept of the critical point. In the SISO relay autotuner the principle is that the critical gain and frequency are found from the oscillations. When a multivariable system is considered there is no longer one critical point, there are multiple points, or rather a critical surface [Campestrini et al., 2006]. Which critical point that is found will depend on the relay settings, and the obtained point will influence the achieved tuning. This issue complicates the tuning procedure for multivariable systems.

Alternative methods to relay feedback exist also for autotuning of TITO systems. One recent option is described in [Pereira et al., 2017], where they use closed-loop setpoint step responses to estimate the process parameters. The method requires at least two setpoint steps and has to ensure that steady-state is reached in between. It also requires PID controllers at start, which is problematic if it is used for start-up of a process when no such information is available. The nominal PID controllers only need to be stabilizing and will not affect the end-result, but they will influence the duration of the overall experiment. Since this thesis focuses on relay experiments, other options will not be described further.

As was stated in [Wang et al., 1997c], there are three main alternatives when it comes to relay autotuning of multivariable systems. Either you could do an *independent* tuning of each loop, a *sequential* tuning or use *decentralized relay feedback*. The multivariable autotuner presented in this thesis is based on a decentralized feedback

experiment, but each of these procedures will be described further in the following corresponding subsections.

Independent tuning The first option that comes to mind is probably to tune each loop independently while leaving the others in manual. This gives the possibility to treat the MIMO system as a number of SISO systems and use the well-established SISO methods on each loop. However, that strategy will not take any cross-couplings into account, which might give really poor controller performances if the interactions in the system are strong. A commonly referred version of independent tuning is to do relay experiments on each loop, where the pairing between inputs and outputs are decided on beforehand, and then use the *biggest log modulus tuning* (BLT) method proposed by [Luyben, 1986], to tune the decentralized controllers. The BLT method introduces an appropriate detuning factor to the controller to keep the system stable in spite of interactions.

Another way of using independent tuning is made in [Menani and Koivo, 2003], where not only the diagonal elements, but also the interactions, are obtained from independent relay experiments. This allows them to tune a multivariable PI controller for the system. The interactions are obtained from measurements of the amplitudes of all outputs at each single relay experiment. This gives information about all $n \times n$ elements from n experiments. It is, however, required that all loops are made to oscillate with the same frequency in order for the method to work, therefore the procedure is divided into four stages where the first two are conducted to find the appropriate design frequency, and the last two are used to get the wanted process data and controller design. The method hence requires numerous experiments.

One could of course also do independent relay tests on every input-output combination, requiring $n \times m$ separate experiments for a system with n outputs and m inputs, to get information about all cross-couplings. Normally, when referring to independent tuning, the assumed controller structure consists of decentralized SISO PID controllers without any decoupling strategy. In that case the cross-coupling information would not be of any use, and only requiring additional experiments. If one would use a MIMO PID, or some decoupling strategy based on the cross-coupling information it could, however, be a solution to the interaction.

Sequential tuning The second option is to do a sequential tuning, where the first loop is tuned while the other loops are in manual, and then the next controller is tuned with the first loop closed and the rest in manual, and so on until all controllers are tuned. This method was first proposed by [Loh et al., 1993]. The sequential procedure gives controllers that are tuned with the information up until that loop available, but no information about the coming loops. Therefore the procedure is normally iterated a couple of times until the controller settings converge. According to e.g. [Shen and Yu, 1994; Yu, 2006], it typically takes three to four relay experiments before a 2×2 system, with a known input-output pairing, converges. A sequential tuning method that does not require multiple iterations was proposed in [Koo et al., 2004]. There FOTD models for all elements in the transfer function matrix are estimated

from the n sequential experiments for an $n \times n$ system. Since it provides all process information, it can be used to change controller tuning or incorrect pairings without requiring more field-experiments, which is a clear benefit. The question of whether or not the entire transfer function matrix should be estimated is addressed in for instance [Shen and Yu, 1994], where it was argued that one of the benefits of the sequential design is that it only identifies the transfer functions really needed for (the decentralized) controller design.

A recent sequential tuning method was introduced in [Ionescu et al., 2016], where relay experiments with and without an extra delay are iterated to find PID controllers from moving the critical point to ensure a desired M_S as in [De Keyser et al., 2012].

The experiment data from sequential relay feedback could also be used for other identification methods, not relying on critical points. In [Wang and Zhang, 2001], the sequential relay feedback was used as one possible way of providing sufficient excitation to an FFT method estimating the parameters of the transfer function matrix. In [Toh and Rangaiah, 2002], extra short sequential relay tests were performed to estimate FOTD or SOTD models using curve-fitting and then tune the corresponding PID controllers from these models. The curve-fitting was made by minimizing the sum of squares of the error between the actual and calculated closed-loop responses.

Decentralized relay feedback The third alternative is to tune all control loops simultaneously. This means that all loops are closed with relay functions at the same time. This is a completely closed-loop experiment, meaning that all cross-couplings will influence the results. This strategy is the one selected for the multivariable (TITO) autotuner in this thesis. Decentralized relay feedback was first proposed in [Palmor et al., 1993], where a desired critical point as well as the steady-state gains were found from n decentralized relay experiments and then Ziegler-Nichols rules were used to find controller parameters for a decentralized PID controller.

The first to use the decentralized relay experiment to tune a full multivariable controller were [Wang et al., 1997c]. They obtained the frequency response $P(i\omega)$ at two points, $\omega = \{0, \omega_c\}$, from n decentralized experiments with slight modifications of the relay amplitudes for the different experiments. In [Chidambaram and Sathe, 2014] the method of [Wang et al., 1997c] was extended to find FOTD and SOTD models from the experiments. A modified version of the decentralized relay experiment was used in [Wang et al., 2003]. There only the first oscillation was used of each of the n experiments, and instead of using critical point information, the experiment data was used to find the transfer function matrix from Fourier analysis. In a recent paper by [Nikita and Chidambaram, 2017], a decentralized relay experiment was used to find the ultimate values. The ultimate gain value was then modified by higher order harmonic terms if the wave-form of the process output was close to triangular.

One drawback with the decentralized relay feedback is that it has to consider the multivariable system and can not rely on SISO methods as the other methods.

Apart from the existence of multiple critical points discussed earlier, analysis of limit cycles has to be extended to multivariable systems. In [Palmor et al., 1995] methods for determining the periods and stability of limit cycles in decentralized relay systems were presented. It was assumed that limit cycles exist, are symmetric, and that all loops oscillate with the same frequency. In [Lin et al., 2003] alternative conditions were given that are also valid for asymmetric limit cycles, while the rest of the assumptions were still the same.

The assumption that all loops are oscillating with the same frequency is made in almost all existing papers on decentralized relay feedback. If the interactions in the system are large enough this will be true, and attempts to find conditions for this have been made in [Loh and Vasnani, 1994]. However, to my knowledge there are no simple universal conditions available. This issue and its implications are discussed further in Paper V.

3

Three Versions of the Autotuner

3.1 Introduction

This thesis proposes three different novel autotuners. The first autotuner is called the τ-tuner, since the normalized time delay τ plays a crucial role in its procedure. The second one is called the NOMAD autotuner, where NOMAD is short for *Noise-robust, Optimization-based Modeling And Design*. The third one is called multi-NOMAD, since it is a multivariable extension of the NOMAD autotuner. Those names were not assigned to the procedures in the original papers, but were introduced to simplify the comparisons in Paper IV and in this thesis. In this chapter the three autotuners will be described, and their differences and similarities will be commented upon. Drawbacks and benefits of the different versions will also be discussed.

3.2 The τ-tuner

The τ-tuner is described thoroughly in Paper I–II. It is an autotuner for SISO systems that uses asymmetric relay feedback to create limit cycle oscillations. It uses the half-periods, i.e., the time between two consecutive relay switches, and the integral of the process output to calculate an ITD model or an FOTD model from analytic equations. The measurements of these features are robust to noise, which makes the τ-tuner more practically applicable than many other autotuners in literature that can find more exact models, but require numerous oscillations in order to ensure good quality of their measurements. Some examples of other relay autotuners were given in Section 2.6.

The τ-tuner finds the normalized time delay from the simple relationship,

$$\tau(\rho, \gamma) = \frac{\gamma - \rho}{(\gamma - 1)(0.35\rho + 0.65)} \tag{3.1}$$

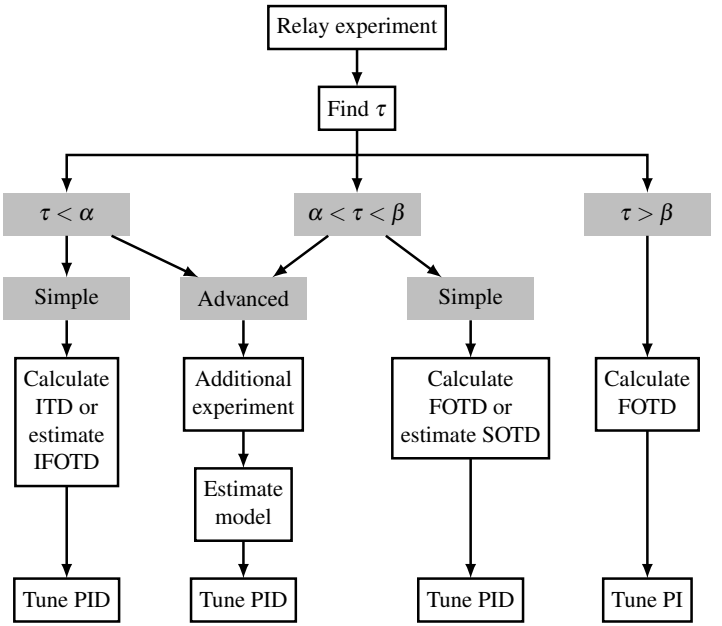


Figure 3.1 Decision scheme based on the estimated normalized time delay. Figure from Paper I.

between the half-period ratio ρ and the asymmetry level γ of the relay, derived in Paper I. The estimated normalized time delay, τ, is used to make model and controller selections, and is also included in the derived model equations. The decision scheme based on τ is shown in Figure 3.1. The version referred to as the τ-tuner in this thesis is the simple version in this figure where only FOTD or ITD models are calculated and used for controller tuning. In the papers the τ-tuner uses the AMIGO tuning method [Åström and Hägglund, 2006]. The AMIGO was chosen since it provides simple tuning rules for the model types that were estimated. Any other method with those features could also have been chosen, and since tuning rules are specialized for a certain modeling method [Garpinger et al., 2014], an option could be to derive a specific tuning rule to combine with these experiments. Since PID controllers with a derivative part need to be filtered in order to be practically useful, a filter design was added to the τ-tuner in Paper IV.

The τ-tuner has the same benefits as the classic relay autotuner from [Åström and Hägglund, 1984] in that it is easy to implement, understand and use. The τ-tuner has been implemented in Matlab/Simulink, the in-house software at Schneider Electric, in the sequential programming language JGrafchart [Theorin and Berner, 2015], and in the simulation environment Modelica/Dymola in [Björk and Levenhammar, 2017]. Apart from being evaluated in simulation studies, the τ-tuner was evaluated in an industrial setting at Schneider Electric Buildings AB in Malmö, Sweden. There it

was used to find controller parameters for a pressure control loop and a temperature control loop in an air handling unit. The results from these experiments are shown in Paper I and Paper II. The τ-tuner was also evaluated for three laboratory processes with different dynamic characteristics in Paper IV.

The τ-tuner is similar to the classic autotuner in experiment lengths and the amount of information retrieved from the experiment. However, the asymmetry of the relay provides the possibility to estimate better models, and hence opens up new and better possibilities for the controller tuning.

Unfortunately, the τ-tuner also shares some drawbacks with the classic autotuner. It is crucial that the process is in (or close to) steady-state when the experiment is started, since any drifting will cause erroneous asymmetry levels. Uncertainties in asymmetry level will influence the estimation of τ in (3.1) causing problems in the decision making and model estimations. For the same reason the τ-tuner is sensitive to disturbances entering during the experiment, and to low resolution in AD/DA converters. These practical issues are discussed in Paper II.

3.3 The NOMAD Autotuner

The NOMAD autotuner uses a short asymmetric relay feedback test as a convenient way of getting experiment data with good excitation to use in a model parameter optimization method. The experiment design and model identification procedure is an extension of the work presented in [Soltesz et al., 2016a], and is described in Paper III. A gradient-descent method is used to estimate FOTD and SOTD models from the data. The model selection is made using the Akaike Information Criterion [Akaike, 1974]. The selected model is then used by a convex-concave optimization method, described in [Hast et al., 2013], to obtain PID controller parameters that minimize the integral error (IE) with constraints on the maximum sensitivities M_S and M_T. The controller design, as well as a filter design, was added to the NOMAD autotuner in Paper IV. In that paper the complete NOMAD autotuner was also evaluated on three laboratory processes with different dynamic properties, showing very good performance.

One of the main features of the NOMAD autotuner is that it uses the entire experiment data set, instead of retrieving a number of specific points or intervals from the experiment. Since it does not need accurate measurements of period times or oscillation amplitudes, it does not need to wait for convergence of the limit cycle oscillations. Hence the experiment can be made much shorter, and is terminated after only 3 relay switches, i.e., one oscillation period. The short experiment time is a great advantage compared to other relay autotuners, partly since it increases the operating availability of the control loop, but mainly since it reduces the risk of disturbances entering during the experiment, deteriorating the results.

A similar approach is taken in [Viswanathan and Rangaiah, 2000] where an SOTD model is identified from closed-loop experiments. The procedure in that

paper also benefits from short asymmetric relay experiments that do not require steady oscillations. The benefit of using an asymmetric relay is emphasized in their study, where they try some different asymmetries, or input biases as they call it, and conclude that symmetric relays (no input bias) was inadequate for identifying an SOTD model. Their studies also showed that while the experiment duration only had marginal effect on the obtained model parameters and IAE value, the accuracy generally improved with input bias. Some differences between the procedure in [Viswanathan and Rangaiah, 2000] and the NOMAD autotuner are that in [Viswanathan and Rangaiah, 2000] they only consider the experiment design and model identification, while NOMAD has a full autotuner capability. The NOMAD also handles the situation when the experiment is started in non-steady state by estimating the initial conditions in addition to the model parameters. This is not done in [Viswanathan and Rangaiah, 2000], and provides a great benefit compared to the τ-tuner and the classic relay autotuner in [Åström and Hägglund, 1984], that are highly dependent on stationary starting conditions.

The NOMAD estimates both an FOTD and an SOTD model and then uses the Akaike Information Criterion to decide which model structure to use, while [Viswanathan and Rangaiah, 2000] always finds an SOTD model. Also, the parametrization of their SOTD model restricts them from finding integrating models. The procedures use similar cost functions, but different optimization methods for the model estimation. Both, however, have problems with local minima. In [Viswanathan and Rangaiah, 2000] this is handled by first applying a global optimization method followed by a local more efficient method, while the NOMAD uses a local efficient method from the start, but initiates it from several random starting points. A better solution to this issue would be beneficial for both approaches.

3.4 The Multi-NOMAD Autotuner

The multi-NOMAD is an extended version of the NOMAD autotuner, implemented for TITO systems. Some modifications were needed to make it suitable for multivariable systems. First of all the experiment used has been extended to a decentralized relay experiment. The selection of this simultaneous approach was mainly motivated by the possibility of sticking to one single experiment, hence reducing the total experiment duration. A difference between the multi-NOMAD and the decentralized methods described in e.g. [Wang et al., 1997c; Chidambaram and Sathe, 2014] is that multi-NOMAD does not assume anything about the level of interaction. The multi-NOMAD, like the NOMAD, uses the entire data set and does not use any critical point, which makes it unaffected by the difficulties with selecting which critical point to use that were described in [Campestrini et al., 2006]. It also allows for shorter experiments, which do not require steady oscillations. These issues are discussed in the first paper on the multi-NOMAD, Paper V.

In the second paper, Paper VI, the decentralized relay experiment was modified

slightly. The reason for this was that it turned out that the static gain of some processes were not estimated accurately. Usually for relay autotuners the estimation of the static gain is not crucial, since the required information is the behavior close to the critical frequency. However, in the tuning method selected, the static gain matrix plays an important role and hence there was a need for better estimates. Another option would be to modify the tuning method, but since we wanted to get models that would be useful independent of tuning method, improving the model accuracy was considered a better option. To increase the excitation at low frequencies the relay amplitudes were changed in the middle of the experiment. This step change in relay output allowed for better static gain estimations. It also made the experiment duration a little longer, instead of using one oscillation period as in the NOMAD autotuner, the multi-NOMAD requires 5 relay switches before terminating. Worth mentioning is also that the experiment is terminated when both loops have completed 5 relay switches, which implies that one loop may have completed more switches at that time. However, that is still a short experiment, comparable or shorter than the ones that have to wait for convergence of limit cycles. In Paper VI the preliminary model identification method from Paper V was exchanged for a version with the same functionality that the SISO version NOMAD has. The multi-NOMAD was also equipped with a filter design and a MIMO PID controller optimization-based tuning method from [Boyd et al., 2016]. The resulting autotuner implementation was tried on a quadruple tank process to evaluate its applicability in practice with satisfactory results.

The multi-NOMAD provides FOTD or SOTD models for all elements of the transfer function matrix. Depending on chosen controller structure and tuning method this could be considered unnecessary, as it was by [Shen and Yu, 1994] in their sequential design method. To design a full multivariable controller this information is usually necessary, while a decentralized (diagonal) PID controller may not use all of it. If a decoupling scheme is used one may only need static information of the cross-couplings and not the full dynamics. However, since it is possible to get all the dynamics from one single experiment, and it is desirable with flexibility in controller choices, the author considers this to be a benefit of the multi-NOMAD.

In the work and evaluation of the multi-NOMAD it is assumed the input-output pairing is already at place. However, since all cross-couplings are excited and regarded in the estimation, it is not very sensitive to an inaccurate pairing. As long as the system is oscillating the information can be used. If the experiment does not induce the wanted oscillations, the pairing should probably be changed and the experiment redone.

3.5 Why Three Autotuners?

To have one autotuner for MIMO systems and another for SISO systems may not be that controversial, but why do we have two different SISO versions? As is seen

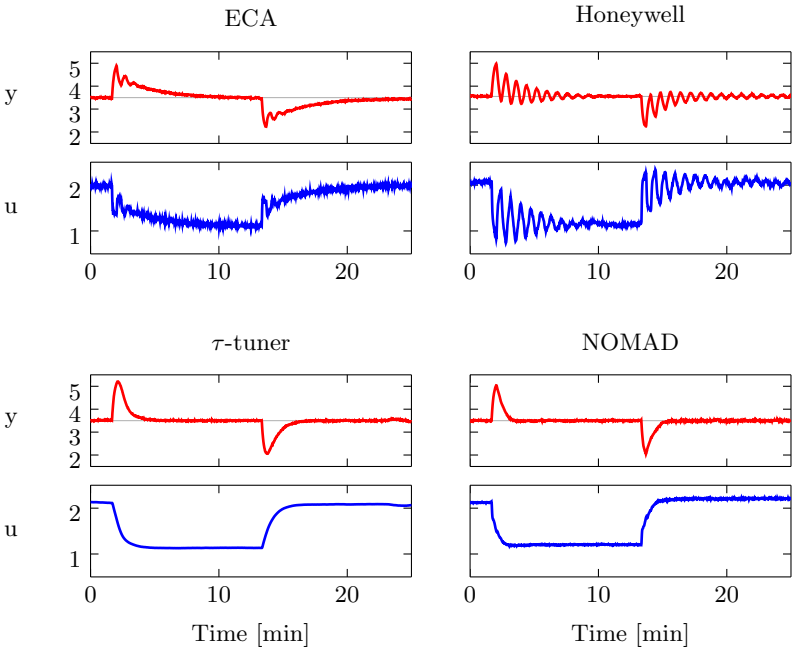


Figure 3.2 Performance comparison of load disturbance attenuation between two industrial autotuners, the τ-tuner and NOMAD. Figure from Paper IV.

in Figure 3.1, already in the work with the τ-tuner the thought appeared of a more advanced version, giving better models and controllers for those processes that benefit from it. In addition to the development of the τ-tuner we therefore thought about what changes we could do to get even better performance. The main objectives were to get better models and be able to use stronger tuning methods. The NOMAD autotuner was developed with this in mind. To evaluate the performance of the simple τ-tuner and the more advanced NOMAD autotuner, and to compare them both to industrial standard autotuners, we performed the study in Paper IV. The results from the study show that the τ-tuner gives reasonable controllers for all processes, but does not provide the same performance as NOMAD for lag-dominated systems. This can be expected by looking at the decision scheme in Figure 3.1 and the discussions about the normalized time delay in Paper I, where it is claimed that lag-dominated processes may gain a lot in performance by more advanced modeling. For the delay-dominated process, shown in Figure 3.2, the performance difference is not so large, and the τ-tuner is outperforming the two industrial autotuners for this process.

While the NOMAD provides better models, better controller tuning and also provides a shorter and more robust experiment, the τ-tuner has the benefit of being

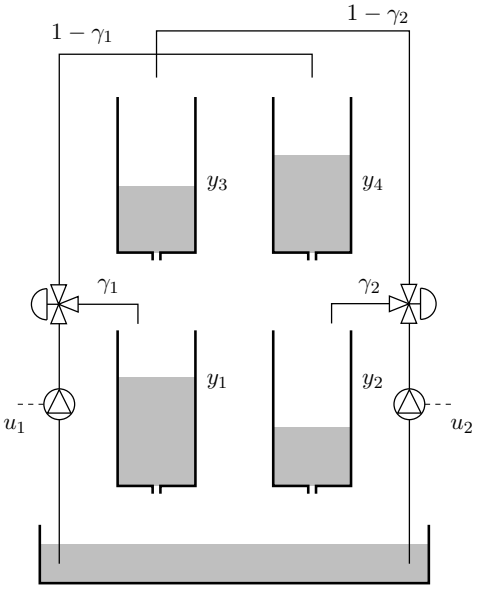


Figure 3.3 Schematics of the quadruple tank described in [Johansson, 2000].

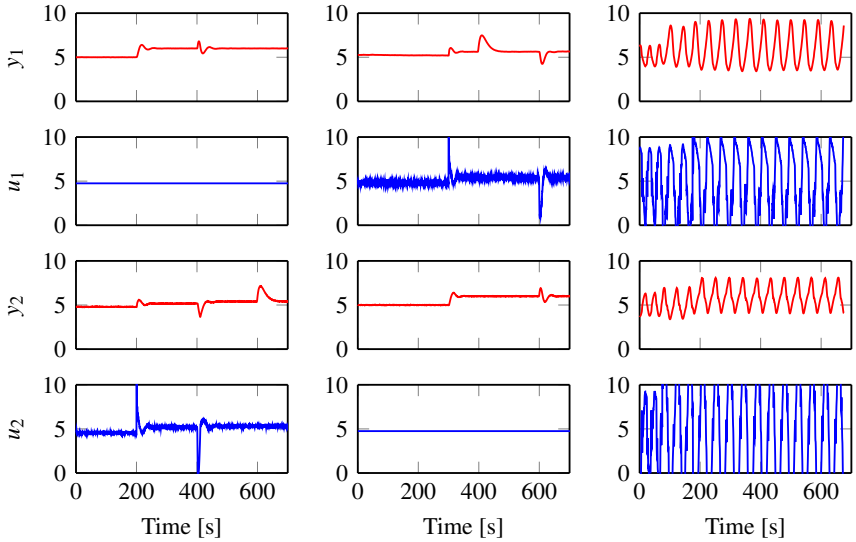
easy to implement and understand. Hence the τ-tuner is a good option if the extra computations required by the NOMAD are not possible to include in the control system.

The need of a specific autotuner for multivariable systems can be understood by considering the following example. A quadruple tank, with schematics seen in Figure 3.3, is used in its non-minimum phase configuration, that is, when most of the water enters through the upper tanks [Johansson, 2000]. The most appropriate input-output pairing, which is $u_1 - y_2$ and $u_2 - y_1$, was chosen before the experiment, and each loop was tuned independently by the NOMAD autotuner. The resulting control performance is shown in Figure 3.4.

As can be seen the results for each individual loop are really good while the other loop is in manual, but when both loops are run at the same time they counteract each other, causing large oscillations in the system. If the interactions in the multivariable system are small you may get decent results also from independent tuning, but if they are large you can get this type of undesirable behavior.

3.6 Additional Remarks

One of the major issues for process control is to obtain control-relevant process models at a low cost [Craig et al., 2011]. Depending on the process complexity and



(a) Controlling y_1 (b) Controlling y_2 (c) Controlling both loops

Figure 3.4 Performance results for the quadruple tank tuned as two separate SISO loops by the NOMAD autotuner. Note that this system is paired $u_1 - y_2$ and $u_2 - y_1$. At $t = 200$ a step is made in the setpoint for y_1 and at $t = 300$ a step is made in the setpoint for y_2. Load disturbances enter at $t = 400$ and $t = 600$. When the loops are run one by one, with the other controller in manual, the results are fine. When both loops run simultaneously, the controllers interact with each other, causing unstable behavior of the system.

how advanced control algorithms that should be used, the proposed autotuners will not be the solution for all, but they are sufficient for many processes and may be a step in the right direction for others. Especially the NOMAD autotuner showed very promising results in the comparative study in Paper IV and will hopefully be made into an industrial product.

All the proposed autotuners strive to be simple and user-friendly. The user should not need to make extensive settings before any experiment can be run, and neither should she or he have to understand in detail how things are done in the autotuner. It is, however, beneficial if the main principles are easy to grasp so that it does not feel like the autotuner performs some kind of magic. The user has to be able to trust the autotuner in making the appropriate decisions.

The selection of controller tuning methods has not been the main focus for the work in this thesis. Since all the proposed autotuners provide models of the processes, a change of tuning method is straightforward if desired.

Apart from the use for the autotuner in a process industry setting, there is also a

need of including autotuning functionality in large simulation environments. When simulating large systems with many sensors and actuators the focus is often on a few critical control loops, while a number of auxiliary variables have to be kept at constant values. The auxiliary signals can be controlled by simple feedback loops with PID controllers. Use of automatic tuning of the auxiliary loops can simplify the work significantly. The autotuner can also be beneficial when there are hidden or complicated dynamics between the input and output of the simulation model. The autotuner can then find a (simplified) model that can be used for controller design. The issue of autotuning for simulation environments was targeted in the Master's Thesis by [Björk and Levenhammar, 2017], where an implementation of the τ-tuner was made in the Modelica/Dymola environment. Evaluations were made on a few thermodynamic processes, which represented the type of systems that could be encountered in process industry, but also on for instance an airplane model with unstable dynamics. All these models were non-linear and of higher orders, still the controllers obtained from the autotuner achieved good performance.

4

Future Work

The main thing that could be done as future work is to try to get the proposed autotuners into industrial products. In order to do that, all optimizations should be implemented in an efficient way. The problem with local minima and initialization of the model identification method should be solved more satisfactorily. Another thing is that all parameter choices should be evaluated further. For instance, the evaluations in Paper IV indicated that the threshold value for when the τ-tuner classified a process as integrating, i.e., the α in Figure 3.1, was set a bit too high in Paper I. During implementation and experimental evaluation more practical issues usually need to be handled. One thing that needs to be regarded is how and when to stop the tuning phase and change to new controller parameters in a smooth way. Also, the controller implementations should include anti-windup strategies and setpoint weighting, and it would be beneficial if the autotuner could provide the parameters required for these. If the autotuners should be used in production it is also important to consider their user-friendliness. Suitable interfaces and what information the user wants to retrieve, are some aspects that need to be developed and evaluated in any industrial implementation of the autotuner.

For all three autotuner versions additional evaluation of controller designs and filter designs could be performed. The autotuners could also be evaluated on additional process types, like for instance unstable systems. Extensive studies on what type of systems that can be handled satisfactory by the autotuners are always beneficial, even though it will never take away the need of user evaluation of the results. It would also be interesting to see if the multi-NOMAD can be scaled to larger dimensions, or if the experiment has to be modified further to provide enough excitation for that.

In the implementation of the τ-tuner in a simulation environment in [Björk and Levenhammar, 2017], the problem of how to bring the system to its desired working point before the experiment starts was addressed. In an industrial setting the process is usually already at its working point, or brought there by manual interaction of the user, but for the simulation models it was desired to have an automated procedure. Some attempts on solving this issue was made, but this is a subject that should be investigated further.

Bibliography

Akaike, H. (1974). "A new look at the statistical model identification". *IEEE transactions on automatic control* **19**:6, pp. 716–723.

Åström, K. J. and T. Bohlin (1966). "Numerical identification of linear dynamic systems from normal operating records". In: *Theory of self-adaptive control systems*. Springer, pp. 96–111.

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 (2001). "The future of PID control". *Control Engineering Practice* **9**:11, pp. 1163–1175.

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.

Berner, J. (2015). *Automatic Tuning of PID Controllers based on Asymmetric Relay Feedback*. Licentiate Thesis ISRN LUTFD2/TFRT--3267--SE. Dept. Automatic Control, Lund University, Sweden.

Björk, M. and R. Levenhammar (2017). *Relay Auto-tuners in Modelica*. MSc Thesis ISRN LUTFD2/TFRT--6032--SE. Available at `www.control.lth.se/publications`. Dept. Automatic Control, Lund University, Sweden.

Boyd, S., M. Hast, and K. J. Åström (2016). "MIMO PID tuning via iterated LMI restriction". *International Journal of Robust and Nonlinear Control* **26**:8, pp. 1718–1731.

Campestrini, L., P. R. Barros, and A. S. Bazanella (2006). "Auto-tuning of PID controllers for MIMO processes by relay feedback". *IFAC Proceedings Vol.* **39**:2, pp. 451–456.

Chidambaram, M. and V. Sathe (2014). *Relay Autotuning for Identification and Control*. Cambridge University Press.

Craig, I., C. Aldrich, R. Braatz, F. Cuzzola, E. Domlan, S. Engell, J. Hahn, V. Havlena, A. Horch, B. Huang, et al. (2011). "Control in the process industries". In: Samad, T. et al. (Eds.). *The impact of control technology*. Available at `www-ieeecss.org`. IEEE Control Systems Society.

Dahlin, E. (1968). "Designing and tuning digital controllers". *Instruments and Control systems* **41**:6, pp. 77–83.

De Keyser, R., O. L. Joita, and C. M. Ionescu (2012). "The next generation of relay-based PID autotuners (part 2): a simple relay-based PID autotuner with specified modulus margin". *IFAC Proceedings Volumes* **45**:3, pp. 128–133.

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

Ender, D. B. (1993). "Process control performance: not as good as you think". *Control Engineering* **40**:10, pp. 180–190.

Friman, M. and K. V. Waller (1997). "A Two-Channel Relay for Autotuning". *Industrial and Engineering Chemistry Research* **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 Congress Seoul, Korea*.

Garpinger, O., T. Hägglund, and K. J. Åström (2014). "Performance and robustness trade-offs in PID control". *Journal of Process Control* **24**:5, pp. 568–577.

Hast, M., K. Åström, B. Bernhardsson, and S. Boyd (2013). "PID design by convex-concave optimization". In: *2013 European Control Conference (ECC)*. IEEE, pp. 4460–4465.

Ionescu, C. M., A. Maxim, C. Copot, and R. De Keyser (2016). "Robust PID autotuning for the quadruple tank system". *IFAC-PapersOnLine* **49**:7, pp. 919–924.

Johansson, K. H. (2000). "The quadruple-tank process: a multivariable laboratory process with an adjustable zero". *IEEE Transactions on control systems technology* **8**:3, pp. 456–465.

Kano, M. and M. Ogawa (2010). "The state of the art in chemical process control in Japan: Good practice and questionnaire survey". *Journal of Process Control* **20**:9, pp. 969–982.

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

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

Koo, D. G., H. C. Park, J. Y. Choi, and J. Lee (2004). "Sequential loop closing identification of multivariable processes using the biased relay feedback method". *Chemical Engineering Communications* **191**:5, pp. 611–624. DOI: `10.1080/00986440490424574`.

Leva, A., C. Cox, and A. Ruano (2002). "Hands-on PID autotuning: a guide to better utilisation". *IFAC Professional Brief*, pp. 1–84.

Leva, A. (1993). "PID autotuning algorithm based on relay feedback". In: *IEE Proceedings D-Control Theory and Applications*. Vol. 140. 5. IET, pp. 328–338.

Li, W., E. Eskinat, and W. L. Luyben (1991). "An improved autotune identification method". *Industrial and Engineering Chemistry Research* **30**:7, pp. 1530–1541. DOI: 10.1021/ie00055a019.

Lin, C., Q.-G. Wang, and T. H. Lee (2003). "Local stability of limit cycles for MIMO relay feedback systems". In: *Control and Automation, 2003. ICCA'03. Proceedings. 4th International Conference on*. IEEE, pp. 937–941.

Lin, C., Q.-G. Wang, and T. H. Lee (2004). "Relay Feedback: A Complete Analysis for First-Order Systems". *Industrial and Engineering Chemistry Research* **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". *Industrial and Engineering Chemistry Research* **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". *Journal of 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.

Loh, A. P., C. C. Hang, C. K. Quek, and V. U. Vasnani (1993). "Autotuning of multiloop proportional-integral controllers using relay feedback". *Industrial and Engineering Chemistry Research* **32**:6, pp. 1102–1107. ISSN: 0888-5885. DOI: 10.1021/ie00018a017.

Loh, A. and V. Vasnani (1994). "Necessary conditions for limit cycles in multiloop relay systems". *IEE Proceedings-Control Theory and Applications* **141**:3, pp. 163–168.

Luyben, W. L. (1986). "Simple method for tuning SISO controllers in multivariable systems". *Industrial & Engineering Chemistry Process Design and Development* **25**:3, pp. 654–660.

Luyben, W. L. (1987). "Derivation of transfer functions for highly nonlinear distillation columns". *Industrial and Engineering Chemistry Research* **26**:12, pp. 2490–2495. DOI: 10.1021/ie00072a017.

Luyben, W. L. (2001). "Getting More Information from Relay-Feedback Tests". *Industrial and Engineering Chemistry Research* **40**:20, pp. 4391–4402. DOI: 10.1021/ie010142h.

Menani, S. and H. Koivo (2003). "New approach on the automatic tuning of multivariable PI controllers using relay feedback". *International Journal of Systems Science* **34**:2, pp. 93–110.

Menani, S. and H. Koivo (2001). "A comparative study of recent relay autotuning methods for multivariable systems". *International Journal of Systems Science* **32**:4, pp. 443–466.

Nikita, S. and M. Chidambaram (2017). "Case studies of improved relay auto-tuning of PID controllers for TITO systems". *Indian Chemical Engineer*, pp. 1–19.

O'Dwyer, A. (2009). *Handbook of PI and PID controller tuning rules*. 3rd Edition. Imperial College Press London. ISBN: 978-1-84816-242-6.

Palmor, Z. J., Y. Halevi, and T. Efrati (1995). "A general and exact method for determining limit cycles in decentralized relay systems". *Automatica* **31**:9, pp. 1333–1339.

Palmor, Z., Y. Halevi, and N. Krasney (1993). "Automatic tuning of decentralized PID controllers for TITO processes". *IFAC Proceedings Volumes* **26**:2, pp. 73–76.

Pereira, R. D., M. Veronesi, A. Visioli, J. E. Normey-Rico, and B. C. Torrico (2017). "Implementation and test of a new autotuning method for PID controllers of TITO processes". *Control Engineering Practice* **58**, pp. 171–185.

Samad, T. (2017). "A survey on industry impact and challenges thereof". *IEEE Control Systems* **37**:1, pp. 17–18. ISSN: 1066-033X. DOI: 10.1109/MCS.2016.2621438.

Samad, T. and A. Annaswamy, (Eds.) (2014). *The Impact of Control Technology*. 2nd edition. IEEE Control Systems Society. URL: http://ieeecss.org/general/IoCT2-report.

Segovia, V. R., T. Hägglund, and K. J. Åström (2014). "Measurement noise filtering for PID controllers". *Journal of Process Control* **24**:4, pp. 299–313.

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 (1996). "Use of biased-relay feedback for system identification". *AIChE Journal* **42**:4, pp. 1174–1180. DOI: 10.1002/aic.690420431.

Shen, S.-H. and C.-C. Yu (1994). "Use of relay-feedback test for automatic tuning of multivariable systems". *AIChE Journal* **40**:4, pp. 627–646.

Shinskey, F. G. (1996). *Process Control Systems: Application, Design and Tuning*. 4th ed. McGraw-Hill, Inc. ISBN: 0-07-057101-5.

Shinskey, F. G. (2002). "Process control: as taught vs as practiced". *Industrial and Engineering Chemistry Research* **41**:16, pp. 3745–3750.

Skogestad, S. (2003). "Simple analytic rules for model reduction and PID controller tuning". *Journal of 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". *Industrial and Engineering Chemistry Research* **45**:23, pp. 7817–7822.

Soltesz, K., P. Mercader, and A. Baños (2016a). "An automatic tuner with short experiment and probabilistic plant parameterization". *International Journal of Robust and Nonlinear Control*.

Soltesz, K., C. Grimholt, and S. Skogestad (2016b). "Simultaneous design of proportional–integral–derivative controller and measurement filter by optimisation". *IET Control Theory & Applications* **11**:3, pp. 341–348.

Theorin, A. and J. Berner (2015). "Implementation of an Asymmetric Relay Autotuner in a Sequential Control Language". In: *2015 IEEE International Conference on Automation Science and Engineering*, pp. 874–879. DOI: 10.1109/CoASE.2015.7294191.

Toh, W. and G. Rangaiah (2002). "A methodology for autotuning of multivariable systems". *Industrial and Engineering Chemistry Research* **41**:18, pp. 4605–4615.

VanDoren, V. (2006). "Auto-tuning control using Ziegler-Nichols". *Control Engineering* **53**:10, pp. 66–70.

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

Viswanathan, P. and G. Rangaiah (2000). "Process identification from closed-loop response using optimization methods". *Chemical Engineering Research and Design* **78**:4, pp. 528–541.

Wang, Y.-G., W.-J. Cai, and M. Ge (2003). "Decentralized relay-based multivariable process identification in the frequency domain". *IEEE transactions on automatic control* **48**:5, pp. 873–878.

Wang, Q.-G., C.-C. Hang, and Q. Bi (1997a). "Process frequency response estimation from relay feedback". *Control Engineering Practice* **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". *Industrial and Engineering Chemistry Research* **36**:2, pp. 375–381. DOI: 10.1021/ie960412+.

Wang, Q.-G. and Y. Zhang (2001). "A novel FFT-based robust multivariable process identification method". *Industrial and Engineering Chemistry Research* **40**:11, pp. 2485–2494.

Wang, Q.-G., B. Zou, T.-H. Lee, and Q. Bi (1997c). "Auto-tuning of multivariable PID controllers from decentralized relay feedback". *Automatica* **33**:3, pp. 319–330.

Yu, C.-C. (2006). *Autotuning of PID controllers: A relay feedback approach*. Springer Science & Business Media.

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

Paper I

Improved Relay Autotuning using Normalized Time Delay

Josefin Berner Tore Hägglund Karl Johan Åström

Abstract

The relay autotuner provides a simple way of finding PID controllers of sufficient performance. By using an asymmetric relay function the excitation of the process is improved. This gives better models, and hence a better tuning, without increasing the time consumption or complexity of the experiment. Some processes demand more accurate modeling and tuning to obtain controllers of sufficient performance. These processes can be singled out by their normalized time delays and be subject to further modeling efforts. The autotuner proposed in this paper provides a simple way of finding the normalized time delay from the experiment, and uses it for model and controller selection. The autotuner has been implemented and evaluated both in a simulation environment and by industrial experiments.

1. Introduction

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 not require an extensive control education for the users. One such method, which has been successful in industry, is the relay autotuner. The main advantages of the relay autotuner are 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. The 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 into PID control, has provided the possibility to improve the relay autotuner. The modification to find a low-order model from the relay experiment was proposed in [Luyben, 1987], where the static gain was assumed to be known and in [Li et al., 1991], where an additional relay experiment was performed. The relay autotuner proposed in this paper uses an asymmetric relay function to increase the excitation in the experiment. A version of the asymmetric relay function was used in [Shen et al., 1996], and later on investigated in e.g. [Kaya and Atherton, 2001], [Lin et al., 2004] and [Berner et al., 2014]. For a more thorough review of the advances in modeling from relay feedback experiments, see [Liu et al., 2013].

The asymmetric relay function gives better models without increasing the complexity or time consumption of the tuning procedure. A low-order transfer function model is obtained from the proposed autotuner, while the original autotuner only yields the gain and phase of 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 low-order models are 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 performance of the control loops is crucial.

2. Background

2.1 PID Tuning

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 a low-order 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.

The aim of the controlled system is to have good load disturbance attenuation, while being robust against process variations and measurement noise. In this paper the performance measure used is the integrated absolute error, or IAE-value, defined as

$$\text{IAE} = \int_0^\infty |e(t)|dt. \tag{1}$$

Here $e(t)$ is the error from a unit step change in the load. The robustness criterion used is

$$M_{ST} = \max(M_S, M_T) \tag{2}$$

where M_S and M_T are the maximum sensitivities, i.e., the largest absolute values, of the sensitivity function S and the complementary sensitivity function T respectively.

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 straightforward.

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 approximation of the process dynamics. One of the most common low-order model approximations is a first order system with time delay, or shortly an FOTD model, defined as

$$P(s) = \frac{K_p}{1 + sT} e^{-sL}. \tag{3}$$

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

$$P(s) = \frac{K_p}{(1 + sT_1)(1 + sT_2)} e^{-sL}. \tag{4}$$

Neither the FOTD model in (3) nor the SOTD model in (4) can be used to describe integrating processes. Therefore the integrating time delayed model, ITD model,

$$P(s) = \frac{k_v}{s} e^{-sL}, \tag{5}$$

and the integrating plus first order time delayed model, IFOTD model,

$$P(s) = \frac{k_v}{s(1+sT)} e^{-sL}, \tag{6}$$

will be used as alternatives for integrating processes.

2.3 Normalized Time Delay

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

$$\tau = \frac{L}{L+T}, \qquad 0 \le \tau \le 1. \tag{7}$$

The normalized time delay characterizes 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 one, 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 a process of higher order dynamics, the normalized time delay is given from the *apparent time constant* and *apparent time delay*. These are obtained from an FOTD model approximation of the process, obtained from step response analysis.

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 improvement for τ close to one. It is also shown that while an FOTD model is sufficient for controller tuning for processes with large τ, processes with small τ can gain a lot from more accurate modeling. The reason for this is that the true time delay gives a fundamental limitation, and the apparent time delay in the FOTD model is a combination of the true time delay and the neglected dynamics. The dynamics added to the time delay can make the difference between the true time delay and the apparent time delay quite large for lag-dominated systems. To be able to design a high-performance controller it is important to get as close to the true time delay as possible, hence the need of better modeling for those processes. This knowledge of τ is essential for making choices in the autotuner procedure, and will be discussed further in Section 6.

The idea of using 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 paper proposes a simpler method to find this information, which will be described in Section 4.

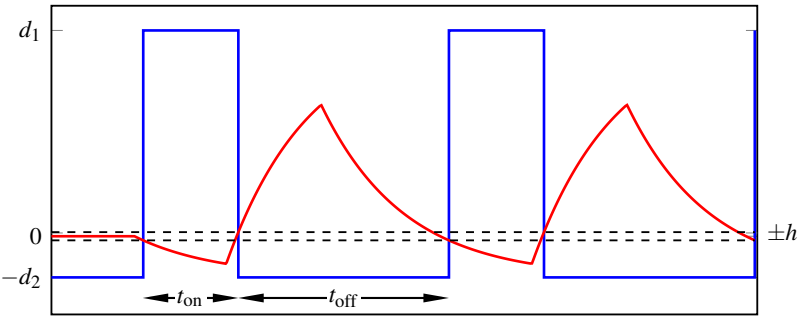


Figure 1. An example of the signals from the asymmetric relay feedback experiment. The relay output u is shown in blue, the process output y is shown in red. The black dashed lines show the hysteresis levels, $\pm h$. The relay output switches between u_{on} and u_{off} every time the process output leaves the hysteresis band.

3. Asymmetric Relay Feedback

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 for the autotuner in this paper is

$$u(t) = \begin{cases} u_{on}, & y(t) < y_0 - h, \\ u_{on}, & y(t) < y_0 + h, \quad u(t^-) = u_{on}, \\ u_{off}, & y(t) > y_0 - h, \quad u(t^-) = u_{off}, \\ u_{off}, & y(t) > y_0 + h, \end{cases} \tag{8}$$

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

$$u_{on} = u_0 + \text{sign}(K_p)d_1, \qquad u_{off} = u_0 - \text{sign}(K_p)d_2. \tag{9}$$

The name *asymmetric* relay reflects that the amplitudes d_1 and d_2 are not equal. This creates the asymmetric oscillations. The asymmetry level of the relay is denoted γ and defined as

$$\gamma = \frac{\max(d_1, d_2)}{\min(d_1, d_2)} > 1. \tag{10}$$

An illustrative example of the inputs and outputs of the asymmetric relay feedback is shown in Figure 1. The half-periods t_{on} and t_{off} are defined as the time intervals where $u(t) = u_{on}$ and $u(t) = u_{off}$ respectively.

The implementation of the relay feedback experiment contains features such as automatic choice of hysteresis level, detection of the sign of the process gain, soft startup and adaptive relay amplitude. Details about the implementation and parameter choices are found in [Berner, 2015].

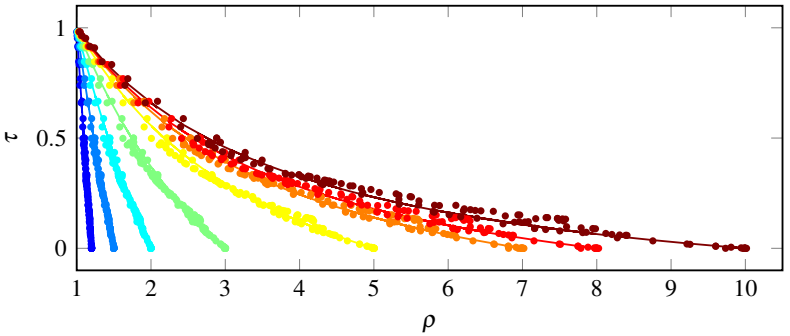


Figure 2. Validation results of the equation for τ, stated in (12). The figure shows the results for $\gamma = \{1.2, 1.5, 2, 3, 5, 7, 8, 10\}$ in different colors from left to right. The solid lines show τ-values calculated from (12), while the dots show the relation between ρ and the true τ-values for the processes in the test batch.

4. Estimation of Normalized Time Delay

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

$$\rho = \frac{\max(t_{\text{on}}, t_{\text{off}})}{\min(t_{\text{on}}, t_{\text{off}})}, \tag{11}$$

is related to the normalized time delay of the process. If the system is delay dominated, τ close to one, the time intervals will be more or less symmetrical even though the amplitudes are asymmetric. When the process is lag dominated, i.e., if τ is small, the half-period ratio instead reflects the asymmetry of the amplitudes. This was shown for FOTD processes under asymmetric relay feedback with no hysteresis in [Berner et al., 2014]. Results which are only valid for FOTD processes and a relay without hysteresis are of limited practical use. However, the observation is valid for a wide range of process types. Figure 2 shows simulation results for a test batch [Åström and Hägglund, 2006] consisting of 134 different processes typical for the process industry. 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 $\tau(\rho = \gamma, \gamma) = 0$. The result is the following equation for the normalized time delay

$$\tau(\rho, \gamma) = \frac{\gamma - \rho}{(\gamma - 1)(0.35\rho + 0.65)}. \tag{12}$$

The equation was validated against the test batch, for some different asymmetry levels γ, and the results are shown in the solid lines in Figure 2.

The errors in determining τ using (12) are shown in Figure 3 for $\gamma = 2$. For all processes in the batch, the estimate stays within 8 % of the correct value, and

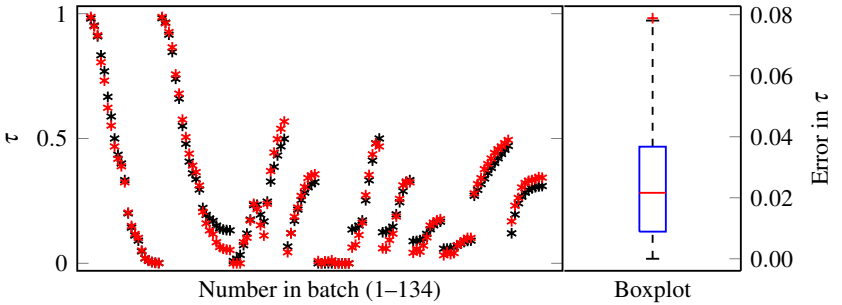


Figure 3. Results of τ-estimation for the processes in the test batch. The left plot shows estimated τ in red, and the true values in black. The right plot shows a boxplot of the absolute errors of the τ-estimation.

the median error is about 2 %. The obtained results are accurate enough to use the estimated τ to classify the process, and to use it as an information source for decision-making in the autotuner.

5. Modeling

Once the experiment is performed we want to find the parameter values for the model structures listed in Section 2.2. Modeling from an asymmetric relay experiment can be done in many ways. Some examples are by using the describing function as in [Shen et al., 1996], by using the A-locus method as in [Kaya and Atherton, 2001], by using the relation between the Fourier series coefficients and the model parameters as in e.g. [Srinivasan and Chidambaram, 2003] or by using a curve fitting approach as in e.g. [Liu and Gao, 2008]. For additional relevant references on different modeling strategies, see [Liu et al., 2013].

The modeling in this work is based on a curve fitting approach, and the focus has been on finding simple, intuitive equations that use measurements robust to noisy data. To find the FOTD and ITD models we use equations where the only measurements needed are the durations t_{on} and t_{off}, and the integral of the process output, I_y, defined as

$$I_y = \int_{t_p} \big(y(t) - y_0\big)\, dt. \tag{13}$$

Here $t_p = t_{\text{on}} + t_{\text{off}}$ is the period time of the oscillation and y_0 is the stationary operation point we started the experiment at. 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 4, and the

integral of the relay output I_u, which analogously to I_y is defined as

$$I_u = \int_{t_p} \big(u(t) - u_0\big)\, dt. \tag{14}$$

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

$$I_u = (u_{\text{on}} - u_0)t_{\text{on}} + (u_{\text{off}} - u_0)t_{\text{off}}. \tag{15}$$

5.1 FOTD Models

The FOTD model defined in (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

$$K_p = \frac{I_y}{I_u}. \tag{16}$$

Note that this does not apply to the symmetric relay, where I_u would always be zero. It follows from (15) that I_u can become zero with the asymmetric relay as well, but only if $t_{\text{off}}/t_{\text{on}} = d_1/d_2$. This means that $\rho = \gamma$, which implies that $\tau = 0$, and for those processes we will use the ITD model.

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

$$t_{\text{on}} = T \ln \frac{h/|K_p| - d_2 + e^{L/T}(d_1 + d_2)}{d_1 - h/|K_p|}, \tag{17}$$

$$t_{\text{off}} = T \ln \frac{h/|K_p| - d_1 + e^{L/T}(d_1 + d_2)}{d_2 - h/|K_p|}, \tag{18}$$

given in [Berner, 2015]. Since K_p can be found from (16), the results in (17) and (18) 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 4, which gives the ratio between L and T as

$$L/T = \frac{\tau}{1 - \tau}. \tag{19}$$

Knowing this ratio, T can be found from either of the two equations (17) or (18), or from an average of both. With T known, it is straightforward to get L from (19).

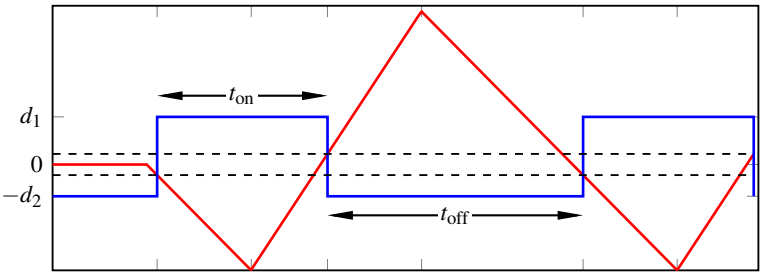


Figure 4. An example of the signals from a relay experiment with an ITD process. The blue line shows the relay output u, the red line shows the process output y. The dashed black lines show the hysteresis. Note the triangular shape of y that is characteristic for an ITD process.

5.2 ITD Models

An integrating process on the form

$$P(s) = \frac{k_v}{s} e^{-sL} \tag{20}$$

can be written as the differential equation

$$\dot{y}(t) = k_v u(t - L). \tag{21}$$

Since $u(t)$ is piecewise constant, so is $\dot{y}(t)$, and hence the shape of y will be triangular, see Figure 4. By considering the output curves, equations for k_v and L can be obtained, see [Berner, 2015] for full derivation. The equations are

$$k_v = \frac{2I_y}{t_{on}t_{off}(u_{on} + u_{off})} + \frac{2h}{u_{on}t_{on}}, \tag{22}$$

$$L = \frac{u_{on}t_{on} - 2h/k_v}{u_{on} - u_{off}}. \tag{23}$$

5.3 SOTD and IFOTD Models

To obtain the somewhat more advanced SOTD and IFOTD models we use the entire experiment data set. The model parameters are estimated from a system identification method based on Newton's method, as in [Berner et al., 2014]. To assure convergence in the iterative method, appropriate initial parameter values are needed. These are obtained from the calculated FOTD or ITD model.

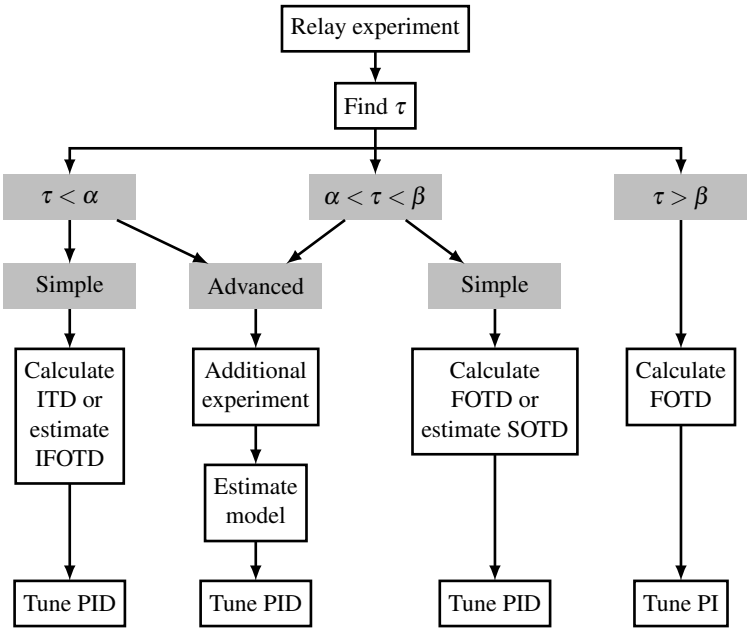


Figure 5. Decision scheme based on the estimated normalized time delay.

6. Tuning Procedure

6.1 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 based on the normalized time delay, τ. The resulting decision scheme is shown in Figure 5. If τ is close to one, it has been shown [Åström and Hägglund, 2006], that an FOTD model is sufficient to describe the process for a control purpose. If τ is smaller, higher-order models can give significantly better results, motivating estimation of an SOTD model. If τ is really small, the time constant is much larger than the time delay. The process can then be considered an integrating process, which implies that an ITD or IFOTD model should be estimated. In this autotuner implementation the limits α and β, shown in Figure 5, are $\alpha = 0.1$ and $\beta = 0.6$.

The low-order models defined in Section 2.2 are obtained as described in Section 5. If it is crucial that we get a really good model we might consider estimating higher-order models. However, that implies that we may need an additional experiment to get even better excitation. This is illustrated in the advanced branch in Figure 5. Information from the relay experiment already performed, can be used to design the additional experiment.

6.2 Controller Design

The choice of controller design is restricted to PID controllers. The low-order models in Section 2.2 were chosen since there exists simple tuning rules for them. This implementation of the autotuner uses the AMIGO tuning rules [Åström and Hägglund, 2006], but it could easily be changed to another tuning rule if desired. If the advanced branch is used to find higher-order models, there are no simple rules, and the PID tuning would instead need to be performed through for example the optimization method in [Garpinger and Hägglund, 2008].

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 close to one. Therefore a PI controller is tuned for large τ, and a PID controller otherwise as shown in Figure 5.

7. Examples

To demonstrate the results of the autotuner we consider the three processes

$$\begin{aligned} P_1(s) &= \frac{1}{(s+1)(0.1s+1)(0.01s+1)(0.001s+1)}, \\ P_2(s) &= \frac{1}{(s+1)^4}, \\ P_3(s) &= \frac{1}{(0.05s+1)^2}\mathrm{e}^{-s}, \end{aligned} \tag{24}$$

where P_1 is lag dominated, P_2 balanced, and P_3 delay dominated. All simulations in this section have been performed with the Matlab/Simulink implementation of the autotuner described in [Berner, 2015].

The experiment output for the three processes are shown in Figure 6. Some implementation features like the adaptive relay amplitude and soft startup are clearly visible in the figure. It is also worth noting the difference in half-period ratios. For the upper (lag-dominated) process the difference between t_{on} and t_{off} is large, while for the lower (delay-dominated) process the time intervals are more or less equal. For P_1 the normalized time delay is calculated to $\tau = 0.04$. Since τ is so small it corresponds to the left branch in Figure 5 and the choice of the autotuner is to calculate an ITD model or estimate an IFOTD model. P_2 has $\tau = 0.37$ and ends up in the middle branch. For P_3 $\tau = 0.93$, which puts it in the right branch and indicates that an FOTD model describes the process sufficiently. However, for comparison reasons an estimated SOTD model is presented as well. The resulting model parameters are listed in Table 1.

Since the most interesting part is not the models in themselves, but rather how good the controllers obtained from these models are, a comparison of five different controllers were made for each process. PI controllers were tuned for the FOTD/ITD

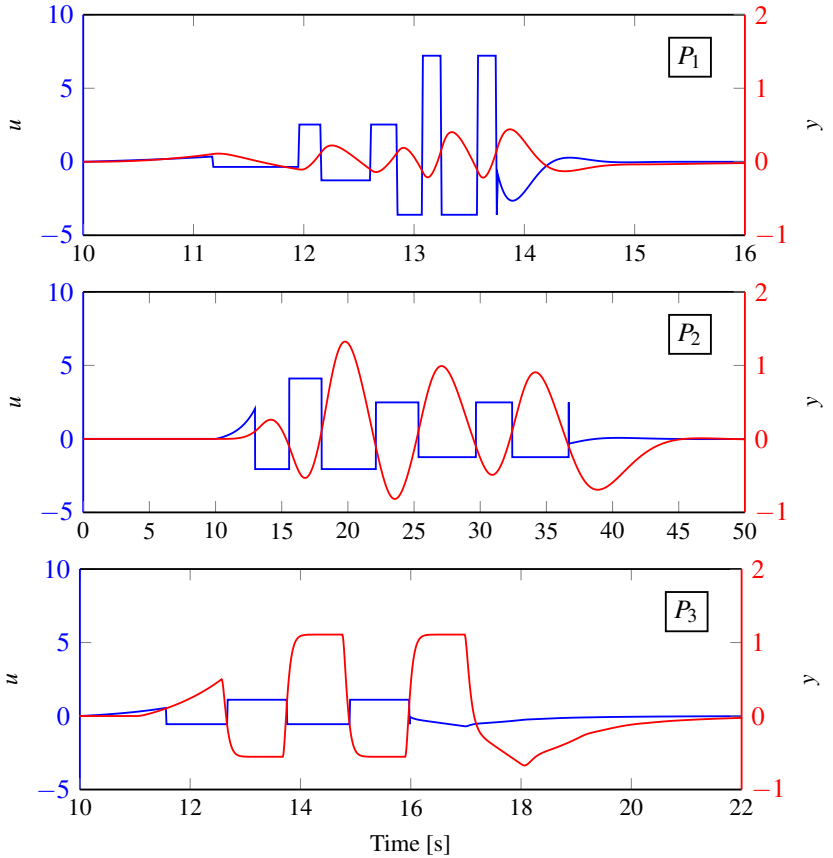


Figure 6. Signals from the relay experiment for P_1 (top), P_2 (middle) and P_3 (bottom). The blue lines show the relay output u, and the red lines show the process output y. Note the different scales on the axes.

Table 1. Resulting model parameters

	Model	k_v	K_p	$T_{(1)}$	T_2	L
P_1	ITD relay	0.73				0.09
$\tau = 0.04$	IFOTD est	0.68		0.04		0.04
P_2	FOTD relay		0.99	3.23		1.89
$\tau = 0.37$	SOTD est		1.05	1.76	1.76	1.00
P_3	FOTD relay		1.00	0.08		1.04
$\tau = 0.93$	SOTD est		1.00	0.05	0.05	1.00

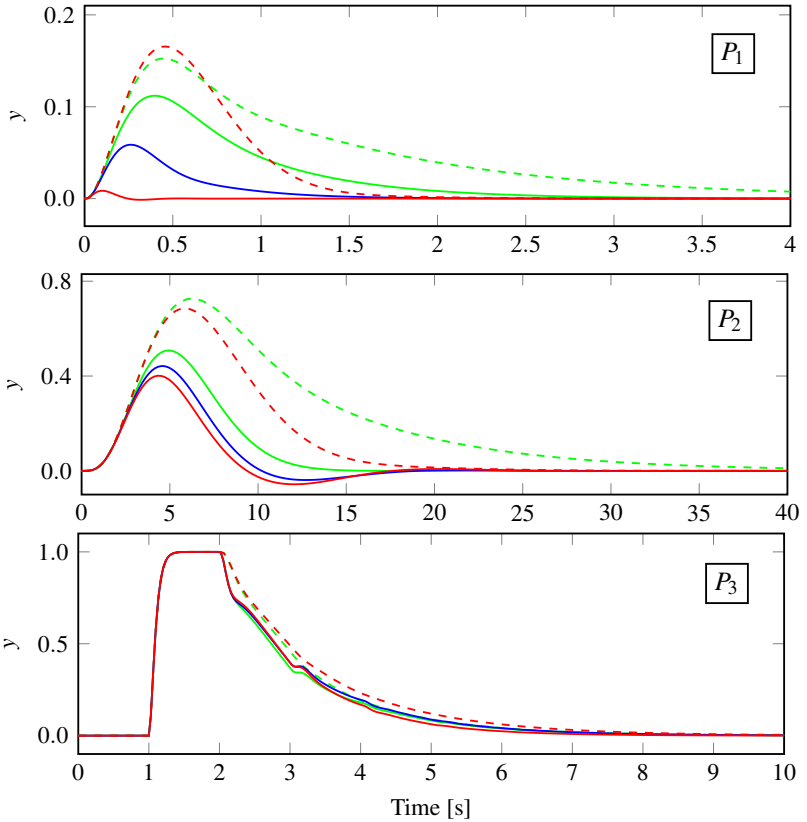


Figure 7. Step responses from a load disturbance on the process input, shown for the closed-loop systems containing the true processes and the different obtained controllers. The green curves show controllers tuned for the ITD/FOTD models, blue shows controllers tuned for IFOTD/SOTD models, red shows controllers tuned for the true processes. The dashed lines indicate the PI controllers, while the solid lines show the PID controllers.

model and the true process. PID controllers were tuned for the FOTD/ITD model, the SOTD/IFOTD model and the true process. The controllers based on models were obtained using the AMIGO rules, while the controllers from the true processes were obtained using the optimization method described in [Garpinger and Hägglund, 2008] where IAE is minimized with the constraint that $M_{ST} \leq 1.4$. The obtained controller parameters, performance and robustness measures are listed in Table 2. Control performances for a step in load disturbance are illustrated in Figure 7.

The results verify the statements made for small values of τ. The derivative part is beneficial, the PID controllers perform much better than the PI controllers, and

Table 2. Controller parameters.

	Controller	K	T_i	T_d	M_{ST}	IAE
	ITD PI	5.48	1.18		1.34	0.215
	ITD PID	7.04	0.70	0.04	1.15	0.100
P_1	IFOTD PID	15.3	0.47	0.04	1.23	0.031
	Optimal PI	4.20	0.49		1.40	0.118
	Optimal PID	89.5	0.09	0.05	1.40	0.001
	FOTD PI	0.36	3.02		1.24	8.487
	FOTD PID	0.98	2.85	0.80	1.35	2.906
P_2	SOTD PID	1.19	2.35	1.11	1.37	2.348
	Optimal PI	0.43	2.25		1.39	5.208
	Optimal PID	1.33	2.11	1.34	1.40	2.134
	FOTD PI	0.17	0.37		1.44	2.158
	FOTD PID	0.24	0.48	0.11	1.41	2.014
P_3	SOTD PID	0.22	0.45	0.13	1.40	2.069
	Optimal PI	0.16	0.37		1.40	2.313
	Optimal PID	0.20	0.40	0.14	1.40	1.988

better modeling can increase the performance significantly. For the lag-dominated process P_1, 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 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.

For the delay-dominated system on the other hand it is clear that neither the derivative part nor more advanced modeling gives better performance than a PI controller tuned from an FOTD model.

8. Industrial Experiment

The autotuner was implemented and tested on an air handling unit provided by Schneider Electric Buildings AB in Malmö, Sweden. The implementation was made in Schneider Electric's software StruxureWare Building Operation. The implementation uses the simplest version of the autotuner, where the experiment data is used to find an FOTD or ITD model, and parameters for a PI/PID controller are tuned by the AMIGO rules.

Pressure in an air duct was controlled by changing the speed of a supply air fan, positioned before the duct. The control signal was normalized to a percentage of the

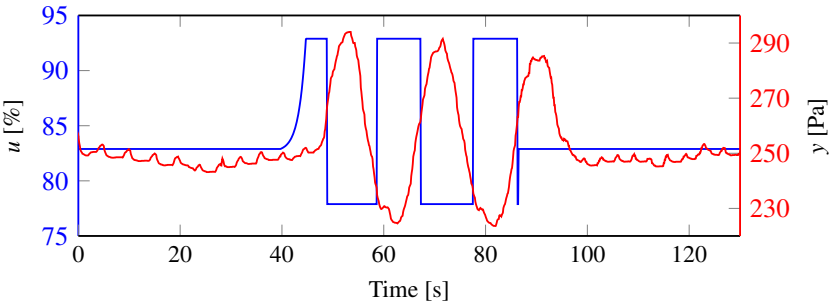


Figure 8. Experiment data from the pressure control loop. The normalized fan speed is shown in blue, the pressure measurement in red. Note the different scales and units on the axes.

full speed of the fan, while the pressure was measured in Pascal. The reference value of the pressure was set to 250 Pa. A relay experiment performed on the system is shown in Figure 8. The asymmetry level, convergence limit, maximum and minimum deviations and the maximum relay deviation, were set according to the default values in [Berner, 2015]. The sample time used during the experiment was $t_s = 0.1$ s.

The experiment started with 40 s measurement of the noise. The figure shows that the signal is noisy, in this experiment the noise was measured to 13 Pa peak to peak. The experiment converges within 45 s, or two and a half oscillation periods, which is fast for this process. The normalized time delay calculated from the experiment was $\tau = 0.77$. Since it was large, an FOTD model was estimated and a PI controller selected. Calculation of the FOTD model parameters, as in Section 5, gave $K_p = 2.29$, $T = 1.92$ and $L = 6.31$. The obtained controller parameters were $K = 0.088$, $T_i = 2.92$. These parameters were used to investigate the control performance. The controller was already present in the system. It had a sampling time of 1 s, and a dead zone of 5 Pa. Results from step changes in the reference value are shown in Figure 9. The step response results are satisfactory. There is an overshoot, but it can be reduced by filtering the setpoint. The dead zone is clearly visible through the long periods of constant control signal, despite 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 10. This also shows satisfactory results. The effect of the load disturbances is removed completely in approximately 20-25 s with rather small overshoots.

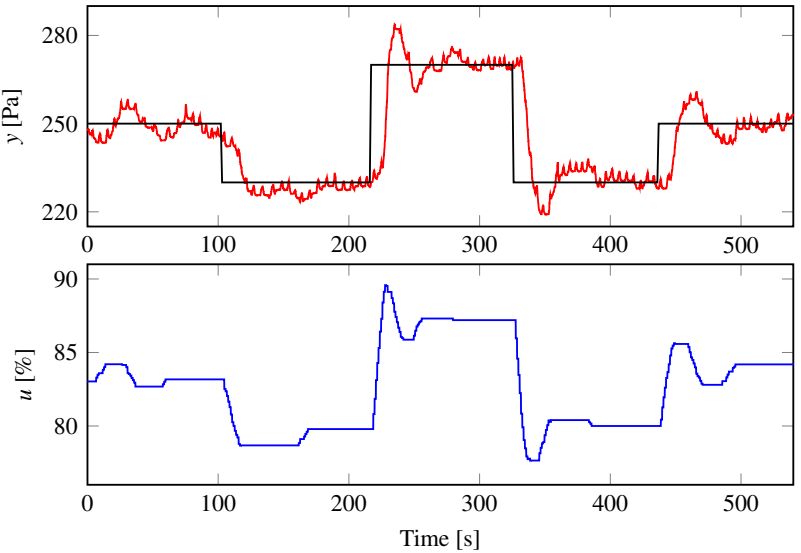


Figure 9. Response to setpoint changes for the system with the controller tuned from the experiment. The upper plot shows the measured pressure in red, and the setpoint in black. The lower plot shows the control signal.

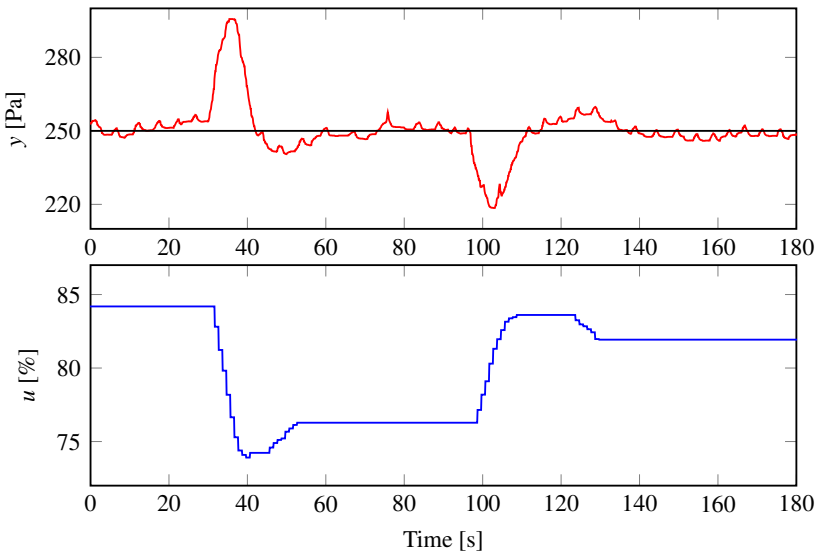


Figure 10. Response to load disturbances. The upper plot shows the measured pressure in red, and the setpoint in black. The lower plot shows the control signal.

9. Conclusions

This paper shows that the asymmetric relay autotuner gives good results in both simulations and real experiments. The asymmetric relay feedback experiment provides an easy way of finding the normalized time delay. The results in the example section clearly 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, together with better tuning can be very useful for processes with a small normalized time delay.

From the experimental results it is concluded that the relay autotuner works satisfactory also in practice. Despite a noisy signal, a model of the process was obtained fast and accurately. The industrial implementation only contained the most simple version of the autotuner, and should be extended with at least the possibility to estimate SOTD and IFOTD models. Different tuning methods could also be considered.

Acknowledgment

The authors are members of the LCCC Linnaeus Center and the ELLIIT Excellence Center at Lund University.

References

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.

Berner, J. (2015). *Automatic Tuning of PID Controllers based on Asymmetric Relay Feedback*. Licentiate Thesis ISRN LUTFD2/TFRT--3267--SE. Dept. Automatic Control, Lund University, Sweden.

Berner, J., K. J. Åström, and T. Hägglund (2014). "Towards a New Generation of Relay Autotuners". *IFAC Proceedings Volumes* **47**:3, pp. 11288–11293.

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

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

Li, W., E. Eskinat, and W. L. Luyben (1991). "An improved autotune identification method". *Industrial and Engineering Chemistry Research* **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". *Industrial and Engineering Chemistry Research* **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". *Industrial and Engineering Chemistry Research* **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". *Journal of Process Control* **23**:10, pp. 1597–1623. DOI: 10.1016/j.jprocont.2013.08.003.

Luyben, W. L. (1987). "Derivation of transfer functions for highly nonlinear distillation columns". *Industrial and Engineering Chemistry Research* **26**:12, pp. 2490–2495. DOI: 10.1021/ie00072a017.

Luyben, W. L. (2001). "Getting More Information from Relay-Feedback Tests". *Industrial and Engineering Chemistry Research* **40**:20, pp. 4391–4402. DOI: 10.1021/ie010142h.

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 (1996). "Use of biased-relay feedback for system identification". *AIChE Journal* **42**:4, pp. 1174–1180. DOI: 10.1002/aic.690420431.

Skogestad, S. (2003). "Simple analytic rules for model reduction and PID controller tuning". *Journal of 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". *Industrial and Engineering Chemistry Research* **45**:23, pp. 7817–7822.

Srinivasan, K. and M. Chidambaram (2003). "Modified relay feedback method for improved system identification". *Computers & chemical engineering* **27**:5, pp. 727–732.

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

Paper II

Asymmetric Relay Autotuning – Practical Features for Industrial Use

Josefin Berner Tore Hägglund Karl Johan Åström

Abstract

The relay autotuner provides a simple way of finding PID controller parameters. Even though relay autotuning is much investigated in the literature, the practical aspects are not that well-documented. In this paper an asymmetric relay autotuner with features such as a startup procedure and adaptive relay amplitudes is proposed. Parameter choices and handling of noise, disturbances, start in non-steady state and other possible error sources are discussed. The autotuner is implemented and tested on an industrial air handling unit to show its use in practice. The experiments show good results, and prove that the proposed simple autotuner is well-suited for industrial use. But the experiments also enlighten possible error sources and remaining problems.

Keywords: Asymmetric Relay Autotuning, PID Control, Practical features, Industrial experiment, Air handling unit

1. Introduction

In an industrial process facility with hundreds of control loops, the benefits of a simple, reliable, automatic way of tuning the controllers are obvious. The relay autotuner has proven to be a good candidate for automatic tuning of PID controllers. Some advantages of the relay autotuner are that it is fast, operates in closed loop and does not disturb the process more than necessary. Another main advantage is that it does not require any knowledge about the process a priori, since the relay feedback experiment automatically excites the process in the frequency range interesting for PID control. The short experiment time is essential, not only due to the overall time-consumption, but also since it reduces the risk of disturbances entering during the experiment.

Since the relay autotuner was introduced in the 1980's [Åström and Hägglund, 1984], many modifications of it have been proposed. Finding a low-order model from the relay experiment was proposed in [Luyben, 1987], where the static gain was assumed to be known and in [Li et al., 1991], where an additional relay experiment was performed. In this paper an asymmetric relay function is used. The asymmetric relay provides a better excitation of the process at lower frequencies than its symmetric counterpart, without making the experiment any more complicated or time-consuming. The original autotuner only gave two parameters, but with an asymmetric relay the low-order models found in e.g. [Luyben, 1987] and [Li et al., 1991] could be found from a single relay experiment without any prior process knowledge. A version of the asymmetric relay function was used in [Shen et al., 1996b], and later investigated in e.g. [Kaya and Atherton, 2001b], [Lin et al., 2004] and [Berner et al., 2014]. For a more thorough review of the advances in modeling from relay feedback experiments, see [Liu et al., 2013].

Although much research has been done on both symmetric and asymmetric relay autotuners, our experience is that focus is seldom on the practical use of the autotuner. Instead exactness or closeness to the true model under ideal simulation conditions is often considered. In this paper we aim for a more practical approach. We use simple low-order models that will of course never describe the true processes exactly. However, the aim is not to get perfect models, but rather to get a good-enough model for tuning a well-performing controller. The focus of this paper is more on practical aspects such as how to choose the experiment parameters and how the models are affected by noise or other disturbances entering the experiment. These investigations are mainly made in a simulation environment and described in Section 4. The autotuner is also tested in a real industrial setting, which is documented in Section 5. The industrial tests were performed on two subsystems of an air handling unit. The experiments were exposed to many of the problems that may be encountered in practice, but still gave overall good results with well-performing controller tunings.

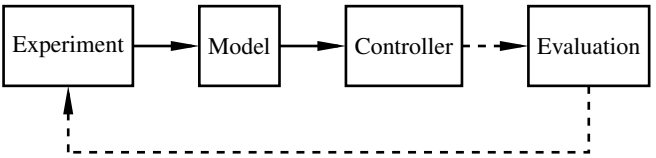


Figure 1. Steps to be designed and performed in an automatic tuning procedure. The dashed lines show the steps that involve the user.

2. Automatic Tuning

The purpose of the autotuner is to give satisfactory controller parameters for a process with completely unknown dynamics. To do this, the autotuner goes through the different steps shown in Figure 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 is also decided what experiment parameters should be used, and what data should be extracted from the experiment. In this paper the experiment is the asymmetric relay feedback experiment, described further in Section 3.

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, defined as

$$\tau = \frac{L}{L+T}, \qquad 0 \leq \tau \leq 1, \tag{1}$$

where L is the apparent time delay and T is the apparent time constant of the process. The model structure choice is made according to the decision scheme in Figure 2, proposed in [Berner, 2015; Berner et al., 2016]. This paper focuses on the simple version where either a first order model with time delay, FOTD model

$$P(s) = \frac{K_p}{1+sT} e^{-Ls}, \tag{2}$$

or an integrating model with time delay, ITD model

$$P(s) = \frac{k_v}{s} e^{-Ls}, \tag{3}$$

is calculated from the experiment.

When the model is found a *Controller* should be designed. This step includes decisions about what controller type to use and how to choose its parameter values. In this paper we use the PID controller on the form

$$C(s) = K\left(1 + \frac{1}{sT_i} + sT_d\right) \tag{4}$$

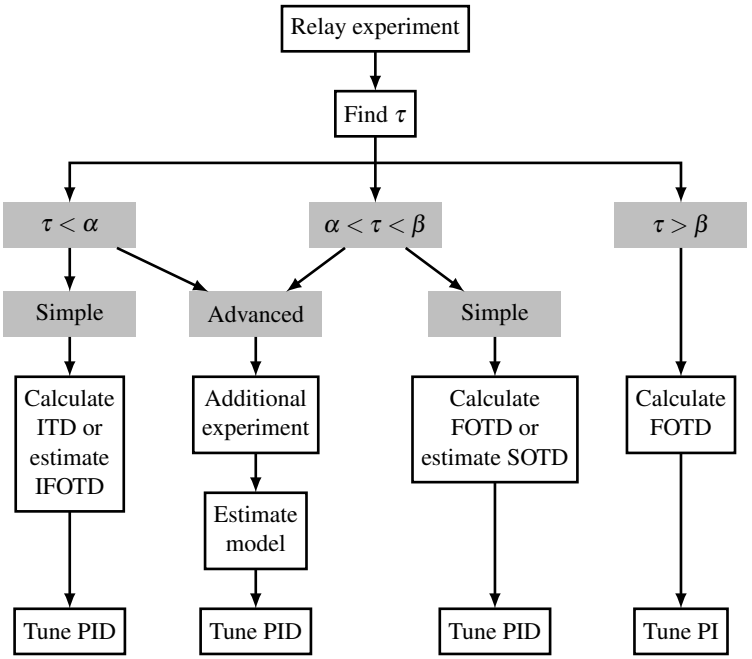


Figure 2. Decision scheme based on the normalized time delay τ. The limits for low and high values of τ are set to $\alpha = 0.1$, $\beta = 0.6$ as in [Berner, 2015].

tuned from the AMIGO rules [Åström and Hägglund, 2006] for the obtained models. Changing to another tuning method is straightforward. The choice of whether or not to use the D-part of the controller can be based on the scheme in Figure 2.

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 in the previous steps. This is mainly a task for the user. One possibility is to use the transient after the experiment as validation data, to see if it agrees with the expected behavior of the obtained process model. Separate evaluation experiments of the controller performance, measuring for instance the integrated absolute error for an added load disturbance, could also be performed.

3. Asymmetric Relay Feedback

The experiment sequence is shown in Figure 3. This section describes the two last steps, the relay feedback and the retrieval of data, while the practical issues of the first steps are described in Section 4.

Let u denote the output signal from the relay function, and y denote the process output signal. The asymmetric relay experiment is started when the system is at

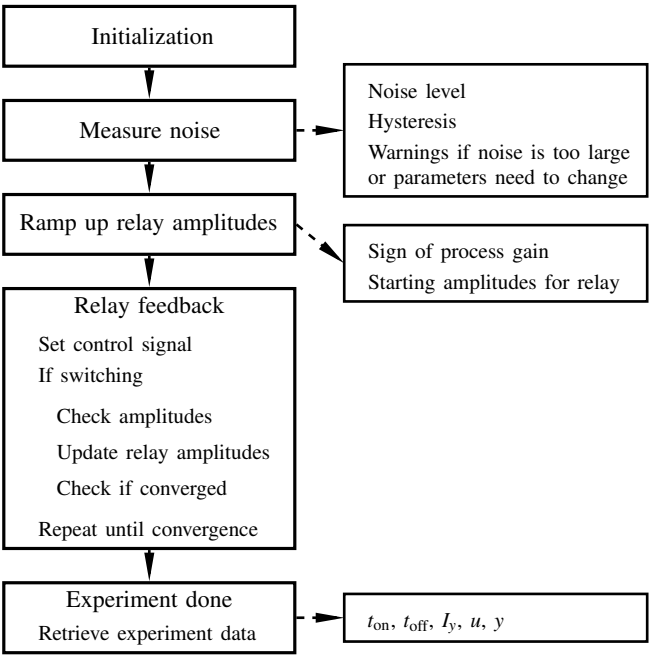


Figure 3. The different sequences of the relay feedback experiment. Also shown are the variables and parameters obtained in that sequence.

equilibrium at the point (u_0, y_0). The asymmetric relay function proposed is

$$u(t) = \begin{cases} u_{on}, & y(t) < y_0 - h, \\ u_{on}, & y(t) < y_0 + h, \quad u(t^-) = u_{on}, \\ u_{off}, & y(t) > y_0 - h, \quad u(t^-) = u_{off}, \\ u_{off}, & y(t) > y_0 + h, \end{cases} \tag{5}$$

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

$$u_{on} = u_0 + d_1 \operatorname{sign}(K_p), \qquad u_{off} = u_0 - d_2 \operatorname{sign}(K_p). \tag{6}$$

The sign of the process gain K_p (or k_v if the process is integrating) is determined during the startup of the experiment, as will be described in Section 4.2.

The name *asymmetric* relay reflects that the amplitudes d_1 and d_2 are not equal. This creates the asymmetric oscillations. The asymmetry level of the relay is denoted γ and defined as

$$\gamma = \frac{\max(d_1, d_2)}{\min(d_1, d_2)} > 1. \tag{7}$$

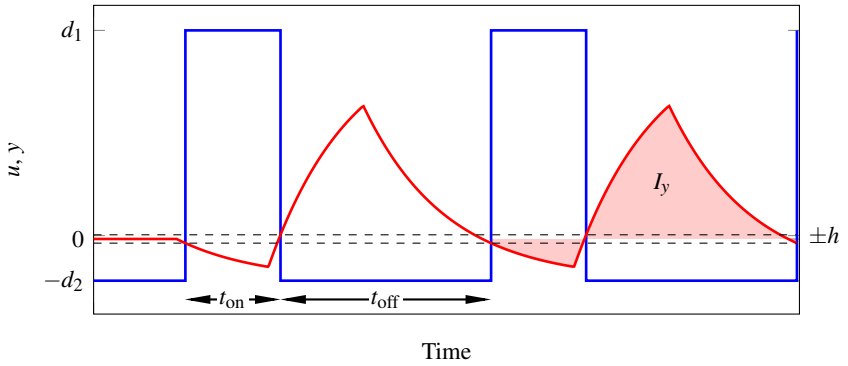


Figure 4. An example of the signals from the asymmetric relay feedback experiment. The relay output u is shown in blue, the process output y is shown in red. The black dashed lines show the hysteresis levels, $\pm h$. The experiment starts with the system in equilibrium at the point (u_0, y_0), which in the figure is only denoted with a zero. 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 is shown in Figure 4. The half-periods t_{on} and t_{off} are defined as the time intervals where $u(t) = u_{on}$ and $u(t) = u_{off}$, respectively.

3.1 Modeling

Once the experiment has converged we want to find the parameter values for the desired model structure. We have focused on finding simple equations that are robust to noisy data. To find the FOTD and ITD models we use equations where the only measurements needed are the durations t_{on} and t_{off}, and the integral of the process output, I_y, shown in Figure 4 and defined as

$$I_y = \int_{t_p} \big(y(t) - y_0\big)\, dt. \tag{8}$$

Here $t_p = t_{on} + t_{off}$ is the period time of the oscillation and y_0 is the stationary operation point we started the experiment at. Both t_{on}, t_{off} and I_y 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 and the integral of the relay output I_u defined in (11).

The normalized time delay can be found from the relay experiment by

$$\tau(\rho, \gamma) = \frac{\gamma - \rho}{(\gamma - 1)(0.35\rho + 0.65)}, \tag{9}$$

as described in [Berner, 2015; Berner et al., 2016], where ρ is the half-period ratio defined as

$$\rho = \frac{\max(t_{\text{on}}, t_{\text{off}})}{\min(t_{\text{on}}, t_{\text{off}})}. \tag{10}$$

The integral of the relay output

$$I_u = \int_{t_p} \big(u(t) - u_0\big)\, dt \tag{11}$$

does not need to be measured from the experiment since it is given by

$$I_u = (u_{\text{on}} - u_0)t_{\text{on}} + (u_{\text{off}} - u_0)t_{\text{off}}. \tag{12}$$

FOTD Models The FOTD model (2) has three parameters: K_p, T and L. One benefit of using the asymmetric relay, is that the static gain, K_p, can be determined and is given by

$$K_p = \frac{I_y}{I_u}. \tag{13}$$

Note that this does not apply to the symmetric relay, where I_u is always zero. It follows from (12) that I_u can become zero with the asymmetric relay as well, but only if $t_{\text{off}}/t_{\text{on}} = d_1/d_2$. This means that $\rho = \gamma$, which implies that $\tau = 0$, and for those processes we will use the ITD model, as indicated in Figure 2.

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

$$t_{\text{on}} = T \ln \frac{h/|K_p| - d_2 + e^{L/T}(d_1 + d_2)}{d_1 - h/|K_p|} \tag{14}$$

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

given in [Berner, 2015]. Since K_p can be found from (13), the results in (14) and (15) 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 use the normalized time delay τ given by (9), and find the ratio between L and T from (1) as

$$L/T = \frac{\tau}{1 - \tau}. \tag{16}$$

Knowing this ratio, T can be found from either of the two equations (14) or (15), or from an average of both. With T known, it is straightforward to get L from (16).

ITD Models The integrating process (3) can be described by the differential equation

$$\dot{y}(t) = k_v u(t - L). \tag{17}$$

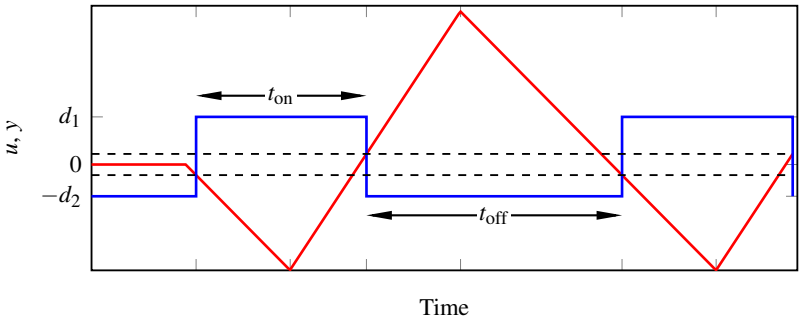


Figure 5. An example of the signals from a relay experiment with an ITD process. The blue line shows the relay output u, the red line shows the process output y. The dashed black lines show the hysteresis. Note the triangular shape of y that is characteristic for an ITD process.

Since $u(t)$ is piecewise constant, so is $\dot{y}(t)$, and hence the shape of y will be triangular, see Figure 5. By considering the output curves, equations for k_v and L can be obtained, see [Berner, 2015] for full derivation. The equations are

$$k_v = \frac{2I_y}{t_{on}t_{off}(u_{on} + u_{off})} + \frac{2h}{u_{on}t_{on}}, \quad L = \frac{u_{on}t_{on} - 2h/k_v}{u_{on} - u_{off}}. \tag{18}$$

Higher Order Models If the normalized time delay τ is not large, the control requirements are high, and the implementation and computing power permit it, the parameters of a higher order model can be estimated. To obtain the second order time delayed (SOTD) or integrating plus first order time delayed (IFOTD) models, shown in the decision scheme in Figure 2, we use the entire experiment data set. The model parameters are estimated from a system identification method based on Newton's method, as in [Berner et al., 2014]. To assure convergence in the iterative method, appropriate initial parameter values are needed. These are obtained from the calculated FOTD or ITD model, see [Berner, 2015] for details. Parameter estimation methods can be used to find other model structures as well, or to improve the accuracy of the obtained models. However, that is not the focus of this paper where we stick to the simple FOTD and ITD models obtained as described in the previous subsections.

4. Practical Considerations

When the autotuner is used in an industrial setting, the conditions may be very different from the ideal simulation environment where the development has been done. In this section some of the practical issues for the autotuner will be presented

and discussed. The first part goes through the choice of relay parameters. The rest of the section discusses how the results from the autotuner are affected by practical issues like noise, load disturbances and low resolution in converters. The evaluation in this section has been performed on a test batch from [Åström and Hägglund, 2006] containing 134 processes representative for many systems in the process industry. All the processes of the test batch are listed in Appendix B.

4.1 Parameter Choices

The relay experiment contains several parameters that have to be chosen. This is done in the implementation of the autotuner and nothing that needs to be done in every new setup. Some of the parameters could, however, be changed by the user if desired. Default values for all parameters are listed in Appendix A, and some of the parameter choices are explained and discussed in further detail in the current section.

Noise level and hysteresis As shown in Figure 3, the first step of the autotuning experiment, after the initialization, is to 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 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 a noise-free simulation environment the hysteresis could be chosen arbitrarily.

Relay amplitudes The question of how to choose the relay amplitudes is subject to some different aspects. For minimal disturbance of the process, the amplitudes should be as small as possible. However, 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

$$\min(d_1, d_2) \geq \frac{\mu h}{|K_p|} \tag{19}$$

where $\mu > 1$ is a constant, is required to get good results. In Figure 6 the accuracy of the estimated τ for the processes in the test batch 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 (19) 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 introducing a lower limit $y_{\text{mindev}} = \mu h$ on the peak deviation we can guarantee that (19) is satisfied since

$$|K_p| \min(d_1, d_2) \geq y_{spd} \geq y_{\text{mindev}} = \mu h. \tag{20}$$

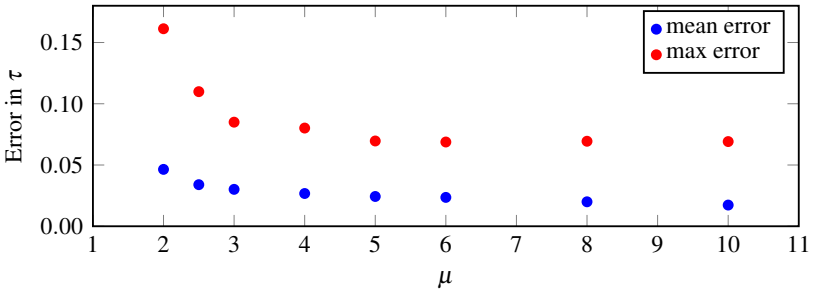


Figure 6. Mean and maximum errors of the τ-estimations, shown for some different values of μ. During these simulations, the relay amplitudes were fix, with the small amplitude $\min(d_1, d_2) = \mu h/|K_p|$ and the large amplitude a factor $\gamma = 2$ larger.

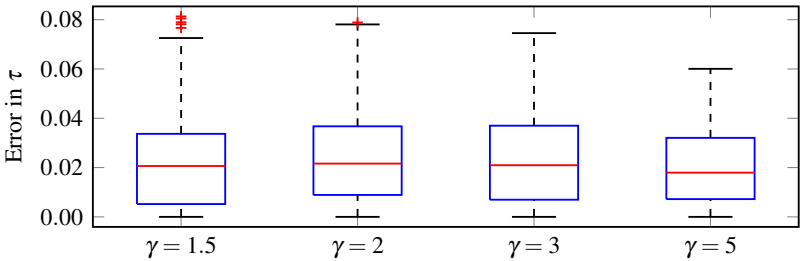


Figure 7. Boxplots of the absolute errors of τ, shown for different values of the asymmetry level γ. On each box, the central mark is the median, the edges of the box are the 25th and 75th percentiles, and the whiskers extend to the most extreme data points. Outliers are plotted individually.

How the lower limit is accomplished in practice is described by the amplitude adjustment in Section 4.2. Note that since there are multiple inequalities in (20) such a large value of μ as in (19) might not be necessary to get good results. With the default parameters listed in Appendix A, the maximum error of τ is 0.08 for the test batch, using the small value $y_{\text{mindev}} = 2h$.

The asymmetry level γ, i.e., the ratio between d_1 and d_2 is also something to consider. Figure 7 and Figure 8 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 are plotted. For the estimates of K_p only the processes that were estimated as FOTD processes, and hence have a finite K_p value, are shown. Note that all the non-integrating processes in the test batch have $K_p = 1$, see Appendix B. 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 do not depend that much on γ.

By forcing y_{spd} to be large, and using a high γ, the results are more accurate. If

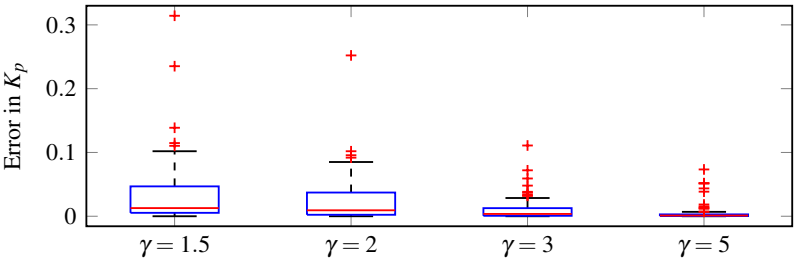


Figure 8. Boxplots of the absolute errors of K_p, shown for different values of the asymmetry level γ. All processes, classified as non-integrating, from the test batch are included.

the autotuner is to be used in simulation environments you could therefore use high values. However, if the autotuner is used on real processes in an industrial setup there will be upper constraints as well. There will be 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 would prefer to. How the upper limits enter into the process of choosing the relay amplitudes is explained in Section 4.2.

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 with noise, 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 takes, t_p, with the time the previous period took, t_p^*. If the relative difference between the period times is smaller than a certain threshold ϵ, i.e.,

$$\left| \frac{t_p - t_p^*}{t_p^*} \right| \leq \epsilon \tag{21}$$

the system is considered to have reached the limit cycle. Alternatively the oscillation amplitudes could be considered instead of the period times, but that approach was not chosen in this paper.

To investigate the effect of ϵ, the processes in the test batch were simulated with the different values $\epsilon = \{0.005, 0.01, 0.05\}$. 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 9 and Figure 10.

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. The mean convergence time for $\epsilon = 0.05$

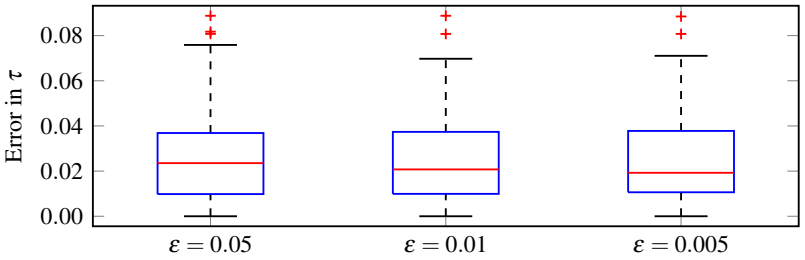


Figure 9. Boxplots of the absolute errors in the estimation of τ, for the three different convergence limits $\epsilon = \{0.05, 0.01, 0.005\}$.

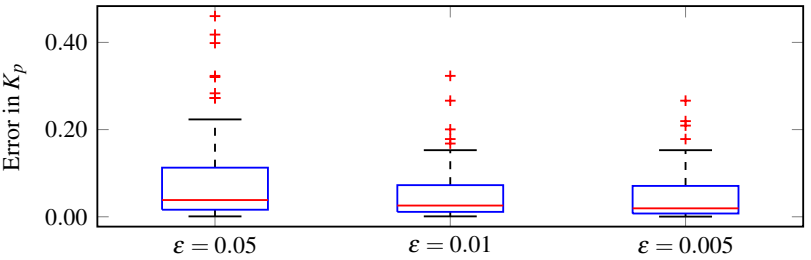


Figure 10. Boxplots of the absolute errors in the estimation of K_p, shown for the three different convergence limits $\epsilon = \{0.05, 0.01, 0.005\}$. All processes, classified as non-integrating, from the test batch are included.

was 0.31 periods shorter than for $\epsilon = 0.01$, while the convergence time for $\epsilon = 0.005$ was in mean 0.52 periods longer than for $\epsilon = 0.01$. At most, $\epsilon = 0.05$ gave a 2.5 periods shorter convergence time, while $\epsilon = 0.005$ made one process take 9 periods more to converge than it did with $\epsilon = 0.01$. Since the accuracy was more or less the same for $\epsilon = 0.01$ and $\epsilon = 0.005$ there is no need to use the lower value, since that increases the experiment time. Increasing the limit to $\epsilon = 0.05$ makes the experiment a little shorter, but the obtained values of K_p are also somewhat worse. Considering the results, the default value $\epsilon = 0.01$ was chosen.

Sampling times The question of how to choose the sampling time for the relay experiment is not trivial, since no information about the process is available when designing the experiment. Note that the sampling time during the experiment is not necessarily the same as the sampling time for the controller. The experiment typically requires a shorter sampling time to get good accuracy. The discussion of sampling times in this section is for the experiment. However, the autotuner could after the experiment also give a suggestion of a suitable sampling time for the controller. A discussion on how to choose sampling time for the controller, based on process information, is given in e.g. [Isermann, 1989, Ch. 5].

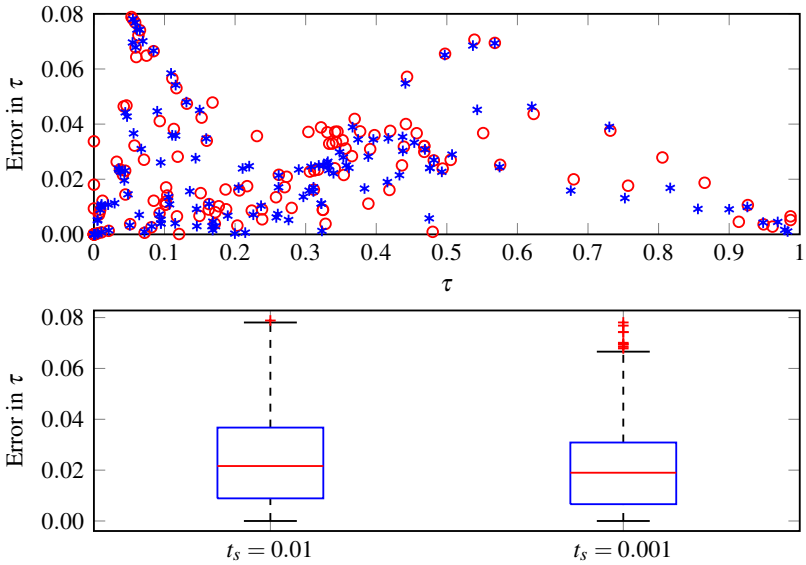


Figure 11. Estimates of τ for two different sampling times. The upper plot shows the absolute error of the estimated τ, as a function of the estimated τ. The red circles show the results for $t_s = 0.01$ s, and the blue stars show the results for $t_s = 0.001$ s. The lower plot shows the corresponding boxplots of the absolute errors.

In the simulation experiments in this paper a fixed sample time of $t_s = 0.01$ s was used as default value. The test batch 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 11 and Figure 12.

The results for τ in Figure 11 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 12 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 I_u, given in (12), 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 1 s. They behave 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 these 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

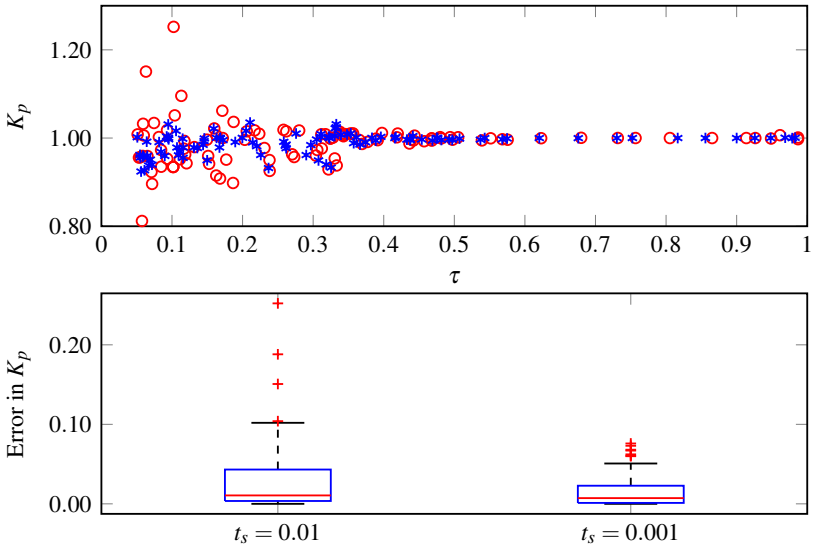


Figure 12. Estimates of the process gain K_p for two different sampling times. The upper plot shows the estimates as a function of the estimated τ. The red circles show the results for $t_s = 0.01$ s and the blue stars show the results for $t_s = 0.001$ s. The lower plot shows boxplots of the absolute error of K_p. In both the upper and lower plot the only processes shown are the ones with an estimated $\tau > 0.05$.

very slow processes 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 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 poorly chosen sampling time is to adjust the sampling time after the first half-period when the approximate speed of the process is known. Another way is to always sample as fast as the hardware permits during the experiment, and then double the sample time and start overwriting samples if the buffer gets full.

4.2 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 paper is inspired by earlier versions of industrial autotuners [ABB, 2010]. The control signal is increased exponentially until one of two things happens. 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 sign of the process gain is determined based on which limit that is broken, and the initial relay

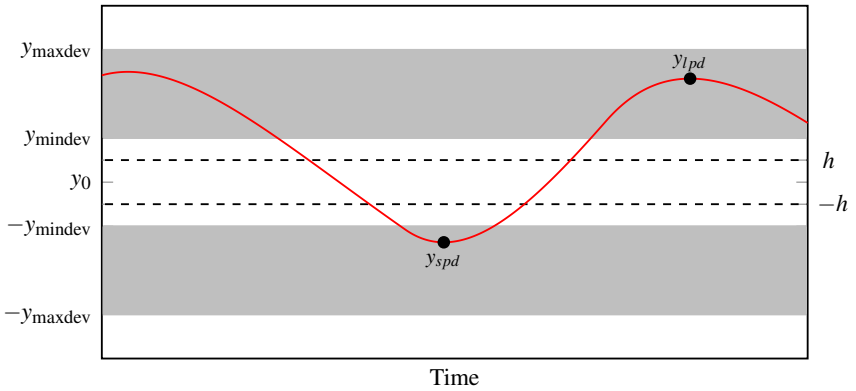


Figure 13. Oscillation restrictions for the process output y. The peak values y_{lpd} and y_{spd} should both stay within the gray-marked areas. To get good results it is necessary that y_{mindev} is a bit larger than the hysteresis level h. The limit y_{maxdev} is due to the fact that the process should not be disturbed too much, this limit may be set by the operator.

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 absolute value, 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 4.1. 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 13. The amplitude adjustment is described in Algorithm 1, but there is some additional logic to make sure that the relay amplitudes never exceed u_{maxdev}.

To exemplify, consider a situation when the lower limit is set to $y_{\text{mindev}} = 2h$, the upper limit is set to $y_{\text{maxdev}} = 6h$ 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

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

The obtained update factor adjusts the measured amplitudes towards the desired values. An example of the startup and amplitude adjustments is illustrated in Figure 14.

if *switching AND not changed in last switch* **then**
  if $\max(y_{dev}) > y_{maxdev}$ **then**
    $y^*_{lpd} = (y_{\text{maxdev}} + y_{\text{mindev}}\gamma)/2$
    $\varphi_u = y^*_{lpd}/\max(y_{\text{dev}})$
  else if $\max(y_{dev}) < y_{mindev}$ **then**
    $y^*_{spd} = (y_{\text{mindev}} + y_{\text{maxdev}}/\gamma)/2$
    $\varphi_u = y^*_{spd}/\max(y_{\text{dev}})$
  else
    $\varphi_u = 1$
  end
  $d_1 = d_1\varphi_u$
  $d_2 = d_2\varphi_u$
end

Algorithm 1: Amplitude adjustments. Here y_{dev} stands for the process value's deviation from the setpoint, y^*_{spd} and y^*_{lpd} stand for the wanted small peak deviation and large peak deviation respectively. The amplitude update factor is denoted φ_u.

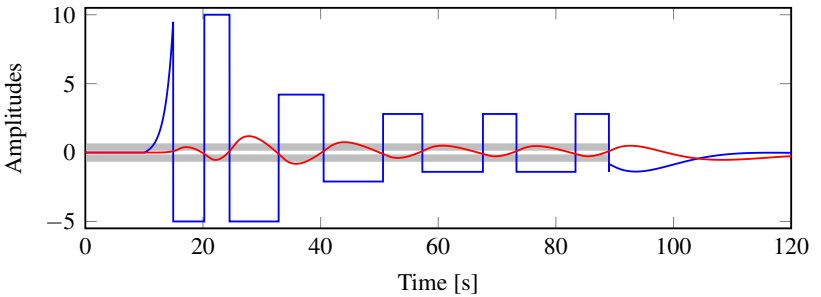


Figure 14. An example of the startup and amplitude adjustments. Here $u_{\text{maxdev}} = 10$, $\gamma = 2$, $y_{\text{mindev}} = 2h$, $y_{\text{maxdev}} = 6h$. The gray areas show the allowed areas for the peak values y_{spd} and y_{lpd}.

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

$$\gamma \leq \frac{y_{\text{maxdev}}}{y_{\text{mindev}}}, \tag{22}$$

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

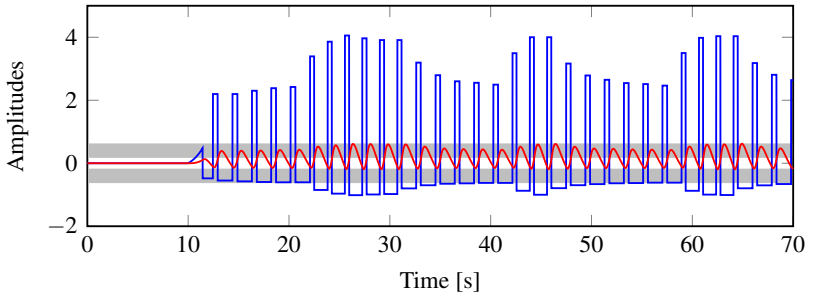


Figure 15. An example of the problems obtained if γ is larger than $y_{\text{maxdev}}/y_{\text{mindev}}$. The gray areas show the allowed areas for the peak values y_{spd} and y_{lpd}. The algorithm will never find relay amplitudes that satisfy both limits and will keep changing the amplitude.

in Figure 15. 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.

4.3 End of Experiment

When the experiment has converged, as described in Section 4.1, the controller is reconnected. In this implementation the relay is disconnected as soon as the new controller parameters are calculated. This happens right after the last switch of the relay, and the new controller parameters are used immediately. The last switch of the relay is seen clearly in some of the experiment figures as a spike before the controller starts. Reconnecting the controller immediately after the last relay switch may not be the best option. In [Theorin and Berner, 2015] another implementation choice was made called the *graceful shutdown* that gives a smoother transition.

Another practical issue concerning the ending of the experiment is that the convergence check cannot be performed at the first relay switches since there is no previous time period to compare with until three half-periods have passed. Neither can the convergence check be performed when the relay amplitudes are adjusted, since the amplitude change influences the period times. The convergence check is restarted three half-periods after the last amplitude adjustment in order to compare the time period t_p to the appropriate t_p^*.

4.4 Measurement Noise

To demonstrate the experiment's robustness to measurement noise, the test batch was simulated with some different noise levels. To introduce noise to the simulations, band-limited white noise was added to the process output. As was mentioned in Section 4.1 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

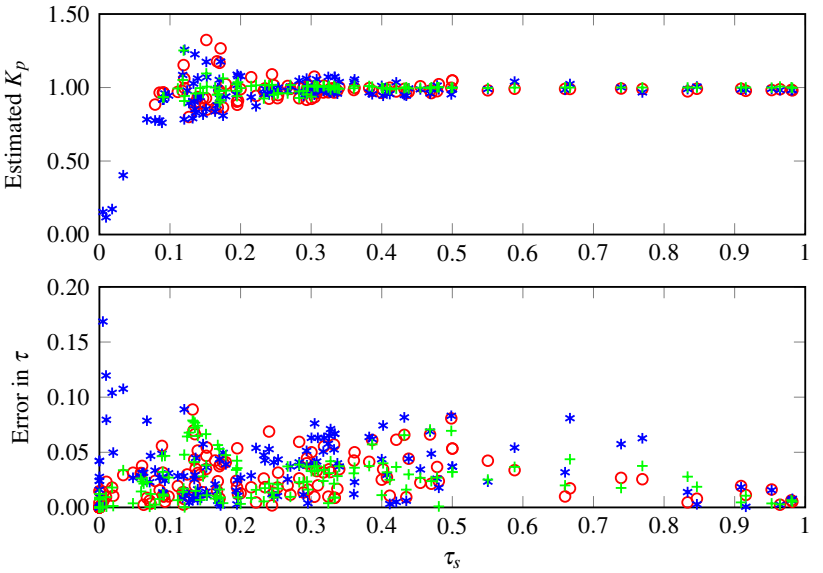


Figure 16. Results for the different noise levels $n_0 = 0$ in green plus signs, $n_0 = 0.12$ in red circles, and $n_0 = 0.38$ in blue stars. The upper plot shows the estimated values of K_p and the lower plot shows the error in the estimated τ. In this figure they are plotted as a function of the true normalized time delay τ_s instead of the estimated τ. In the plot for K_p, only the processes classified as non-integrating are included. Note that this means that there should ideally only be points for $\tau_s > 0.1$. However, in the noisiest experiments some processes were wrongly classified as non-integrating which gives the problematic blue stars with $\tau_s < 0.1$ in the upper plot.

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 16, and the corresponding boxplots are shown in Figure 17 and Figure 18, respectively.

Figure 16 shows that the processes with small τ give the worst estimates. This emphasizes that it could be worthwhile to put some extra effort in modeling these processes. The figures for K_p only contain the processes classified as non-integrating, i.e., with an estimated $\tau > 0.1$, but Figure 16 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 deviate 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.

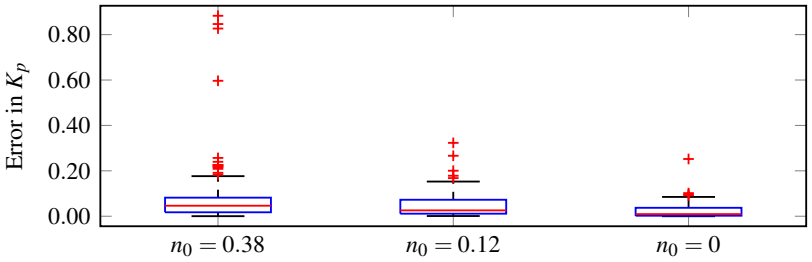


Figure 17. Boxplots of the absolute errors of K_p, shown for different noise levels. All processes, classified as non-integrating, from the test batch are included.

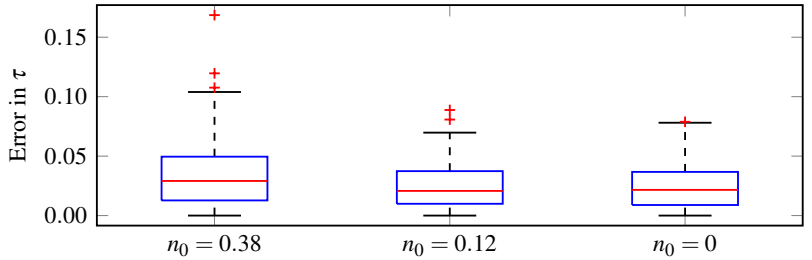


Figure 18. Boxplots of the absolute errors of τ, shown for different noise levels.

To further illustrate the effect of a noisy experiment, we take a closer look at Figure 19 where one of the processes from the test batch, namely

$$P(s) = \frac{1}{(s+1)^5}, \tag{23}$$

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. This is the reason why the noisy experiment is faster, since it doesn't have to adjust the relay amplitudes due to the larger deviations allowed. In reality the upper limit on the deviation may not be allowed to be so large. This implies that it may be necessary to filter the signal before performing the relay experiment if the noise level is too large.

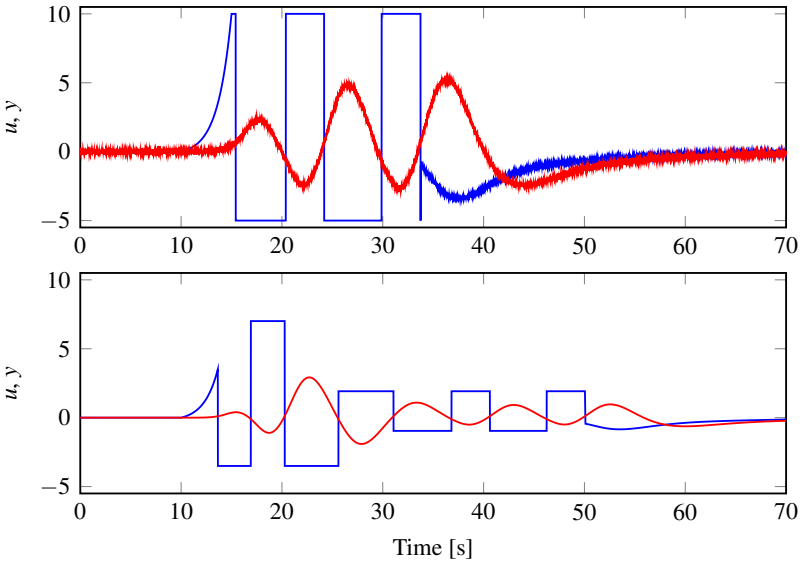


Figure 19. Relay experiment for the process $P(s) = 1/(s+1)^5$. The upper plot shows the signals when noise is added to the process output, while the lower plot shows the corresponding noise-free simulation. The blue line shows the relay output, the red line shows the process output.

4.5 Effects of Quantization

The resolution of A/D and D/A converters may influence 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 unknown 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 20, where the results are compared to a continuous D/A converter.

The results are worse for the less resolved D/A converter. The reason to the difference can be understood from the example shown in Figure 21. This example shows the worst case achieved in the test batch simulation. The process in the example,

$$P(s) = \frac{e^{-0.7s}}{s(0.3s+1)}, \tag{24}$$

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 look 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

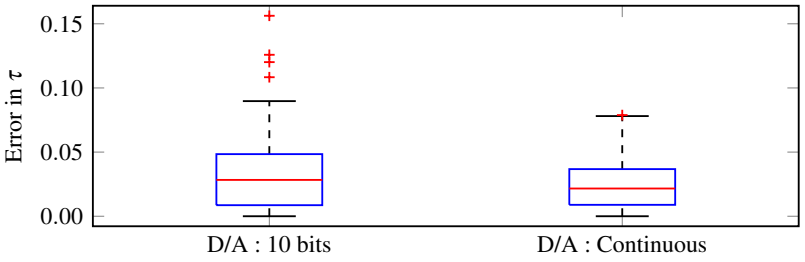


Figure 20. Boxplots of the absolute errors of τ, shown for two different resolutions of the D/A converter.

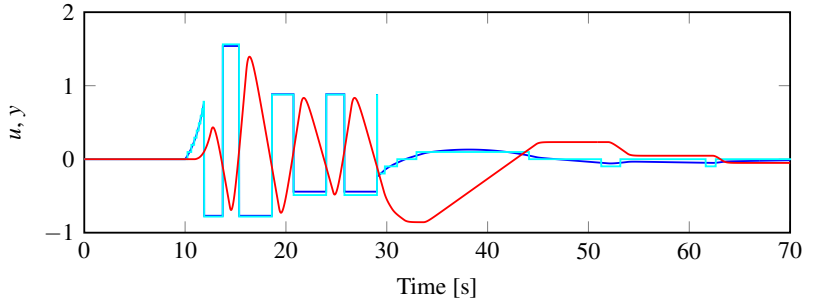


Figure 21. The effect of quantization in the D/A converter. The calculated control signal u is shown in blue, while the output from the 10 bit D/A converter, u^q, is shown in turqoise. The process output y is shown in red. The final relay amplitudes are $d_1 = 0.88$, $d_1^q = 0.88$, $d_2 = 0.44$, $d_2^q = 0.49$.

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. Figure 22 shows 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.

4.6 Load Disturbances

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 obtained. This will be obvious to the user and the experiment can be

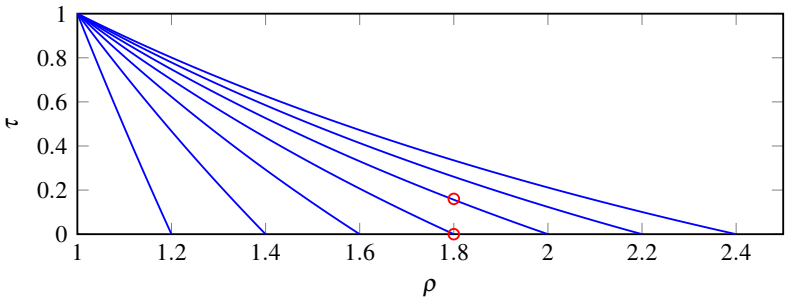


Figure 22. The τ curves corresponding to (9), plotted for the different asymmetry levels $\gamma = \{1.2, 1.4, 1.6, 1.8, 2, 2.2, 2.4\}$. The points marked with red circles show the difference between the value we should have obtained if we knew that $\gamma = 1.8$ and the value we get when we think that $\gamma = 2$.

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 4.5. 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. However, 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.

4.7 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 [Åström and Hägglund, 2006] for the obtained FOTD model. The results of the experiments are shown in two different figures. In Figure 23 the relay experiment starts immediately when the error is small enough. In Figure 24 the experiment starts with measurement of the noise level, as described in Section 4.1.

As can be seen from Figure 23 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 24 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 23 and Figure 24 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.

5. Industrial Experiments

After the investigations of practical issues performed in simulations, it was time to try the usefulness on an industrial system. The autotuner was implemented and tested on an air handling unit provided by Schneider Electric Buildings AB in Malmö, Sweden. 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.

5.1 Integration of the Autotuner in an Industrial System

To test the autotuner on the air handling unit, it was first implemented in Schneider Electric's software StruxureWare Building Operation. The autotuner implementation was made as a script program in the Building Operation server. In 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

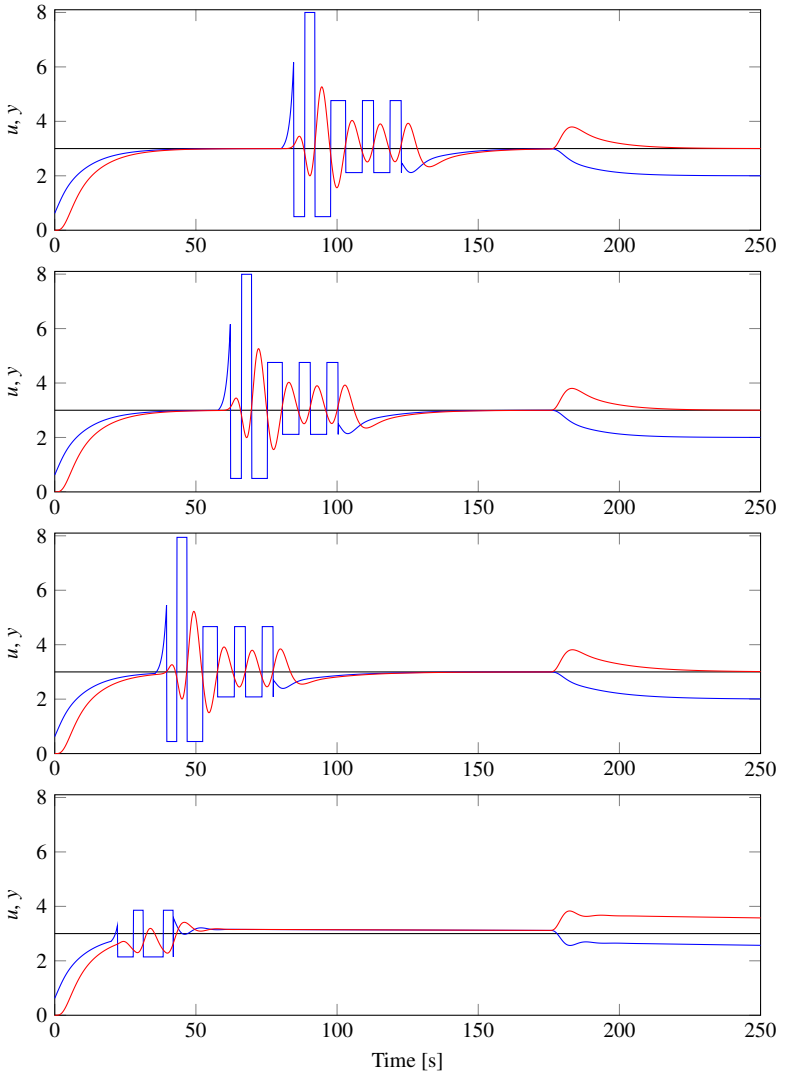


Figure 23. Results when the relay experiment is started at a non-steady state. The error at the experiment start is $y_e = \{0.001, 0.01, 0.1, 0.5\}$, listed from the top figure to the bottom one. The resulting PI controller parameters achieved from the experiment are $K = \{0.26, 0.26, 0.24, 0.75\}$ and $T_i = \{3.38, 3.38, 4.05, 171\}$, listed from top to bottom. At time 175 s a step load disturbance is added and the control performance can be seen.

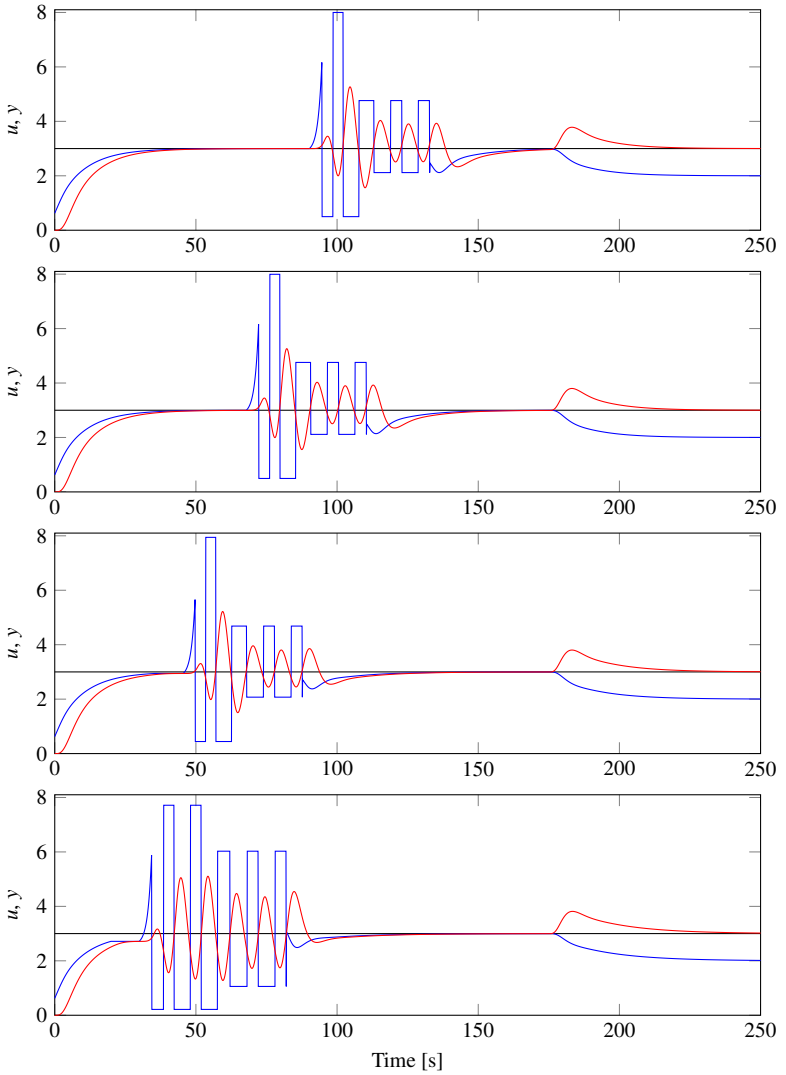


Figure 24. Results when the relay experiment is started at a non-steady state. In this figure the experiment starts with measuring the noise level for 10 s before the relay starts. The error at the experiment start is $y_e = \{0.001, 0.01, 0.1, 0.5\}$, listed from the top figure to the bottom one. The 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.

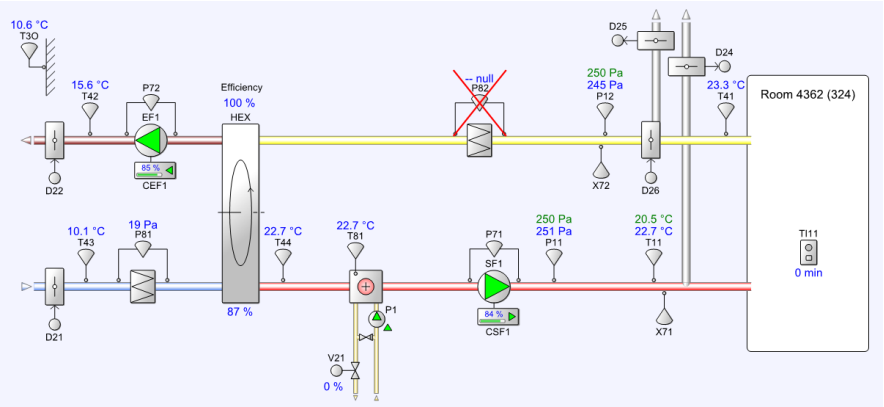


Figure 25. Schematics of the air handling unit.

the possibility of code development and testing by simulations. The autotuner implementation includes all the sequences of the relay feedback experiment shown in Figure 3. 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 in the system. 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.

5.2 System Description

The schematics of the air handling unit is shown in Figure 25, and pictures of the system are shown in Figure 26 and Figure 27. Outside air enters the system through the duct with the blue arrow sign, to the left in Figure 26. 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 27. The heated air is then led through the duct with the red arrow sign to the right in Figure 26, 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 26 from the right. The temperature and the pressure

Figure 26. The air handling unit.

Table 1. Parameters from the pressure control experiment.

	τ [-]	K_p [Pa]	T [s]	L [s]	K [1/Pa]	T_i [s]
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

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 26.

5.3 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 term pressure in this paper refers to the gauge pressure, not the absolute pressure. The control signal is normalized to a percentage of the full speed of SF1, while the pressure is measured in Pascal. The reference value of the pressure was 250 Pa. Two relay experiments were performed on the system. The calculated model and controller parameters for the two experiments are listed in Table 1. The data from the second experiment is also shown in Figure 28. The experiment started with 40 s measurement of the noise. The figure shows that the signal is quite noisy, in this experiment the noise level was measured to $n_0 = 6.51$ Pa. With this noise level,

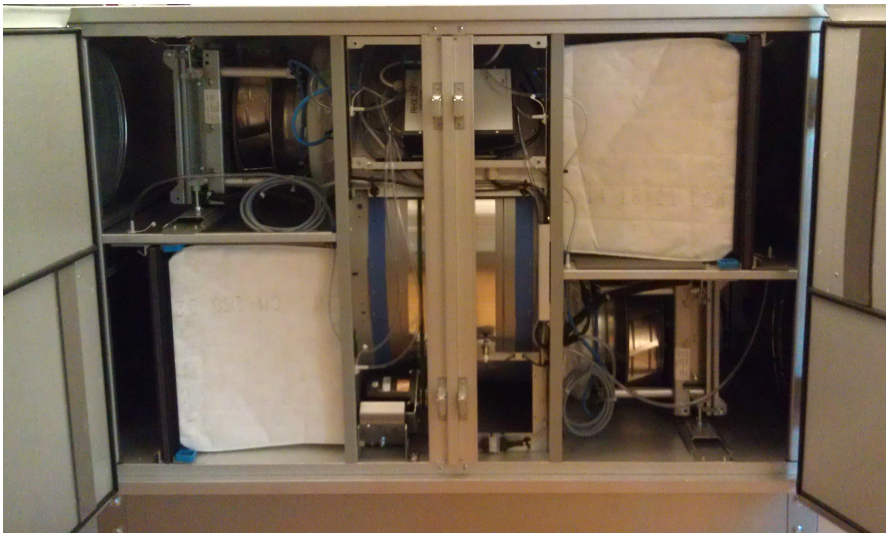


Figure 27. The box contains the heat exchanger, filters and fans. The supply air passes in the lower part, while the exhaust air passes in the upper part. The rotational heat exchanger in the middle transfers heat from the exhaust air to the supply air.

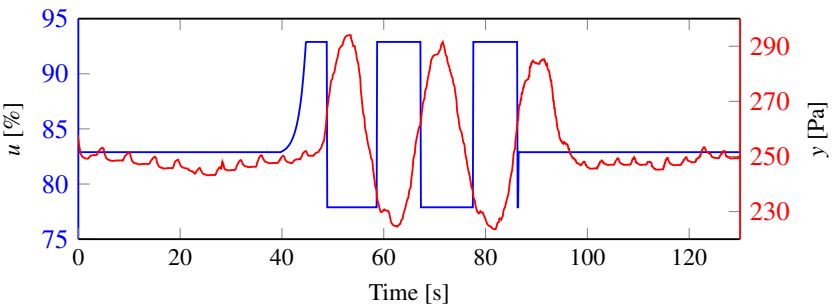


Figure 28. Experiment data from the pressure control loop. The normalized fan speed is shown in blue, the pressure measurement in red. Note the different scales and units on the axes.

the hysteresis value was set to $h = 2n_0 = 13.02$ Pa. The sampling 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 Appendix A.

Table 1 shows that the results of the different experiments are quite similar, with the largest difference in the estimates of K_p. Since τ was large, a PI controller was

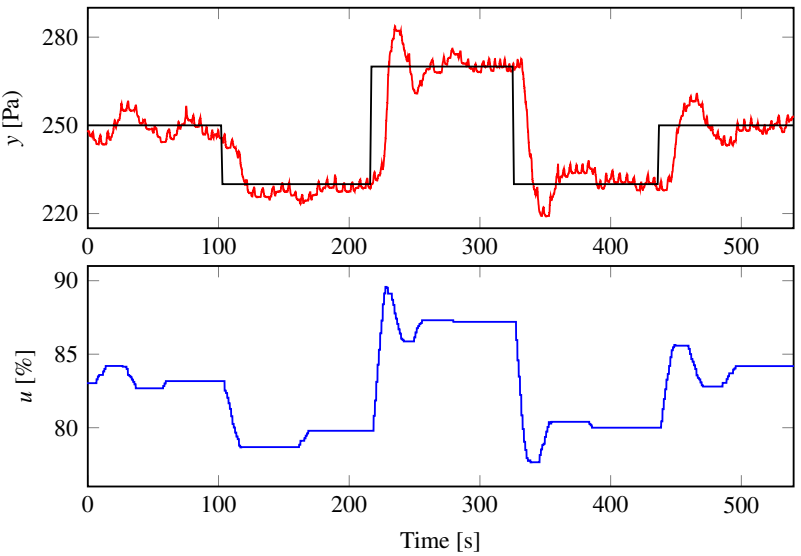


Figure 29. Response to setpoint changes for the system with the controller tuned from the experiment. The upper plot shows the measured pressure in red, and the setpoint in black. The lower plot shows the control signal.

tuned, according to the scheme in Figure 2. 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 period of 1 s, and had a dead zone of 5 Pa. Results from step changes in the reference value are shown in Figure 29. The step response results are satisfactory. There is an overshoot, but that can be reduced by filtering the setpoint. The presence of the dead zone is clearly visible from the long periods of constant control signal and process output deviations from the setpoint.

Step load disturbances of unknown sizes were introduced by manually adjusting a damper. The response to these, shown in Figure 30, are also satisfactory. The effect of the load disturbances are removed completely in approximately 20-25 s with small overshoots.

5.4 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

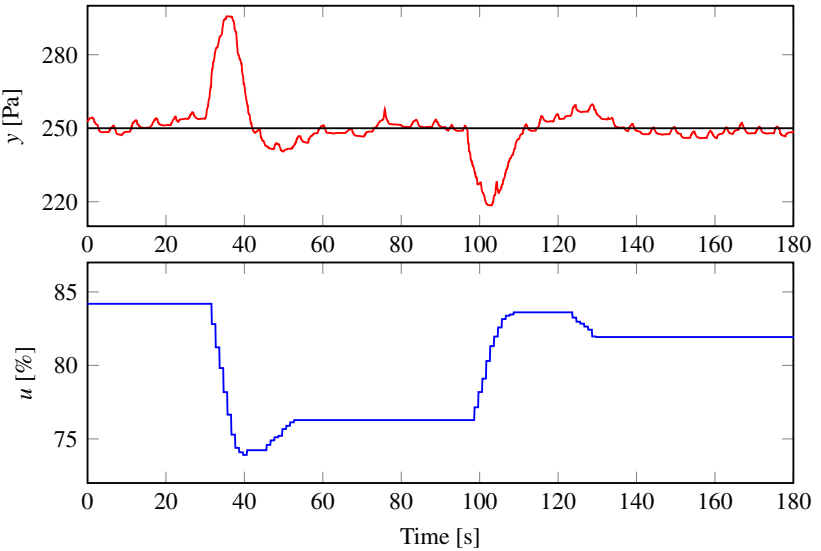


Figure 30. Response to load disturbances. The upper plot shows the measured pressure in red, and the setpoint in black. The lower plot shows the control signal.

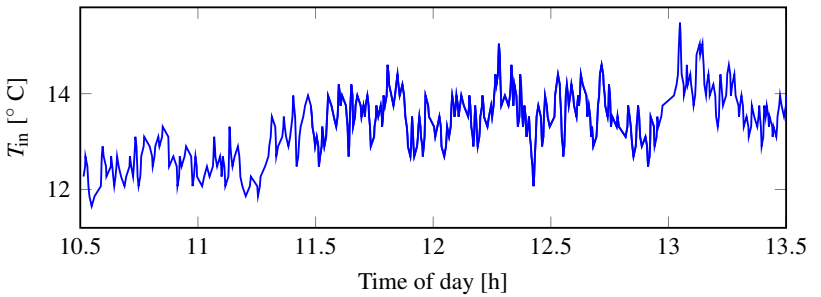


Figure 31. The temperature of the inlet air, T_{in}, measured by sensor T43.

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 31. This temperature deviation is treated as a load disturbance and caused some trouble during the experiments.

One relay experiment is shown in Figure 32. The noise was measured for 40 s and the measured noise level was $n_0 = 0.045°C$. This gave a hysteresis level of $h = 0.09°C$. The same default values as for the pressure control experiment were used, except for a change in the convergence limit from $\epsilon = 0.01$ to $\epsilon = 0.05$. The reason for this change is that the varying load from the inlet air temperature

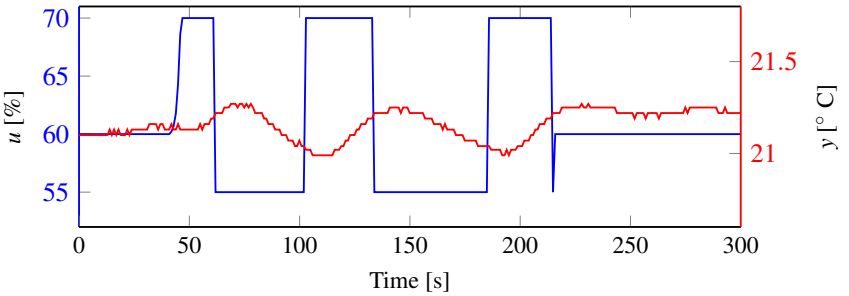


Figure 32. A relay experiment performed on the temperature control loop. The blue line shows the normalized speed of the heat exchanger, and the red line shows the supply air temperature. Note the different scales and units on the axes.

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

$$P(s) = \frac{0.035}{35.8s + 1} e^{-8.8s}. \tag{25}$$

From this model the parameters for a PI controller were calculated to $K = 26.6°\text{C}^{-1}$ and $T_i = 29.2$ s. The PI controller was then connected and some setpoint changes were performed to check the obtained controller performance. During the step tests the 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 33. 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, $u_{\text{maxdev}} = 20$. The data from this experiment is shown in Figure 34. This experiment showed another problem with the process. The behavior of 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 33. The 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

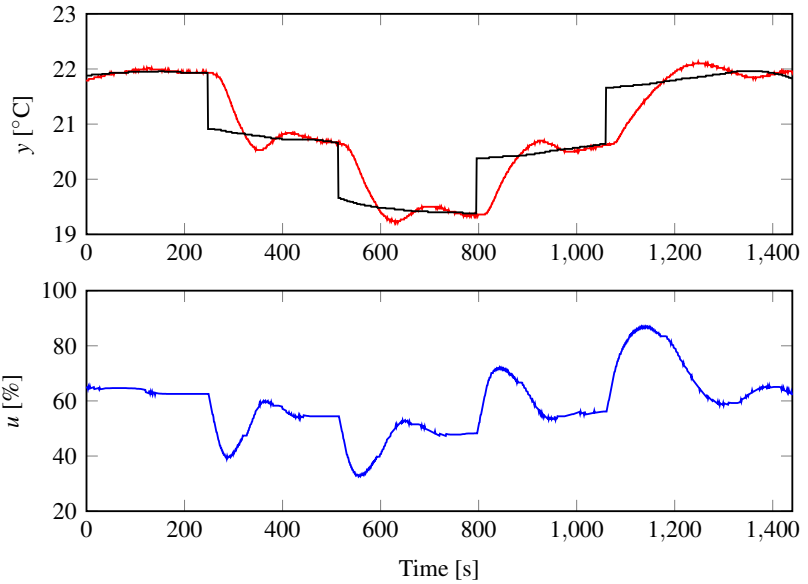


Figure 33. Response to setpoint changes. The upper plot shows the supply air temperature in red, and the setpoint in black. The lower plot shows the control signal.

the half-period ratio ρ, and as can be seen in Figure 34, 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.

5.5 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 obvious insight is that default values cannot be used unless all signals are normalized. As an example consider the hysteresis value, which default value used in simulations 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 make sense.

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

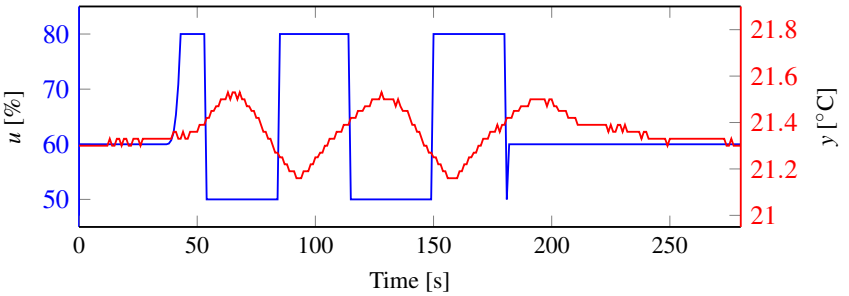


Figure 34. A relay experiment that suffers from the nonlinearities in the system. The blue line shows the normalized speed of the heat exchanger, and the red line shows the supply air temperature.

matched to give good results. However, the autotuner could give suggestions of either a suitable dead zone or low-pass filter time constant for the controller. 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 by 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 autotuner 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.

6. Conclusions

This paper proposes a simple asymmetric relay autotuner for use in the industry. The asymmetric relay function, together with some added practical features, results in an improved version of the classic relay autotuner. The improvement does not come with any added complexity or time-consumption compared to the original one. The autotuner could be further improved, at the cost of more computation, by using an advanced version with more complex modeling through parameter estimation methods and controller design by optimization. The improved excitation provided

by the asymmetry is beneficial also for the parameter estimation methods. However, the simple version used in this paper showed good results both in simulations and industrial tests.

The main disadvantage of the proposed autotuner is that it can be sensitive to uncertainties in the asymmetry level. These could be caused by either load disturbances, nonlinearities or quantization in converters. The way to handle this is to keep the experiment times short, the amplitudes small and to know the actuation levels. It, however, emphasizes the need to evaluate the obtained controller tuning. The need for short experiment times to avoid load disturbances is very important, and something that is sometimes forgotten in methods that require tens of oscillation periods in order to get the perfect model. Those algorithms are not useful in practice.

The proposed autotuner contains logic and algorithms to automatically set all parameters, find out the sign of the process gain and adapt the relay amplitudes to appropriate levels. It is shown that the results are good even in noisy environments and if the system has not really reached steady-state when the experiment is started. The tuner uses a decision scheme based on the normalized time delay to decide on model structure and controller type (PI or PID). The autotuner could also suggest sampling intervals, appropriate dead-zones or filter constants for the obtained controller from the information achieved in the experiment.

All in all, this paper provides a simple, yet fully functional, autotuner that gives well-performing PID controllers both in simulations and for real industrial systems.

Acknowledgment

The authors are members of the LCCC Linnaeus Center and the ELLIIT Excellence Center at Lund University.

A. 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 earlier in the paper, but some parameters need to be commented here. The *Large process value* 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.

Explanation	**Notation**	**Value**
Asymmetry level	γ	2
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

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.

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

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

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

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

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

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

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

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

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

References

ABB (2010). *Compact 800 Engineering, Compact Control Builder AC 800M 5.1, Binary and Analog Handling*. 3BSE041488-510. ABB.

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.

Berner, J., T. Hägglund, and K. J. Åström (2016). "Improved Relay Autotuning using Normalized Time Delay". In: *2016 American Control Conference (ACC)*. IEEE, pp. 1869–1875.

Berner, J. (2015). *Automatic Tuning of PID Controllers based on Asymmetric Relay Feedback*. Licentiate Thesis ISRN LUTFD2/TFRT--3267--SE. Dept. Automatic Control, Lund University, Sweden.

Berner, J., K. J. Åström, and T. Hägglund (2014). "Towards a New Generation of Relay Autotuners". *IFAC Proceedings Volumes* **47**:3, pp. 11288–11293.

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.

Isermann, R. (1989). *Digital Control Systems: V. 1 Fundamentals Deterministic Control*. Springer-Verlag.

Kaya, I. and D. Atherton (2001a). "Exact parameter estimation from. relay autotuning 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". *Journal of 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". *Industrial and Engineering Chemistry Research* **45**:13, pp. 4726–4731. DOI: 10.1021/ie060270b.

Li, W., E. Eskinat, and W. L. Luyben (1991). "An improved autotune identification method". *Industrial and Engineering Chemistry Research* **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". *Industrial and Engineering Chemistry Research* **43**:26, pp. 8400–8402. DOI: 10.1021/ie034043a.

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

Luyben, W. L. (1987). "Derivation of transfer functions for highly nonlinear distillation columns". *Industrial and Engineering Chemistry Research* **26**:12, pp. 2490–2495. DOI: 10.1021/ie00072a017.

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.

Shen, S.-H., J.-S. Wu, and C.-C. Yu (1996a). "Autotune Identification under Load Disturbance". *Industrial and Engineering Chemistry Research* **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 Journal* **42**:4, pp. 1174–1180. DOI: 10.1002/aic.690420431.

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 Autotuner in a Sequential Control Language". In: *2015 IEEE International Conference on Automation Science and Engineering*, pp. 874–879. DOI: 10.1109/CoASE.2015.7294191.

Paper III

Short and Robust Experiments in Relay Autotuners

Josefin Berner **Kristian Soltesz**

Abstract

This paper demonstrates how second-order time-delayed models adequate for PID controller synthesis can be identified from significantly shorter relay experiments, than used in previous publications to obtain first-order time-delayed models. Apart from having good noise robustness properties, the proposed method explicitly addresses non-stationary initial states of the dynamics to be identified, and handles constant load disturbances.

1. Introduction

1.1 Background

PID control is a widespread, well-studied, and well-understood technology, and its applications include almost all areas where closed-loop controllers are employed. There exist several text books (see for example [Åström and Hägglund, 2006]) on the topic, and several tuning rules, ranging from simple rules of thumb [Ziegler and Nichols, 1942], to optimization-based alternatives like e.g. [Garpinger and Hägglund, 2008].

All commonly used PID tuning rules rely on dynamic models of the process to be controlled. These models are almost always assumed to be linear (or local linearizations of nonlinear dynamics). Tuning of the PID controller consists in the choice of three parameters. Due to its low complexity, the controller is most often used for processes that can be adequately described by low-order models, such as the first-order time-delayed (FOTD) or second-order time-delayed (SOTD) models. If the dynamics cannot be adequately approximated by second order dynamics, it is advisable to use a more advanced controller type.

Since the tuning of PID controllers is a well-studied problem, the main challenge is often the acquisition of a model of the dynamics to be controlled. The main approaches for arriving at process models are first principle modeling, system identification, or a combination of the two. The former requires insight, while the latter relies on proper experiment design. For these reasons both approaches tend to be expensive, in terms of (experienced control engineer) man hours.

The above has motivated the development, and subsequent success, of automatic tuning procedures, which rely on automatic generation of a system identification experiment, matched to the dynamics of the process to be modeled. The most wide-spread approach is the relay auto tuner, introduced in [Åström and Hägglund, 1984]. The experiment is achieved by closing a negative feedback loop over a relay nonlinearity, as illustrated in Figure 1. In its original form, this provides an estimate of the critical frequency of the process, and its associated gain. This information is used to find controller parameters. Several extensions to the relay experiment procedure

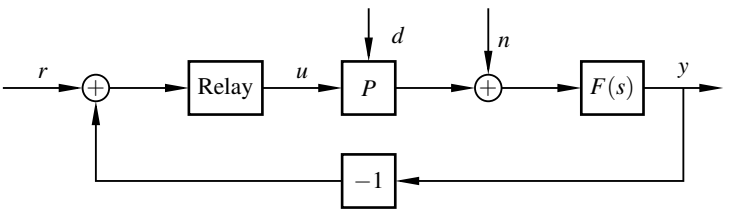


Figure 1. Block diagram of the identification setting, showing process P, asymmetric relay, filter F, identification input u and output y, load disturbance d, and measurement noise n.

have been proposed to identify FOTD models (and sometimes also SOTD models) in for instance [Kaya and Atherton, 2001; Lin et al., 2004; Luyben, 1987]. By using all data points, rather than only peak values and associated times, it is possible to successfully identify the models in a more noise-robust way. It was demonstrated in [Soltesz et al., 2016] that it suffices to use very short experiments – even under significant measurement noise – if the experiment starts in stationarity, and executes in the absence of load disturbances. These assumptions limit the applicability, as it is hard to ensure perfect stationarity prior to starting the relay experiment. It is also generally hard to safeguard for the presence of load disturbances during the relay experiment.

In this paper we present an identification procedure, which explicitly takes non-stationary initial states into account. It also handles the presence of constant load disturbances during the experiment. Thus, it provides a practically applicable extension to the relay autotuner, allowing for the identification of both FOTD and SOTD models under realistic experiment conditions, while keeping the experiment time short, and the number of heuristically determined experiment parameters low. Paired with a PID synthesis method of the user's choice, the presented method provides a complete autotuner.

1.2 Setting and Assumptions

It is assumed that the process dynamics P, to be identified, can be adequately modeled by an SOTD system $\hat{P}$, see (2).

Without loss of generality it can be assumed that $y = 0$ and $u = 0$ at the operating point of interest. Other operation points can be handled after an affine transformation. We also assume that the signals are normalized. Usually that refers to both $0 < y < 1$ and $0 < u < 1$, but since we will be oscillating around our operating point zero, we instead rescale the interval to $-5 < y < 5$ and $-5 < u < 5$. The time units are also just a matter of scaling, in this paper we will consider time units to be seconds, but it could just as well be minutes or hours depending on the application.

Both u and y are synchronously zero-order-hold sampled, with period t_s. The choice of t_s is made so that the total experiment will consist of approximately 250 samples. This is done to show that a small data buffer size is sufficient to perform the experiment, but faster sampling could of course be used if possible. Effects of discretization are neglected based on the assumption that a modern AD converter, typically with at least 16 bit resolution, is used.

2. Experiment

2.1 The relay

An asymmetric relay function, as defined in [Berner et al., 2016a] with an asymmetry level $\gamma = 2$, is used in this paper. The experiment will cause y to vary within an interval around 0. A large interval increases the signal-to-noise ratio, which is

obviously positive. However, there are several situations where it is not tolerable to move y arbitrarily far from 0 (due to constraints on the process state). For nonlinear processes, large variations may take y outside the interval around 0, within which the process can be adequately described by linear dynamics. Ideally, it would therefore be preferable to specify bounds on the admissible interval. However, this is not possible since P of Figure 1 is unknown. Consequently, we will instead specify an admissible interval for u. The relay amplitudes are set in the startup of the experiment, as in [Berner et al., 2016a], but are restricted to the admissible interval. In this paper we have used maximum values of u that correspond to a control signal interval of $u_{\max} - u_{\min} = 1$, i.e., 10% of the control signal range. Consequently the larger relay amplitude is restricted to 0.67 and the smaller to 0.33 for the given asymmetry level $\gamma = 2$. For a well-designed process the steady-state gain between actuator and sensor should be close to unity, which would result in y varying within 10% as well. The startup procedure is supposed to take care of the cases where the process is not as well-designed. However, if the user has other information or restrictions, it could be used to set the admissible interval of u accordingly.

The main experiment used throughout this paper starts with measuring the noise level and setting a hysteresis band, and is terminated once the relay has switched $M = 3$ times. The number of relay switches constitutes a trade-off between input excitation (both in term of spectral concentration and signal-to-noise ratio) and experiment duration. A motivation to the short experiment duration is given in Section 4.

2.2 Noise and Disturbances

The sensor model consists of an additive noise source n, which can be regarded stationary and white in the frequency band of interest for identification. The noise assumption is motivated by the nature of commonly occurring (thermal) sensor noise. The noise is measured during the startup of the experiment, and the hysteresis level h is then set to 3 times the noise level to prevent the noise from causing the relay to chatter. Since a sufficiently high signal-to-noise ratio is required, and the signal is restricted to lie within the admissible interval, there may be a need of filtering the noise if its amplitude is too high. This could be done by introducing a low-pass filter $F(s)$, as in Figure 1. By using the filtered output signal, together with the input signal run through the same filter, the identification could be performed in the same way as for the unfiltered case.

Ever since the introduction of the relay autotuner in [Åström and Hägglund, 1984] it has been assumed and required that the user starts the experiment when the system is in steady-state. In [Berner et al., 2016a] it was shown that small deviations from steady-state did not deteriorate the resulting models that much, as long as steady-state was reached during the startup of the experiment. To ensure that the system is started in *total* stationarity is not practically possible, and to know what is a sufficiently small deviation is hard. Therefore a way of taking care of initial states

Table 1. Experiment parameters used in simulation.

Parameter	Value	Description
γ	2	Relay asymmetry
Δu_{max}	1	Control signal interval, $u_{\max} - u_{\min}$
M	3	Number of relay switches
N	≈ 250	Number of samples per experiment
t_s		Sample time, adjusted to get $N \approx 250$
h	3	Hysteresis to noise ratio
σ_n	0.1	Noise standard deviation
v_0	0.08	Offset in control signal for initial state x_0

separate from zero is added to the identification method in this paper and described in Section 3.4.

Unknown load disturbances are a large problem for relay experiments. This is the main motivation for keeping the experiment time as short as possible. This method is, however, not as sensitive to load disturbances as for instance the one in [Berner et al., 2016a]. If the constant load disturbance has been present for a long time it will only change the nominal control signal level, which will not affect the experiment at all. If the load disturbance enters just before (or exactly at) the starting point of the experiment, it will have the same effect as a change in initial state, which the proposed method handles explicitly. If, on the other hand, the load disturbance enters or changes during the experiment it will still cause problems. The risk for this to occur is limited by the very short experiment duration.

2.3 Experiment parameters

The parameters used for the experiments in Section 4 are listed in Table 1. In addition to the parameters in the table, initialization of the experiment consists of $u = 0$ during one time unit, to characterize measurement noise, followed by u being exponentially increased towards $u_{\max}$ during the next time unit. Those timings differ from the parameters used in [Berner et al., 2016a] but are reasonable for these experiments.

The initial state is set to

$$\boldsymbol{x_0} = -A^{-1} B v_0 \Delta u_{max}, \tag{1}$$

where v_0 is the control signal corresponding to x_0 in stationarity, and A and B are the state space matrices of the processes used in the simulation examples.

3. Identification

3.1 Model Parametrization

The models we consider in this paper are of the form

$$\hat{P}(s) = \frac{b_1 s^{m-1} + b_2 s^{m-2} + \ldots + b_m}{s^n + a_1 s^{n-1} + \ldots + a_n} e^{-sL}, \tag{2}$$

where $1 \leq m \leq n \leq 2$. The restriction $n \leq 2$ allows FOTD models, SOTD models, as well as second-order time-delayed models with a zero (SOTDZ). With the chosen parametrization all these model types could be integrating by setting $a_n = 0$.

As motivated in Section 2.2, we want to estimate the initial state(s) $\boldsymbol{x_0}$ in addition to the model parameters, to be robust toward not starting in total stationarity. The parameters that will be estimated are therefore $\boldsymbol{\theta} = \begin{bmatrix} \boldsymbol{b} & \boldsymbol{a} & L & \boldsymbol{x_0} \end{bmatrix}$, where $\boldsymbol{a} = \begin{bmatrix} a_1 & \ldots & a_n \end{bmatrix}$, $\boldsymbol{b} = \begin{bmatrix} b_1 & \ldots & b_m \end{bmatrix}$, and $\boldsymbol{x_0} = \begin{bmatrix} x_1(0) & \ldots & x_n(0) \end{bmatrix}$.

3.2 Output Error Formulation

The output data from the experiment on the process P is collected in $\boldsymbol{y} = \begin{bmatrix} y_1 & \ldots & y_N \end{bmatrix}^\top$, and the corresponding output data vector for the estimated process $\hat{P}$ is denoted $\hat{\boldsymbol{y}}$.

In this paper an output error ($\mathcal{L}_2$) method is employed to identify a $\boldsymbol{\theta}$, which (locally) minimizes the cost

$$J(\boldsymbol{\theta}) = \frac{t_s}{2} \boldsymbol{\varepsilon}^\top \boldsymbol{\varepsilon}, \tag{3}$$

where $\boldsymbol{\varepsilon} = \hat{\boldsymbol{y}} - \boldsymbol{y}$. An interior-point method[1] [Byrd et al., 2000] is employed to find a (local) minimum of (3). Like most local optimization methods, convergence properties of the proposed method are significantly improved if exact expressions for the Jacobian

$$\nabla J(\boldsymbol{\theta}) = t_s \boldsymbol{\varepsilon} (\nabla \hat{\boldsymbol{y}})^\top, \tag{4}$$

and corresponding Hessian

$$\Delta J(\boldsymbol{\theta}) = \frac{t_s}{2} \Delta \left(\boldsymbol{\varepsilon}^\top \boldsymbol{\varepsilon} \right) = t_s (\nabla \hat{\boldsymbol{y}})^\top \nabla \hat{\boldsymbol{y}} + t_s \boldsymbol{\varepsilon}^\top \Delta \hat{\boldsymbol{y}}, \tag{5}$$

are available (as opposed to finite-difference approximations). In the context of this paper, the nabla operator is defined as

$$\nabla = \begin{bmatrix} \nabla_{\boldsymbol{b}} & \nabla_{\boldsymbol{a}} & \dfrac{\partial}{\partial L} & \nabla_{\boldsymbol{x}_0} \end{bmatrix}, \tag{6}$$

where

$$\nabla_{\boldsymbol{b}} = \begin{bmatrix} \dfrac{\partial}{\partial b_1} & \ldots & \dfrac{\partial}{\partial b_m} \end{bmatrix}, \tag{7}$$

[1] The method has been invoked from the Matlab `fmincon` function with solvers `trust-region-reflective` and `sqp` for results in this paper.

and where $\nabla_{\boldsymbol{a}}$ and $\nabla_{\boldsymbol{x}_0}$ are defined analogously. The Laplace operator is defined through the outer product $\Delta = \nabla^\top \nabla$.

Based on the reasonable assumption that $\boldsymbol{\varepsilon}$ and $\Delta\hat{\boldsymbol{y}}$ are generally uncorrelated, while $(\nabla\hat{\boldsymbol{y}})^\top \nabla\hat{\boldsymbol{y}} \succeq 0$, it was suggested in [Åström, 1980] to approximate the Hessian by the positive semi-definite term, when considering output error $\mathcal{L}_2$ problems. We will adopt this approximation, and with a slight abuse of notation (re)define

$$\Delta J(\boldsymbol{\theta}) = t_s (\nabla\hat{\boldsymbol{y}})^\top \nabla\hat{\boldsymbol{y}}. \tag{8}$$

3.3 State-space formulation

To find the gradients needed for the Jacobian (and Hessian) for the identification method we will consider a state-space representation of an augmented system, with output

$$\hat{\boldsymbol{y}}_e = \begin{bmatrix} 1 & \nabla_{\boldsymbol{b}} & \nabla_{\boldsymbol{a}} & \dfrac{\partial}{\partial L} \end{bmatrix} \hat{\boldsymbol{y}}, \tag{9}$$

that is, a system that in addition to $\hat{\boldsymbol{y}}$ also outputs its gradients with respect to the model parameters.

We will be using the notation $0_{i\times j}$ for the zero matrix with i rows and j columns, and $I_{i\times j}$ for the identity matrix where, assuming $i \leq j$, the last $j - i$ rows have been removed. If only one index is given, the matrix is assumed to be square.

The un-delayed version of the original system (2) can be written in state-space form as

$$\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right] = \left[\begin{array}{c|c} -\boldsymbol{a} & 1 \\ I_{n-1\times n} & 0_{n-1\times 1} \\ \hline \tilde{\boldsymbol{b}} & 0 \end{array}\right], \tag{10}$$

where

$$\tilde{\boldsymbol{b}} = \begin{bmatrix} 0_{1\times n-m} & \boldsymbol{b} \end{bmatrix} \tag{11}$$

is a zero-padded version of $\boldsymbol{b}$, matching the dimension of $\boldsymbol{a}$.

Since the experiment data $\boldsymbol{u}$ and $\boldsymbol{y}$ are zero-order-hold sampled, it will suffice to consider the correspondingly discretized version of (10), with system matrices $\{\Phi, \Gamma, \tilde{\boldsymbol{b}}, 0\}$, where

$$\{\Phi, \Gamma\} = \left\{ e^{At_s}, \int_0^{t_s} e^{At}\, dt B \right\}. \tag{12}$$

Our model is thus given by

$$\begin{aligned} \boldsymbol{x}(k+1) &= \Phi\boldsymbol{x}(k) + \Gamma u_d(k),\ \boldsymbol{x}(0) = \boldsymbol{x}_0, \\ \hat{y}(k) &= \tilde{\boldsymbol{b}}\,(\boldsymbol{x}(k) - \boldsymbol{x}_0), \end{aligned} \tag{13}$$

where $u_d(k)$ are elements of

$$\boldsymbol{u_d} = q^{-k_L} \boldsymbol{u}. \tag{14}$$

In (14), q^{-1} is the (non-circular) backward shift operator, and k_L the integer closest to L/t_s.

The contribution $\tilde{\boldsymbol{b}}\boldsymbol{x}_0$ from the initial state to the output y is subtracted, to be consistent with the experiment described in Section 2, which is expected to start with the output of P being 0.

The initial state estimate $\boldsymbol{x}_0$ lacks interpretation, as generally, the structure (or even order) of our model (13) does not match that of the process P to be identified. In fact, the only use of $\boldsymbol{x_0}$ is to improve the other parameter estimates.

Expressions for the sensitivities with respect to the model parameters have been presented previously in [Soltesz et al., 2010]. Results for the discrete time counterparts are found in [Åström and Bohlin, 1966]. In this paper, simplified expressions of those in [Soltesz et al., 2010], valid under the equi-temporal zero-order-hold sampling assumption, are used. To make the sensitivity computations tractable, we assume that u is independent of $\boldsymbol{x}$. This is a fair approximation, given the experiment of Section 2, where the process operates in open-loop, except at the time instances when the relay switches.

The matrix

$$\hat{\boldsymbol{y}}_e = \begin{bmatrix} 1 & \nabla_{\boldsymbol{b}} & \nabla_{\boldsymbol{a}} & \dfrac{\partial}{\partial L} \end{bmatrix} \hat{\boldsymbol{y}} \tag{15}$$

is obtained as the output of the system

$$\begin{aligned} \boldsymbol{z}(k+1) &= \Phi_e \boldsymbol{z}(k) + \Gamma_e u_d(k), \\ \hat{y}_e(k) &= C_e(\boldsymbol{z}(k) - \boldsymbol{z}_0) + D_e w(k), \end{aligned} \tag{16}$$

where the extended state vector $\boldsymbol{z}$ is

$$\boldsymbol{z} = \begin{bmatrix} \boldsymbol{x} \\ -\nabla_{\boldsymbol{a}} \boldsymbol{y} \end{bmatrix}, \tag{17}$$

$\boldsymbol{z}_0$ is the zero padded initial model state

$$\boldsymbol{z}(0) = \begin{bmatrix} \boldsymbol{x}_0 \\ 0_{n\times 1} \end{bmatrix}, \tag{18}$$

and the system matrices of (16) are the discretized counterparts of

$$\left[\begin{array}{c|c} A_e & B_e \\ \hline C_e & D_e \end{array}\right] = \left[\begin{array}{ccc|c} A & & 0_n & B \\ \begin{bmatrix} \tilde{\boldsymbol{b}} \\ 0_{n-1\times n} \end{bmatrix} & & A & 0_{n\times 1} \\ \hline C & & 0_{1\times n} & D \\ \begin{bmatrix} 0_{m\times n-m} & I_m \end{bmatrix} & & 0_{m\times n} & 0_{m\times 1} \\ 0_n & & -I_n & 0_{n\times 1} \\ \tilde{\boldsymbol{r}} & & 0_{1\times n} & \boldsymbol{q} \end{array}\right]. \tag{19}$$

The extended system matrices $\{\Phi_e, \Gamma_e\}$ relate to $\{A_e, B_e\}$ as $\{\Phi, \Gamma\}$ relate to $\{A, B\}$, see (12). The row vectors $\boldsymbol{q}$ and $\tilde{\boldsymbol{r}}$ in (19) are used in the computation of $\partial\hat{\boldsymbol{y}}/\partial L$. They are defined through the quotient $\boldsymbol{q}$ and remainder $\boldsymbol{r}$ of the polynomial division, or equivalently the deconvolution, of the vectors $\begin{bmatrix}-\tilde{\boldsymbol{b}} & 0\end{bmatrix}$ and $\begin{bmatrix}1 & \boldsymbol{a}\end{bmatrix}$, where $\tilde{\boldsymbol{r}}$ is $\boldsymbol{r}$ with its first element removed. The origin of these expressions is found in [Soltesz et al., 2010].

3.4 Initial State Sensitivity

By definition $\nabla_{\boldsymbol{x}_0}\boldsymbol{x}(0) = I_n$. The assumption made in Section 3.3, that u is independent of $\boldsymbol{x}$, results in $\nabla_{\boldsymbol{x}_0}u_d$ being uniformly zero, and from the state update equation of (13) we consequently obtain $\nabla_{\boldsymbol{x}_0}\boldsymbol{x}(k) = \Phi^k$. Combining this with the output equation of (13), and again utilizing that $\nabla_{\boldsymbol{x}_0}\boldsymbol{x}(0) = I_n$, yields $\nabla_{\boldsymbol{x}_0}\hat{y}(k) = \tilde{\boldsymbol{b}}\Phi^k - \tilde{\boldsymbol{b}}$. This expression can be obtained recursively through simulation of the system

$$\begin{aligned}\boldsymbol{w}(k+1) &= \Phi^\top \boldsymbol{w}(k),\ \boldsymbol{w}(0) = \tilde{\boldsymbol{b}}^\top,\\ \nabla_{\boldsymbol{x}_0}\hat{y}(k) &= \boldsymbol{w}^\top(k) - \tilde{\boldsymbol{b}}.\end{aligned} \tag{20}$$

3.5 Calculating the gradient expressions

The expressions for the gradients have been previously derived in [Soltesz et al., 2010], but to make it clearer for the readers we exemplify the calculations in a slightly different way here. The SOTDZ case, where $n = m = 2$, gives

$$Y = \frac{b_1 s + b_2}{s^2 + a_1 s + a_2} U_d, \tag{21}$$

where the delayed input is defined as $U_d = e^{-sL}U$. By introducing the states

$$X_1 = \frac{s}{s^2 + a_1 s + a_2} U_d, \tag{22}$$

$$X_2 = \frac{1}{s^2 + a_1 s + a_2} U_d, \tag{23}$$

(21) can be written on state-space form as

$$\begin{aligned}\dot{x} &= \begin{bmatrix}-a_1 & -a_2\\ 1 & 0\end{bmatrix} x + \begin{bmatrix}1\\0\end{bmatrix} u_d\\ y &= \begin{bmatrix}b_1 & b_2\end{bmatrix} x,\end{aligned} \tag{24}$$

which corresponds to the system in (10). To find the gradients we extend the state-space system to get the output vector

$$\boldsymbol{y_e} = \begin{bmatrix}\boldsymbol{y} & \nabla_b \boldsymbol{y} & \nabla_a \boldsymbol{y} & \dfrac{\partial \boldsymbol{y}}{\partial L} & \nabla_{x_0}\boldsymbol{y}\end{bmatrix}. \tag{25}$$

The gradient with respect to $\boldsymbol{b}$ is given by

$$\begin{aligned}\frac{\partial Y}{\partial b_1} &= \frac{s}{s^2 + a_1 s + a_2} U_d = X_1,\\ \frac{\partial Y}{\partial b_2} &= \frac{1}{s^2 + a_1 s + a_2} U_d = X_2,\end{aligned} \tag{26}$$

hence the state-space representation for the output $\nabla_b \boldsymbol{y}$ is the same as for y with exception that the C-matrix is now I_2.

The gradient with respect to a is given by

$$\begin{aligned}\frac{\partial Y}{\partial a_1} &= -s\frac{b_1 s + b_2}{(s^2 + a_1 s + a_2)^2} U_d = -\frac{s}{s^2 + a_1 s + a_2} Y,\\ \frac{\partial Y}{\partial a_2} &= -\frac{b_1 s + b_2}{(s^2 + a_1 s + a_2)^2} U_d = -\frac{1}{s^2 + a_1 s + a_2} Y.\end{aligned} \tag{27}$$

By introducing the additional states

$$z_1 = -\frac{\partial y}{\partial a_1}, \quad z_2 = -\frac{\partial y}{\partial a_2},$$

we get

$$\begin{aligned}\dot{z_1} &= -a_1 z_1 - a_2 z_2 + y = -a_1 z_1 - a_2 z_2 + b_1 x_1 + b_2 x_2\\ \dot{z_2} &= z_1.\end{aligned} \tag{28}$$

The state-updates in (28) coincides with those for $\boldsymbol{z}$ in (19), and since the output equals $-\boldsymbol{z}$ the C-matrix is $-I_2$.

The gradient with respect to L is

$$\frac{\partial Y}{\partial L} = -s\frac{b_1 s + b_2}{s^2 + a_1 s + a_2} e^{-sL} U \tag{29}$$

The numerator can be rewritten as

$$b_1 s^2 + b_2 s = b_1(s^2 + a_1 s + a_2) + b_2 s - b_1 a_1 s - b_1 a_2 \tag{30}$$

resulting in

$$\frac{b_1 s^2 + b_2 s}{s^2 + a_1 s + a_2} = b_1 + \frac{(b_2 - b_1 a_1)s - b_1 a_2}{s^2 + a_1 s + a_2}. \tag{31}$$

This gives the following expression for the gradient:

$$\begin{aligned}\frac{\partial Y}{\partial L} &= -b_1 U_d - \frac{(b_2 - b_1 a_1)s - b_1 a_2}{s^2 + a_1 s + a_2} U_d\\ &= -b_1 U_d - (b_2 - b_1 a_1) X_1 + b_1 a_2 X_2.\end{aligned} \tag{32}$$

The quotient b_1 becomes part of a direct term, while the remainder from the polynomial division enters as the C-matrix of the original states $\boldsymbol{x}$. Since the polynomial

division in (31) is equal to the deconvolution of the vectors $\begin{bmatrix} -\tilde{\boldsymbol{b}} & 0 \end{bmatrix}$ and $\begin{bmatrix} 1 & \boldsymbol{a} \end{bmatrix}$ this is in accordance with (19).

The gradient with respect to $\boldsymbol{x_0}$ is simulated from a separate system, as described in Section 3.4.

3.6 Intializing the identification

The identification method needs to be started with an initial guess of the parameter vector $\boldsymbol{\theta}$. Unfortunately, starting from the zero-vector, as would be the first attempt, does not always work out well. Most of the times a good model is found from that starting vector, but sometimes the algorithm get stuck in another point. By using two different solvers in the Matlab `fmincon` method some of these problems were removed, but still there is a need for initializing the identification differently. We have chosen to initialize the system by starting the FOTD estimation from the zero-vector, as well as a number (in this study 30) of randomly chosen $\begin{bmatrix} \boldsymbol{b} & \boldsymbol{a} & \boldsymbol{L} \end{bmatrix}$ vectors. The reason for taking random points instead of a grid of the parameters, is that some parameters may be more significant than others, and the random pick gives more options for each parameter, as described in [Bergstra and Bengio, 2012]. The interval for the random choices were restricted with help from the maximum time delay $L_{\max}$, and normalized time delay τ, which can both be roughly estimated from the obtained half-period intervals. $L_{\max}$ is simply chosen as the shortest duration between two consecutive relay switches. For the estimate of τ we refer to [Berner et al., 2016b]. However, that method requires convergence of the experiment, and does not support $\boldsymbol{x_0} \neq \boldsymbol{0}$, which suggests that the value of τ we obtain here is very approximate.

The initialization of the SOTD model is based on the obtained FOTD model, and the initialization of the SOTDZ model is based on the obtained SOTD model.

The estimate of the initial state $\boldsymbol{x_0}$ is initialized to zero for the FOTD model, and then either started from zero, or based on the obtained $\boldsymbol{x_0}$-estimate, for the higher order models.

3.7 Model Selection

Both an FOTD, an SOTD and an SOTDZ model are estimated for each process. The choice of which model to use is then based on the Akaike Information Criteria (AIC) [Akaike, 1974]. The chosen model is the one with lowest value of

$$J_{AIC} = \log(J) + \frac{2p}{N}, \tag{33}$$

where p is the number of model parameters and N the number of data samples. The AIC is known to sometimes choose over-parametrized models, and there may be better model selection tools, but that is not the focus of this paper.

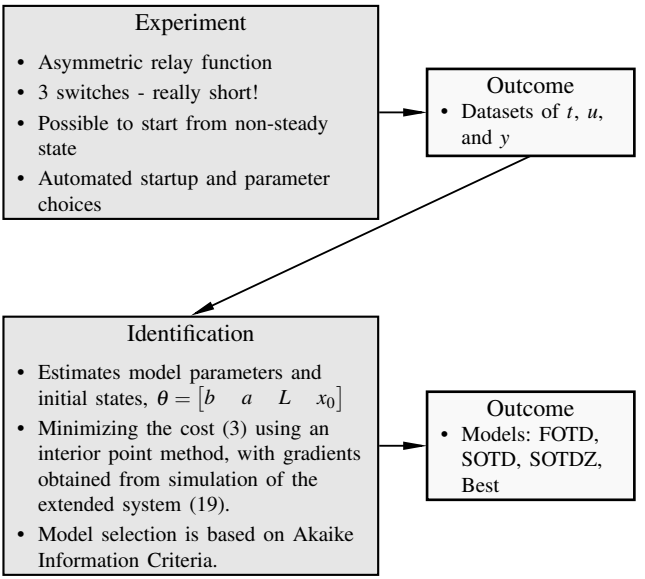


Figure 2. Schematic summary of the proposed experiment and identification method.

4. Simulation Study

The proposed method, briefly summarized in Figure 2, is evaluated by four example processes from the test batch in [Åström and Hägglund, 2006], namely:

$$P_1 = \frac{1}{(s+1)(0.1s+1)(0.01s+1)(0.001s+1)}, \tag{34}$$

$$P_2 = \frac{1}{(s+1)^4}, \tag{35}$$

$$P_3 = \frac{1}{(0.05s+1)^2} e^{-s}, \tag{36}$$

$$P_4 = \frac{1-0.5s}{(s+1)^3}. \tag{37}$$

These examples were chosen due to their differing properties. P_1 is lag-dominated, P_2 is balanced, P_3 is delay-dominated, and P_4 is non-minimum phase. P_1–P_3 have been used as example processes in several other papers, for instance [Berner et al., 2016b].

The outputs from the experiments are shown in Figure 3. As can be seen, the experiments are short and noisy, but the obtained models fit the data very well. It can be seen for P_2 and P_4 that the FOTD models do not perfectly fit the data,

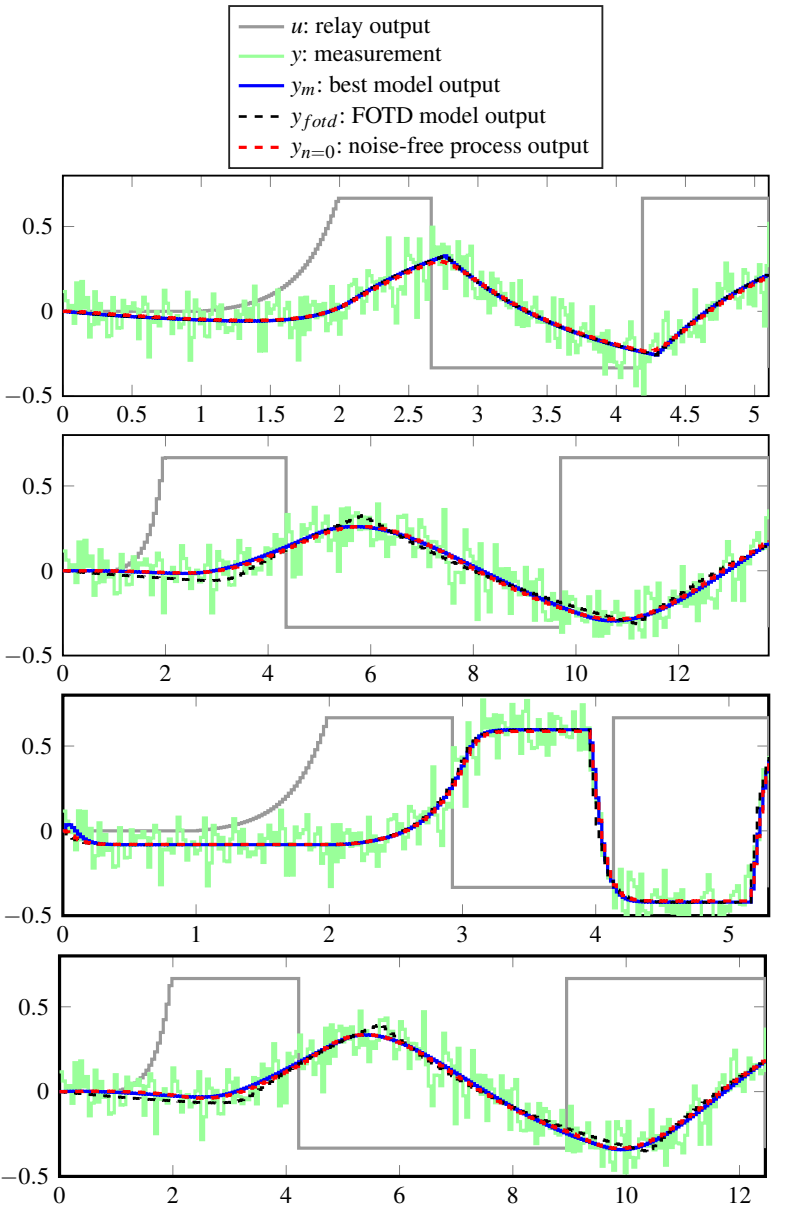


Figure 3. Outputs from the experiments and identified models. The different subplots show P_1, P_2, P_3, and P_4 from top down.

while the best models seem to do. The "best model" was chosen using AIC, see Section 3.7. Figure 4 shows Bode plots of the different estimated models, together with those of the processes. The estimated models constitute good approximations of the processes, for frequencies up to phase lags of $-180°$, which are the relevant frequencies for PID control. The only exception is that the FOTD model is chosen as the best model for P_1 even though it is seen that the SOTD model follows the magnitude curve much better. This issue will be discussed further in Section 5.

The benefit of using the initial state estimation is demonstrated in Figure 5. Here the best models obtained when $\boldsymbol{x_0}$ is being estimated are compared to the best models when it is not. While the assumption that $\boldsymbol{x_0} = \mathbf{0}$ yields acceptable results in some cases, the models generally improve when $\boldsymbol{x_0}$ is explicitly estimated.

Estimation of the initial state(s) could introduce problems, since it adds more parameters to be estimated, and model dynamics may be wrongly interpreted as initial state(s). Therefore we also investigated the case where the initial state estimation was active while the experiment started in stationarity. The results from this test showed a slight deterioration in one of the obtained models, while the other three were satisfactory. To avoid this possible problem the models could be identified both with and without the initial-state estimation active, and then the best model could be picked according to AIC, as is done for the different model orders.

Another issue to consider is whether the experiments are sufficiently long or if the results would improve from longer experiments. In Figure 6 the obtained results are compared to those from experiments utilizing 5 relay switches. As can be seen the difference in obtained models between the different experiment lengths are very small.

5. Discussion

The simulation study shows that FOTD and SOTD models can be very well estimated even in the presence of noise and non-stationary starting conditions, including constant load disturbances. The limits for the initial states are mainly that they cannot be so large that the "true" nominal level is close to (or outside) the hysteresis limits. If it is, the output may leave the hysteresis band on the wrong side, which could cause a non-oscillating experiment.

The model selection by AIC does not always give the best result. As shown in Figure 4, P_1 gets an FOTD as its best model, while $P_2 - P_4$ get SOTD models. For P_1 the plot indicates that the SOTD model is actually better and should have been chosen. On the other hand, P_3 would be just as good with the FOTD model, but there the SOTD model is chosen instead. A large part of the obtained cost is due to the large noise level, which makes the relative difference between the obtained models rather small, and sometimes that results in the "best" model not being picked. A possible way to handle this more robustly is to include information about the normalized time delay in the model choice. In [Åström and Hägglund,

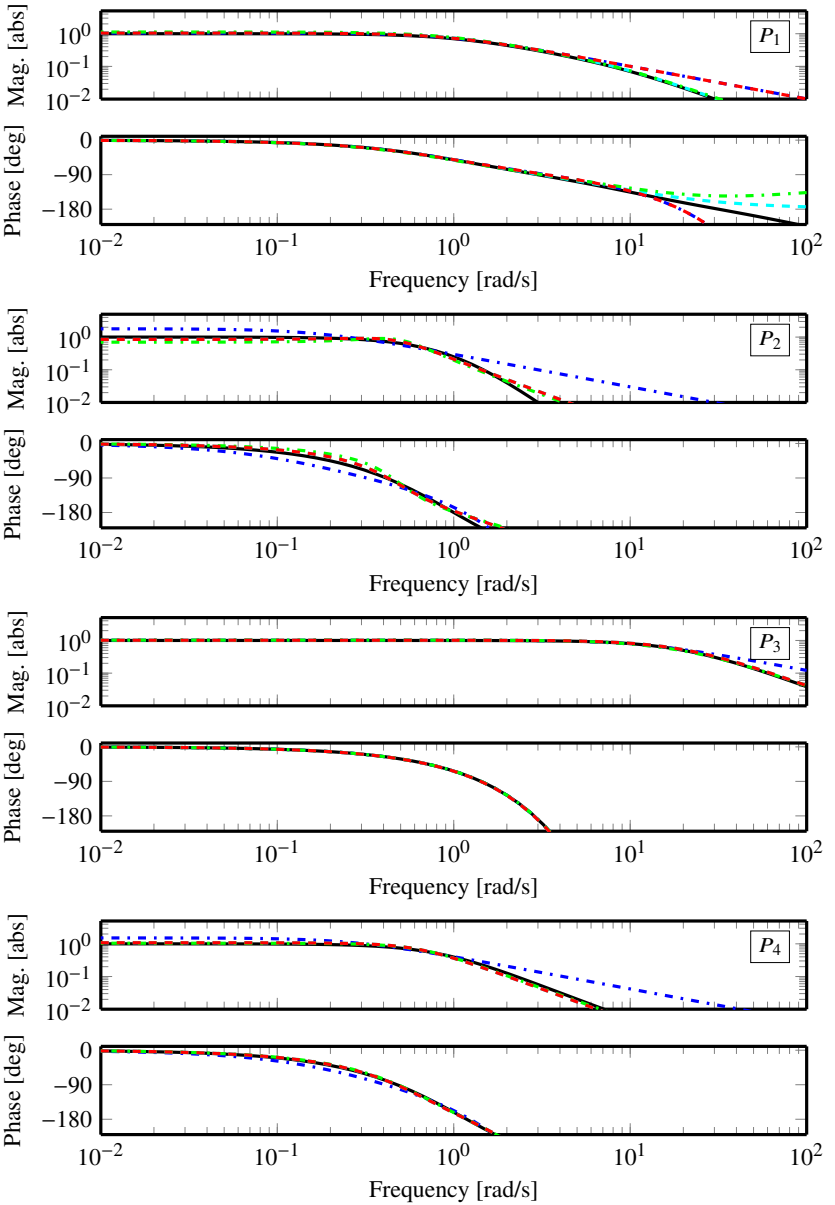


Figure 4. Bode plots of the processes and identified models. The true process is shown in solid black, the FOTD model is shown in dashed-dotted blue, the SOTD model in dashed cyan, and the SOTDZ model in dashed-dotted green. The best model according to AIC is shown in dashed red.

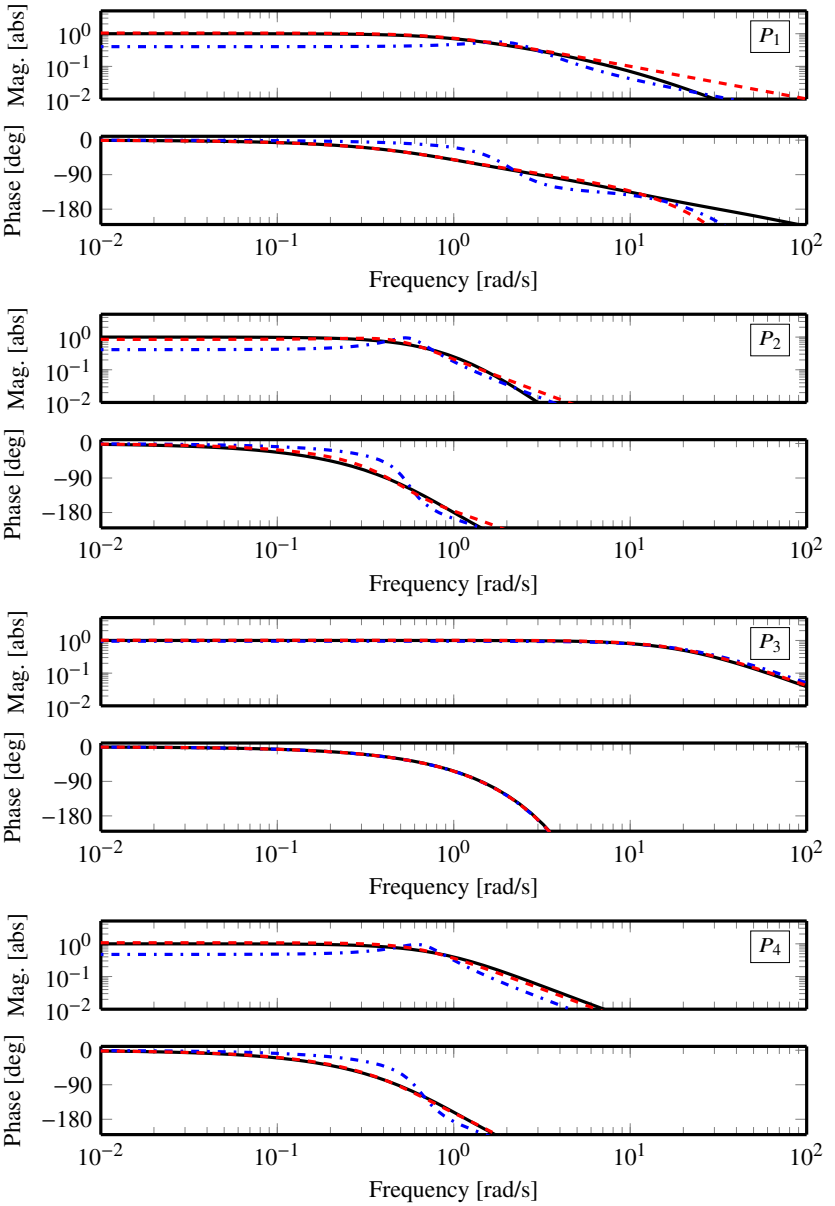


Figure 5. Bode plots of obtained models for the example processes. The estimation of the initial state(s) $\boldsymbol{x_0}$ was active in the red dashed model and inactive in the blue dashed-dotted model. The true process is shown in solid black.

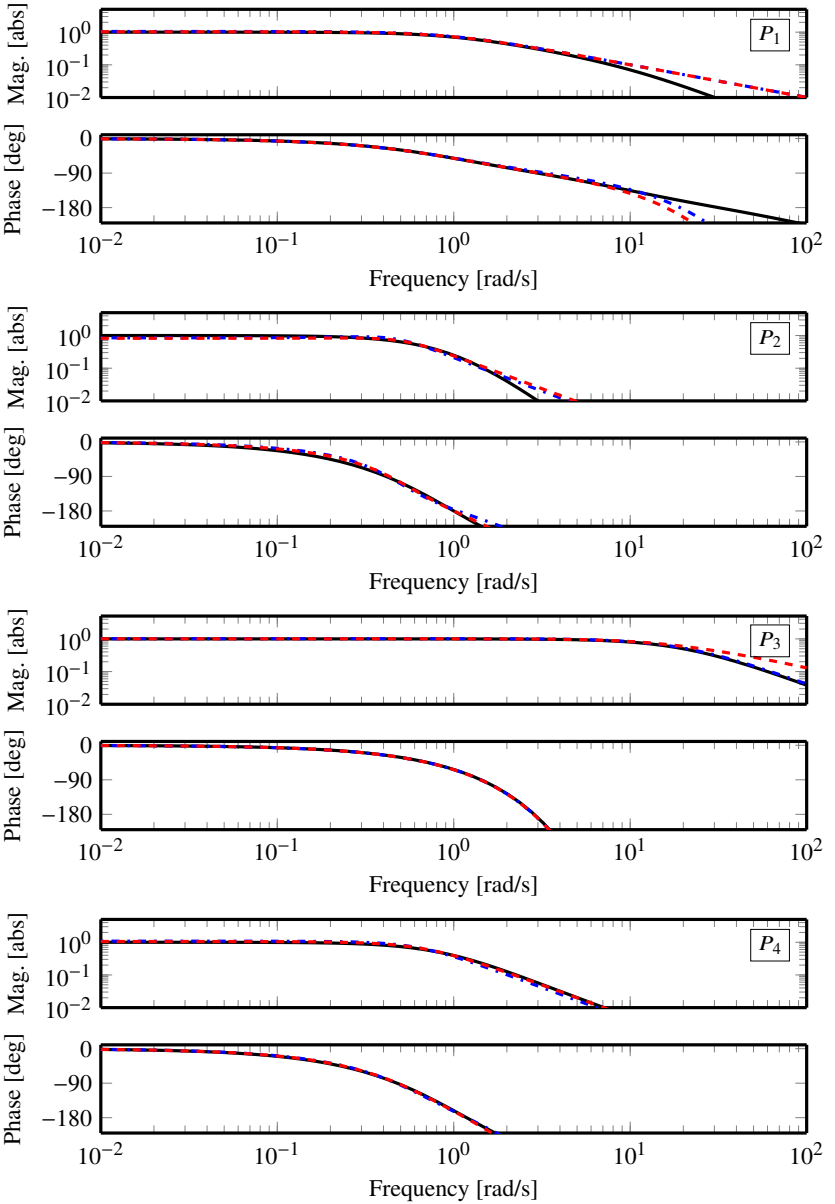


Figure 6. Bode plots of the obtained models for the example processes. The best model from an experiment of 3 switches is shown in dashed-dotted blue, and 5 switches shown in dashed red. The true process is shown in solid black.

2006] and [Berner et al., 2016b], it is discussed how delay-dominated systems like P_3 are sufficiently described by an FOTD model, while lag-dominated systems like P_1 could sometimes be much better described by SOTD models.

One would think that P_4 should get an SOTDZ model as it can capture the non-minimum-phase zero. However, the cost was exactly the same for the SOTDZ model as for the SOTD model, and hence not low enough to make it the chosen model since it has more parameters. Apparently the experiment is not showing the non-minimum-phase behavior enough, or its influence is instead interpreted as a different initial state or included in the time-delay. Since the output data fit of the SOTD model is already more or less perfect, the feeling is that it cannot be improved much by the SOTDZ model, and additional tests on finding zeros from the experiment are not showing very promising results. Due to this, our recommendation is to stick to estimating FOTD and SOTD models only. If an SOTDZ model is really needed, for instance if the process has slow zeros that need to be cancelled out by the controller, the experiment needs to be re-designed to better capture the characteristics of the zeros.

We want the experiment to contain at least an entire oscillation period, and increasing the experiment length did not change the obtained models much, which implies that a duration of three switches is sufficient. The short experiment length prevents the risk of disturbance changes during the experiment and is therefore important.

6. Conclusions and Future Work

The proposed method works well in finding FOTD and SOTD models from short experiments. The experiments can be started without waiting for steady-state since we estimate the initial states, and are ended without waiting for limit cycle convergence. Constant load disturbances that enter before the experiment are taken care of, but if something happens during the experiment we could still be in trouble. That's why the really short experiment time is beneficial. The obtained models are sufficient for PID control. By improving the model selection, the proposed method could yield even better results.

The proposed experiment and identification needs to be combined with a tuning method to result in a complete autotuner. This should then be evaluated and compared to other autotuners, preferably on real-world processes.

Acknowledgment

The authors would like to thank Tore Hägglund and Karl Johan Åström for interesting discussions regarding this work. The authors are members of the LCCC Linnaeus Center and the ELLIIT Excellence Center at Lund University.

References

Akaike, H. (1974). "A new look at the statistical model identification". *IEEE transactions on automatic control* **19**:6, pp. 716–723.

Åström, K. J. and T. Bohlin (1966). "Numerical identification of linear dynamic systems from normal operating records". In: *Theory of self-adaptive control systems*. Springer, pp. 96–111.

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. (1980). "Maximum likelihood and prediction error methods". *Automatica* **16**:5, pp. 551–574.

Bergstra, J. and Y. Bengio (2012). "Random search for hyper-parameter optimization". *Journal of Machine Learning Research* **13**:Feb, pp. 281–305.

Berner, J., T. Hägglund, and K. J. Åström (2016a). "Asymmetric relay autotuning–Practical features for industrial use". *Control Engineering Practice* **54**, pp. 231–245.

Berner, J., T. Hägglund, and K. J. Åström (2016b). "Improved Relay Autotuning using Normalized Time Delay". In: *2016 American Control Conference (ACC)*. IEEE, pp. 1869–1875.

Byrd, R., J. Gilbert, and J. Nocedal (2000). "A trust region method based on interior point techniques for nonlinear programming". *Mathematical Programming* **89**:1, pp. 149–185.

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

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

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

Luyben, W. L. (1987). "Derivation of transfer functions for highly nonlinear distillation columns". *Industrial and Engineering Chemistry Research* **26**:12, pp. 2490–2495. DOI: 10.1021/ie00072a017.

Soltesz, K., T. Hägglund, and K. J. Åström (2010). "Transfer function parameter identification by modified relay feedback". In: *American Control Conference*. Baltimore, MD, pp. 2164–2169.

Soltesz, K., P. Mercader, and A. Baños (2016). "An automatic tuner with short experiment and probabilistic plant parameterization". *International Journal of Robust and Nonlinear Control*.

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

Paper IV

An Experimental Comparison of PID Autotuners

Josefin Berner Kristian Soltesz Tore Hägglund

Karl Johan Åström

Abstract

In this paper two novel autotuners are compared with two industrially available ones. The aim is to see if the research frontline can improve the industry standard of today. Experiments are made on three laboratory processes with different characteristics. Two lag-dominated processes of which one is a level control problem with fast dynamics, and one a temperature control problem with slow dynamics, as well as one delay-dominated level control process. Both the experiments and the obtained controller performances are evaluated and discussed. The results show that the performance of the state-of-the-art industrial autotuners can be significantly improved.

Keywords: PID control, Automatic tuning, Process industry, Relay feedback, Comparative study.

Manuscript submitted to journal

1. Introduction

Automatic tuning of PID controllers is a useful feature for any user who does not have the time, knowledge or desire to manually tune his or her control loops. Especially in the process industry, where a factory may have hundreds or thousands of different flows, levels, temperatures, concentrations etc. that need to be controlled, the benefit of a fast and reliable way of finding appropriate controller parameters is large. One such procedure that is common in industry is the relay autotuner.

The principle of relay autotuning is as follows. By closing the feedback loop with a relay function, that switches between two values depending on the process output, the process is forced into oscillations. See Figure 1 for the setup and Figure 2 for a typical experiment output. From the oscillations process data can be obtained and used to tune a PID controller. The original relay autotuner [Åström and Hägglund, 1984] uses the period time and amplitude of the induced oscillations in order to find the critical point where the process Nyquist plot intersects the negative real axis. If a hysteresis band is added to reduce shattering due to noise, a slightly different point is obtained. The controller parameters are then found by moving this point to give the open-loop system specified amplitude and phase margins.

The autotuner from [Åström and Hägglund, 1984], that was developed in the 1980's, is probably still the most common one in industrial DCS systems today. Several of the major vendors use this procedure. It has been implemented in industrial

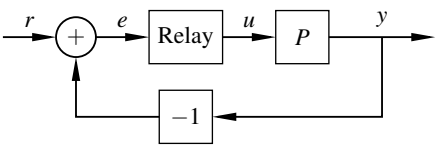


Figure 1. Setup for the relay feedback experiment, where r is the reference value, e is the control error, u is the relay output and y is the process output.

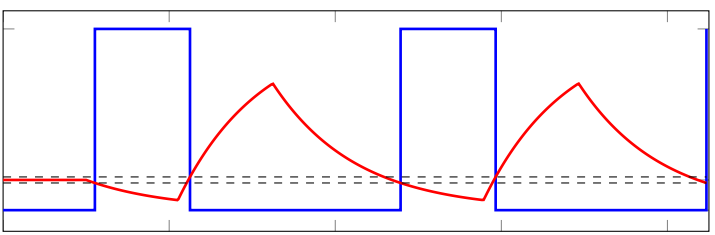


Figure 2. Outputs from a relay feedback experiment. The relay output (blue) switches between its two values every time the process output (red) leaves the hysteresis band (dashed black), causing the process to oscillate. If the amplitudes of the relay function are different, as in this figure, the relay is said to be asymmetric.

control systems such as the ABB ECA600, the ABB 800 XA and it is also the base of the autotuner feature in e.g. Emerson Delta V. Even though the knowledge about PID control has been improved and the available computing power of control systems has increased dramatically since the 1980's, it does not seem to have affected available industrial autotuners much. In academic literature many modifications and improvements have been suggested to the relay autotuner. For instance [Luyben, 1987; Li et al., 1991; Friman and Waller, 1997] have modified the autotuner to find first- or second-order models with time delay of the process instead of a single frequency point. The excitation of the process has also been improved by the usage of asymmetric relay functions in e.g. [Shen et al., 1996; Kaya and Atherton, 2001]. A review of the current state of process modeling from relay experiments can be found in [Liu et al., 2013]. However, these improvements do not seem to have made their way out to the industrial products.

The question we aim to answer in this paper is whether this is since the old autotuners are performing well-enough, or if they could actually be significantly improved by including recent scientific development. In order to do that, we compare two new autotuners developed by the authors, to the ABB ECA600 [ABB, 1999] containing the traditional autotuner [Åström and Hägglund, 1984] and the more recent autotuning algorithm Accutune III™ provided in the Honeywell UDC3200 [Honeywell, 2012]. Since the autotuners are mainly used in industrial settings, a simulation study would not cover typical problems. Consequently the comparisons will be performed on laboratory processes that feature many of the issues encountered in practice like noise, non-linearities, disturbances, low converter resolutions etc.

Even if the autotuning possibility has been available in most control systems for some decades it is not always used, resulting in unnecessarily poor control performances in many systems [Ender, 1993; Desborough and Miller, 2002; Kano and Ogawa, 2010]. Reasons for this may be that the users are either not aware of the feature or do not feel confident in using it. Therefore it is important to ensure that the autotuners, apart from giving satisfactory results, are easy to understand and use also for non-experienced users. In this study we will therefore do as few manual interactions as possible with the autotuner settings, and will not assume any knowledge of neither the process nor control theory.

2. The Study

This study compares and evaluates four different autotuners, described in Section 3, on three processes that are described in Section 4. The autotuners are evaluated on their experiment duration, their user-friendliness, and on the performance of the obtained controllers. Since the industrial autotuners do not provide the user with any models that they base the controller designs on, no comparison can be made between estimated models.

Three experiments are run by each autotuner, and a representative controller

setting is chosen for comparison to the other autotuners. The multiple experiments are performed in order to reduce the risk of disturbances affecting the result of one of the autotuners causing an unfair comparison. All obtained controller parameters are presented in Section 5, along with the performance experiments.

The most important aim for a controller in process industry is to be able to handle load disturbances, and the performance tests in this comparison are therefore focused on load disturbance attenuation. The disturbances were designed and controlled to ensure that identical disturbances affected each of the controllers on the specified process.

3. The Autotuners

The four autotuners used in this study are presented in this section. As stated in the introduction the aim is to compare industrial standard autotuners to recent developments in academic literature. The selection of industrial autotuners include the ABB ECA600 further described in Section 3.1 and the Honeywell UDC3200 described in Section 3.2. A motivation to this choice is that we wanted one autotuner containing an implementation of the original procedure from [Åström and Hägglund, 1984], which is still the most common one, as well as one more recent version of a relay autotuner available on the market today. By this choice we want to find out what improvements have been made in the industrial controllers during the last 30 years. There are of course other brands and procedures that could have been chosen, as well as completely different autotuning principles based on step responses or other open-loop experiments. We did, however, restrict this study to only contain relay autotuners.

The academic autotuners we selected are implementations of two versions developed by the authors. The τ-tuner, described further in Section 3.3, is procedure-wise very similar to the original autotuner, but has some modifications to obtain better models. The NOMAD autotuner, described in Section 3.4, utilizes more data and requires more computational power, but allows for shorter experiment times. We could have chosen to include many other academic autotuners in this study, e.g. the ones presented in [De Keyser et al., 2012; Chidambaram and Sathe, 2014; Yu, 2006]. Since we did not have any implementations of these autotuners, and since we wanted to find out how our proposed autotuners compared to the industry standard, we decided to restrict ourselves to this selection.

3.1 ABB ECA600

The ABB ECA600 controller, in this paper referred to as *ECA*, is shown in Figure 3. The operator's manual [ABB, 1999] describes it as *"... a dual loop controller with advanced control functions. In addition the ECA600 has comprehensive logical and arithmetical data processing facilities. Its five analog inputs, three analog outputs,*

Figure 3. The ABB ECA600 controller.

four digital inputs and six digital outputs can be used to solve almost any process control problem."

The built-in autotuner is based on [Åström and Hägglund, 1984] and provides PID parameters for a controller on serial form immediately from the experiment. The autotuner is accessible after enabling it in the configuration menu. Before starting the tuning the user has to make sure that the process value is in steady state close to the setpoint, otherwise the tuning may fail. The autotuner starts by measuring the noise level for 5 s to set an appropriate hysteresis level. The relay amplitudes are restricted to be less than 10 % of the control interval by default. If the autotuner notices that the process variable deviations are too large (or too small) during the experiment it will adjust the relay amplitudes if the restrictions allow it. The PID controller includes a first-order filter for the derivative part of the controller with a filter time constant $T_f = T_d/8$.

There is an option to tell the autotuner whether the controller dynamics should be `Normal`, `DeadTime`, `PI` or `pPI`. Since this study focuses on how an inexperienced user would be able to use the autotuners, we do not assume to have this kind of knowledge and hence always use the default `Normal`.

3.2 Honeywell UDC3200

To compare with a more recent controller we chose the Honeywell UDC3200 (referred to as *Honeywell*) with the autotuning feature *Accutune III*™. According to its manual [Honeywell, 2012] it is *"... an ideal controller for regulating temperature and other process variables in numerous heating and cooling applications, as well as in metal working, food, pharmaceuticals, semiconductor, testing and environmental work"*. In spite of being mainly considered as a temperature controller, the manual [Honeywell, 2012] also claims that *"This standard feature* [read: Accutune III™] *provides a truly plug and play tuning algorithm, which will, at the touch of a button or through a digital input, accurately identify and tune any process including those with deadtime and integrating processes"*

The tuning can be started whenever `TUNE` has been enabled in the setup menu, but to start the autotuner the controller must be switched to automatic mode. A choice between `SLOW` and `FAST` tuning can be made. This does not affect the experiment, but calculates the controller parameters differently. Since we do not assume to know anything about the process we used the `SLOW` option for all experiments. The autotuner switches the relay output two full cycles between the maximum and minimum control output levels, independent of the setpoint value. PID parameters for a controller on serial form are obtained from the experiment and immediately used. As far as we know, this autotuner does not design any filter for the controller, and the only filter used in our experiments is a noise filter with the default time constant value 1 s.

3.3 τ-tuner

The autotuner described in [Berner et al., 2016b; Berner et al., 2016a] is here denoted as the τ-tuner. It uses an asymmetric relay function with adjustable amplitudes and runs until limit cycle convergence is reached. From the relay oscillations it uses the half-period times and the integral of the process output over one cycle, to calculate an integrating or first-order model with time delay of the process. It then uses the AMIGO [Åström and Hägglund, 2006] tuning rules to obtain parameters for a PI or PID controller on parallel form. It uses the normalized time delay, τ, to select model and controller structure. The normalized time delay is defined as

$$\tau = \frac{L}{L+T}, \tag{1}$$

where L is the deadtime and T is the time constant of the process dynamics. More details about how the autotuner utilizes τ can be found in [Berner et al., 2016b]. The τ-tuner is simple to implement and use, but as was stated in [Berner et al., 2016a] it can be quite sensitive to quantization, non-linearities and non-stationary starting conditions. The autotuner setup in these experiments uses a restriction on the large relay amplitude to be maximum 10% of the available control range, it uses 10 s in the beginning of the experiment to measure noise and decide on a hysteresis level, but also has a minimum hysteresis level of 0.5% of the process output range.

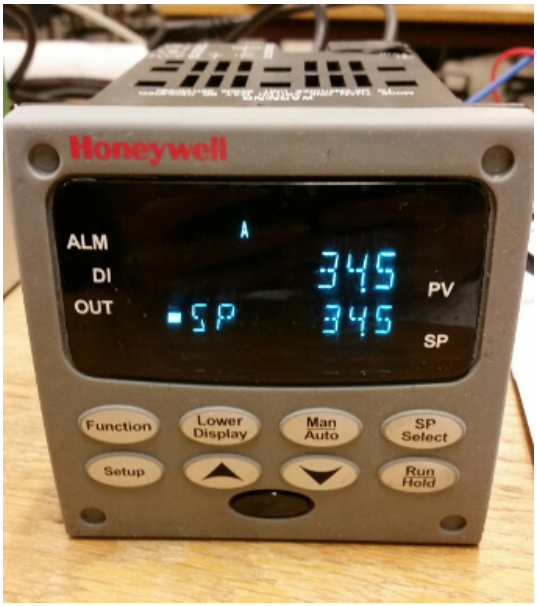


Figure 4. The Honeywell UDC3200 controller.

In the papers [Berner et al., 2016b; Berner et al., 2016a], no filter was designed for the controller. Since we need some roll-off for high frequencies to reduce the impact of noise, we let the autotuner design a second order filter on the form

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

The filter time constant T_f is chosen as

$$T_f = \frac{1}{5\omega_{180}}, \tag{3}$$

where ω_{180} is the frequency where the estimated process model has a phase lag of 180°. The filter $F(s)$ is used on the entire controller, not only the derivative part.

The model, filter, controller parameters and some additional experiment data are available to the user of this autotuner.

3.4 NOMAD-autotuner

The experiment and modeling part of the *Noise-robust Optimization-based Modeling And Design Autotuner* (NOMAD) is described in [Berner and Soltesz, 2017]. This autotuner performs the same experiment as the τ-tuner, with two exceptions. The first exception is that it does not have to be initiated in steady-state, since it estimates

the initial conditions. The second exception is that it does not wait for convergence of the limit cycle oscillations but instead stops the experiment after three relay switches. It then uses a gradient-descent algorithm to find first-order or second-order time-delayed models from the entire data set. It chooses which model to use by the Akaike Information Criteria [Akaike, 1974]. The usage of the entire data set distinguishes the NOMAD-autotuner from the ECA and τ-tuner, that only use certain time intervals, amplitudes etc. to get their models. It can be argued that the need of less data is a benefit of these other methods, but it is also what causes their need to wait for limit-cycle convergence as well as making them more sensitive to noise.

No controller tuning method was specified for the method in [Berner and Soltesz, 2017] so for this paper we added that to get a complete autotuner. The filter design was chosen the same way as for the τ-tuner, hence described by (2) and (3). The controller tuning chosen is the convex-concave optimization method described in [Hast et al., 2013]. Here the integrated error

$$IE = \int_0^\infty e(t)dt \tag{4}$$

from a step load disturbance on the process input, is minimized with constraints on the maximum values of the sensitivity function

$$S(s) = \frac{1}{1 + P(s)C(s)}, \tag{5}$$

and complementary sensitivity function

$$T(s) = \frac{P(s)C(s)}{1 + P(s)C(s)}. \tag{6}$$

The process $P(s)$ that is entered to the optimization program is the filtered process, since the controller design and the filter design are connected. The PID parameters obtained are for a controller on parallel form.

Since both the modeling and controller design are optimization-based, this autotuner requires a lot more computations than the original autotuner. However, the calculations performed are rather cheap, and with an efficient implementation and the computing power available today they can be made in the order of seconds.

4. Processes

To evaluate the autotuners on different types of dynamics we chose three processes with different characteristics. Two lag-dominated processes, where one has fast dynamics and one is slow, as well as one process that is delay-dominated. This

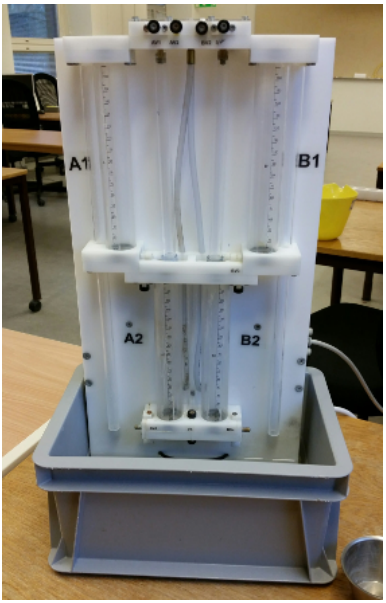


Figure 5. The quadruple tank used for level control experiments on the lower tank and the delayed upper tank.

information about the process dynamics was only used to choose suitable processes and was assumed to be unknown throughout the tuning experiments. The process hardware consisted of a quadruple tank and a batch tank available at the control laboratory at Lund University. The input and output signals of the processes were normalized to $u \in [0, 10]$ and $y \in [0, 10]$. The processes will be described further in their respective subsections.

4.1 The quadruple tank

The quadruple tank shown in Figure 5 is a version of the one described in [Johansson, 2000]. The tank can be used as both a single-input single-output system and as a multivariable system. For the experiments in this paper only one side of the quadruple tank was used, in order to get the single-input single-output configuration. This system consists of a pump that pumps water into the upper tank, which then flows through down to the lower tank. For the first experiment, called *Level control*, the measured process variable is the water level of the lower tank. The dynamics can be described by a second-order lag-dominant system with time constants T_1 and T_2, and an average residence time $T_{ar} = T_1 + T_2 \approx 20$ s.

The quadruple tank was also used for the delay-dominant process. For the *Delayed tank* experiment, the measured variable was the water level of the upper tank. The dynamics of this process can be described by a first-order system with a time

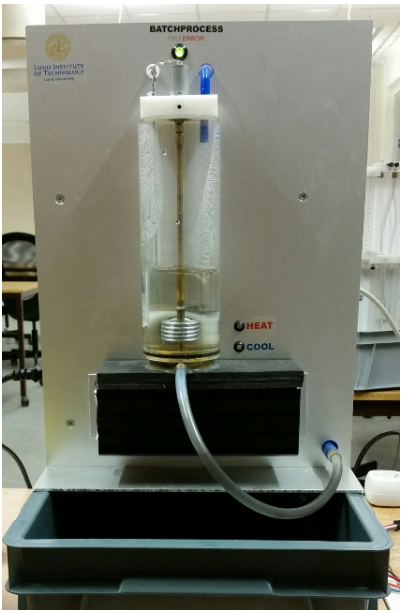


Figure 6. The batch tank used for temperature experiments.

constant $T \approx 10$ s. To get it delay-dominated a time delay of 20 s was added to the control signal.

4.2 The batch tank

The batch tank is shown in Figure 6. It consists of one inflow pump, one outflow pump, a heater, a cooler, and an agitator. Measurements of the water level and temperature are available. For the experiment on this process, referred to as *Temperature control*, we used a fix water level, with both pumps turned off. The agitator was on all the time and a split-range controller was used to control the heater and cooler. The control range of 0-10 was split up so that $u = 0 - 5$ corresponded to Cooler = 100% – 0% and $u = 5 - 10$ corresponded to Heater = 0% – 30%. The restriction on the heater was made to balance the heating and cooling capacities of the process.

5. Experiments and Results

The results from the three processes are illustrated by plots of the relay experiments as well as the obtained controller performance for load disturbances acting on the process inputs. Before the experiments were started the processes were brought to steady-state at the desired setpoint levels. This startup phase has been discarded in order to only show the actual experiments.

For each process three experiments were made with each autotuner to see how consistent they were. Obtained PID controller parameters for the different experiments are listed in Table 1-3. Since both the parametrization and the filter design differ between the controllers, a comparison of the bode plots of the resulting (filtered) controllers are shown for each process. From these results a representative set of parameters was chosen and used for the controller performance experiments. The performance experiments are shown, and the integrated absolute error

$$IAE = \int_0^{t_{\text{end}}} |r(t) - y(t)| dt, \tag{7}$$

for each performance experiment, where t_{end} is the experiment duration, is listed along with the controller parameters in the respective table.

5.1 Level control of the lower tank

A representative relay experiment for each of the autotuners is shown in Figure 8. Note the differences in experiment duration and signal deviations. This is a rather fast process, hence the short experiment duration. The resulting controller parameters are listed in Table 1 and Bode plots of the (filtered) controllers are shown in Figure 7. It can be seen that ECA, Honeywell and the τ-tuner are very consistent, while NOMAD is varying a bit more due to differences in the estimated models. However, the performance was still similar between the obtained NOMAD controllers.

To evaluate the controller performance two step load disturbances, of equal magnitude but opposite sign, were added to the input of the process. This was done by opening a valve at $t = 100$ s, decreasing the inflow to the upper tank with 50 %, and then closing that valve again at $t = 300$ s so that all the pumped water once again entered the tank. The performance of the obtained controllers are shown in Figure 9.

Two versions of the τ-tuner is shown in this plot. As was described in [Berner et al., 2016b] this autotuner makes some decisions based on the estimated value of the normalized time delay τ. In this case the estimated value was low, $\tau = 0.07$, and the process was wrongly classified as an integrating model with time delay. The dashed curve shows what the controller performance would have been if the process would instead have been classified as a first-order model with time delay, and the controller tuning would have been based on that. The results indicate that the threshold value for this classification should be reconsidered.

The results show that the NOMAD responds much faster and outperforms the others. The ECA controller comes in second place while the Honeywell controller has too little integral action and recovers very slowly. The τ-tuner is about as bad as Honeywell for the solid curve, but improves a bit for the dashed curve.

5.2 Temperature control of the batch tank

An illustrative relay experiment for each of the autotuners is shown in Figure 11. Note that the time scale for this process is in minutes since it is a much slower process than

Table 1. PID parameters and IAE values from the level control experiments. The chosen parameters for each autotuner are marked with bold text. Note that ECA and Honeywell are on serial form, while τ-tuner and NOMAD are on parallel form.

Autotuner	K	T_i	T_d	**IAE**
	2.06	12.8	3.20	
ECA	2.24	12.1	3.00	
	2.22	**12.2**	**3.00**	**145**
	1.52	24.6	6.00	
Honeywell	**1.41**	**24.0**	**6.00**	**453**
	1.40	24.0	6.00	
	1.52	29.7	1.86	
τ-tuner	1.44	30.7	1.92	
	1.47	**30.8**	**1.92**	**518**
	4.08	7.07	4.01	
NOMAD	12.8	5.33	3.19	
	5.77	**8.09**	**3.66**	**51**

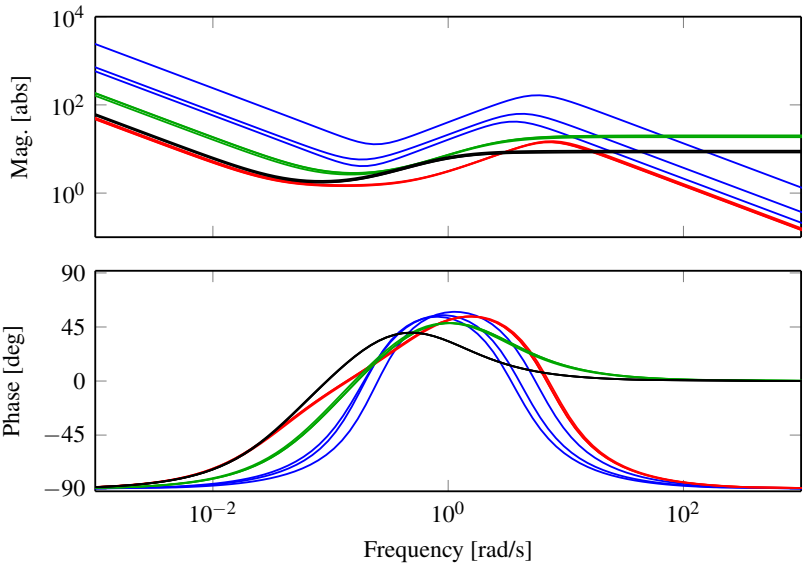


Figure 7. Bode plot of the obtained controllers for the level control of the lower tank. The different autotuners are ECA (green), Honeywell (black), τ-tuner (red) and NOMAD (blue).

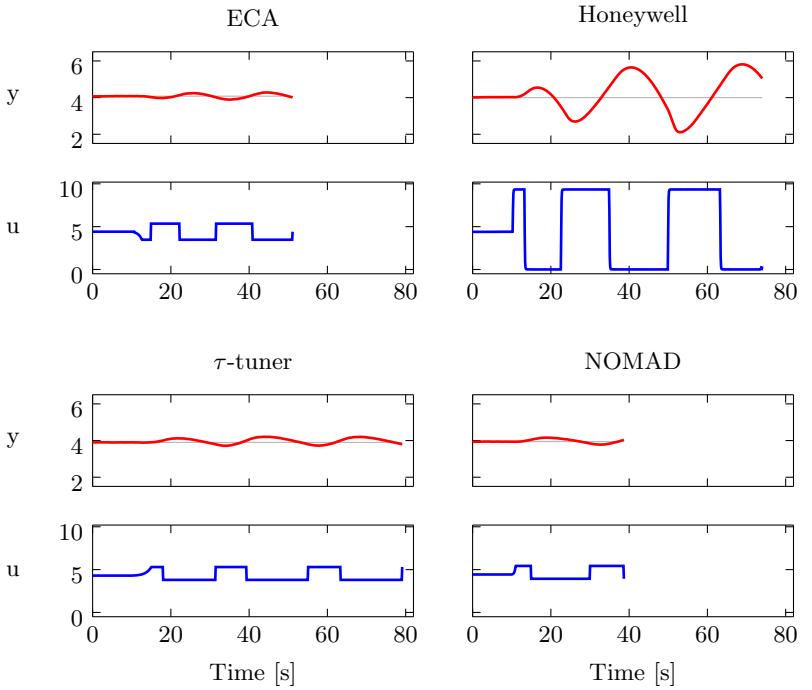


Figure 8. Relay experiments on the level control of the lower tank for the four autotuners. Note the short experiment time of the NOMAD autotuner and the large control and process deviations of the Honeywell autotuner.

the level control. From the experiment plots it is clear that the Honeywell controller is fastest for this process due to its larger relay amplitudes. The ECA controller is also fast, while the other two are slower. In this case it is not the number of relay switches that influences the experiment duration the most, but rather the time period of the oscillations. Why the different experiments get different time periods is discussed further in Section 6. It is also very clear from the experiment plots that the resolution of the AD-converters is low for this process, the quantization levels are clearly visible.

The controller parameters obtained from the experiments are listed in Table 2, and Bode plots of the controllers are shown in Figure 10. As can be seen the industrial controllers are both very consistent while the τ-tuner and NOMAD are varying more.

The controller performances for the respective autotuners are shown in Figure 12. The Honeywell controller performs well on this process, while the ECA is slow and almost not reaching steady-state within the 700 s shown in the plots. The NOMAD and τ-tuner performance are quite similar, but the high gain of the NOMAD

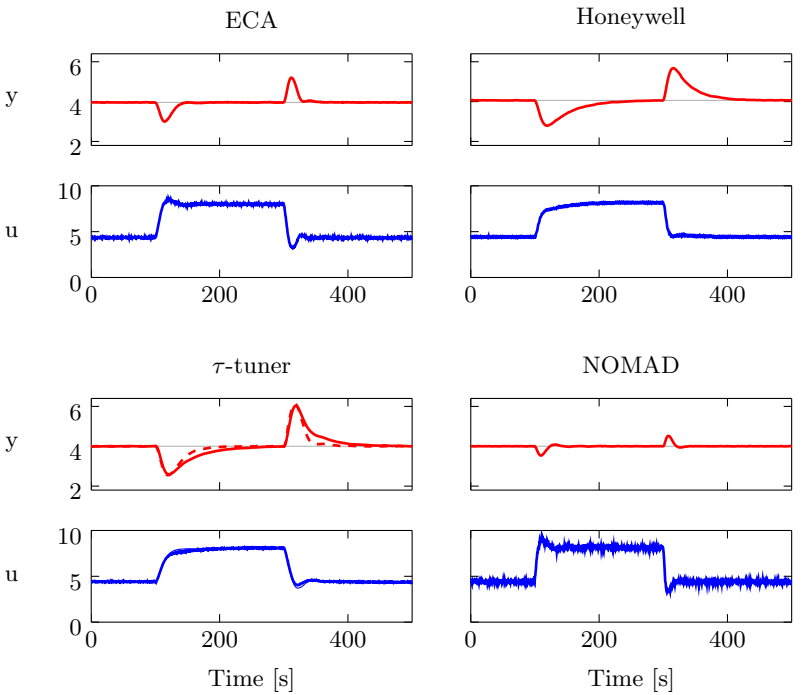


Figure 9. Controller performance for level control of the lower tank. At $t = 100$ s a step load disturbance was added to the process input and at $t = 300$ s that disturbance was removed. The τ-tuner has two versions in this plot, one controller based on an integrating model with time delay (solid), and one version tuned from a first-order model with time delay (dashed).

controller makes it saturate the control signal for a short while in the beginning of the disturbance. They both yield a small overshoot, but recover from it much faster than the ECA does. It can also be noted that even if the Honeywell controller does not have as high gain as the NOMAD and hence rises slightly slower and without an overshoot, the control signal is at least as varying as for the NOMAD controller due to the large derivative part that is affected by the quantization noise.

5.3 Level control of the delayed upper tank

The relay experiments for the delayed tank are shown in Figure 14. Worth noting in the figure is that both the ECA controller and the τ-tuner are adjusting their relay amplitudes during the experiment to decrease the process deviation. This increases the experiment time for them by a few half-periods. It is also clear from the figure that the Honeywell controller does not perform a good experiment on this

Table 2. PID parameters and IAE values for the temperature control experiments. The chosen parameters for each autotuner are marked with bold text. Note that ECA and Honeywell are on serial form, while τ-tuner and NOMAD are on parallel form.

Autotuner	K	T_i	T_d	**IAE**
ECA	10.10	174.7	0	
	10.36	**170.7**	**0**	**33**
	11.12	166.4	0	
Honeywell	33.60	111.6	28.2	
	33.86	**111.6**	**28.2**	**15**
	34.62	109.8	27.6	
τ-tuner	38.16	88.09	5.51	
	24.75	46.23	6.94	
	30.16	**118.8**	**7.43**	**18**
NOMAD	106.2	39.14	17.2	
	74.02	**45.17**	**20.0**	**13**
	61.39	48.01	20.5	

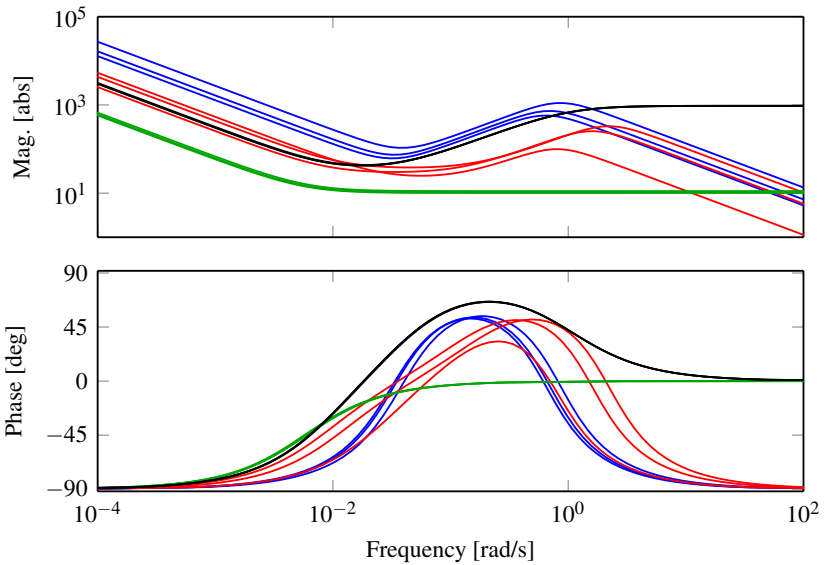


Figure 10. Bode plot of the obtained controllers for the temperature control of the batch tank. The different autotuners are ECA (green), Honeywell (black), τ-tuner (red) and NOMAD (blue).

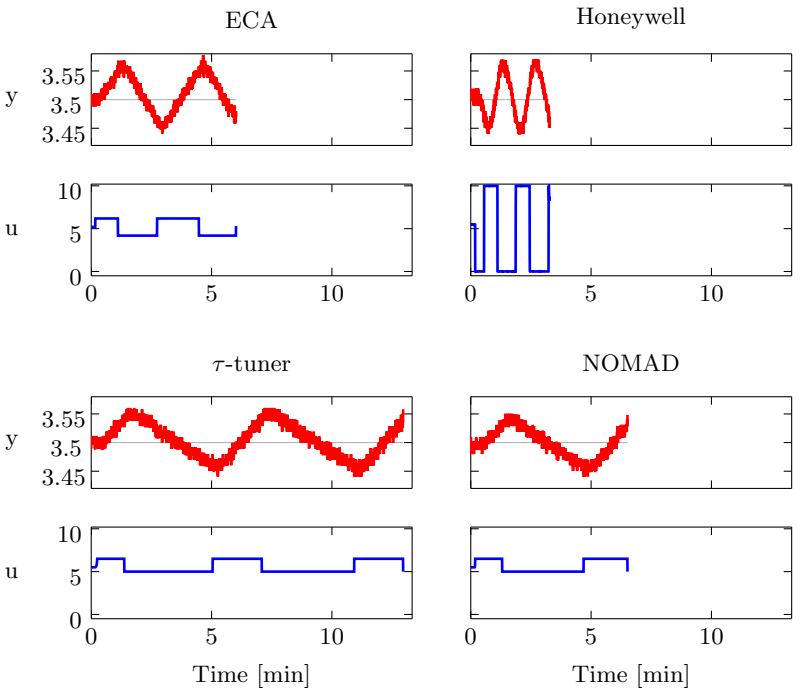


Figure 11. Relay experiments for the temperature control of the batch process. Note that the time scale is in minutes for this process. Also note the different experiment duration that is a consequence of the different oscillation periods, this will be discussed further in Section 6.

process. Its large relay amplitudes make the tank overflow and then become empty at every second relay switch. This makes the result from the Honeywell controller very unreliable for this process, and only one experiment was performed, but the resulting controller parameters will still be used and evaluated for comparison. The obtained controller parameters are listed in Table 3 and Bode plots of the controllers are shown in Figure 13. From the parameters it is seen that the τ-tuner chooses a PI controller since it classifies the process as delay-dominated, see [Berner et al., 2016b] for details on this choice.

The controller performances are shown in Figure 15. A step load disturbance was introduced at $t = 100$ s by adding an additional constant flow of water to the tank. The additional flow was removed again at $t = 800$ s. The NOMAD and τ-tuner are showing very good control results for this process, while the ECA is really slow and Honeywell is both oscillating and slow. Comparing the IAE values gives a different message, since the IAE for the Honeywell controller is slightly smaller than for the τ-tuner. This implies that IAE should be combined with restrictions on robustness.

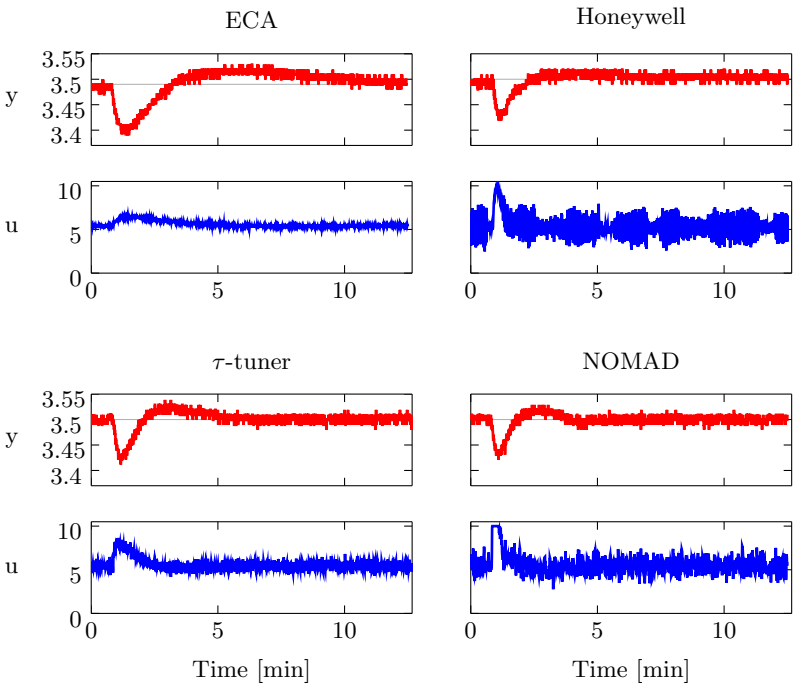


Figure 12. Controller performance for the temperature control of the batch tank. At time 50 s a load disturbance was introduced by changing 20 ml of the heated water to ambient tempered water in the tank.

6. Discussion

The results show that ECA is performing well for the level control of the lower tank, but is slow for the other two processes. The autotuner is easy to use, but the user has to ensure that the process is in steady-state before starting the experiment in order to get good results. If the process value is far from the setpoint when starting the experiment, the system will issue a warning, but apart from that it is the user's responsibility to ensure stationarity.

The Accutune III™ in the Honeywell controller works very well for the temperature control process, but yields very slow controllers for the other processes. Its large experiment amplitudes are a problem, especially for the time-delayed process, where they cause the process value to saturate at every switch, resulting in unreliable results and causing operational problems. The Honeywell controller is branded as a temperature controller so it is not surprising that it gives the best results for the temperature control process. However, since it claims to give good results for any

Table 3. PID parameters and IAE values from the experiments on the upper tank with delay. The chosen parameters for each autotuner are marked with bold text. Note that ECA and Honeywell are on serial form, while τ-tuner and NOMAD are on parallel form. Since the Honeywell controller was not able to do the experiment without overflowing the tank and emptying it, only one experiment was performed on it.

Autotuner	K	T_i	T_d	**IAE**
ECA	0.23	61.0	15.2	
	0.29	48.0	12.0	
	0.28	**57.4**	**14.3**	**379**
Honeywell	**0.49**	**36.0**	**9.00**	**218**
τ-tuner	0.11	12.3	0	
	0.11	**11.8**	**0**	**224**
	0.32	8.88	0	
NOMAD	0.20	14.7	8.24	
	0.21	14.6	8.44	
	0.21	**14.7**	**8.32**	**152**

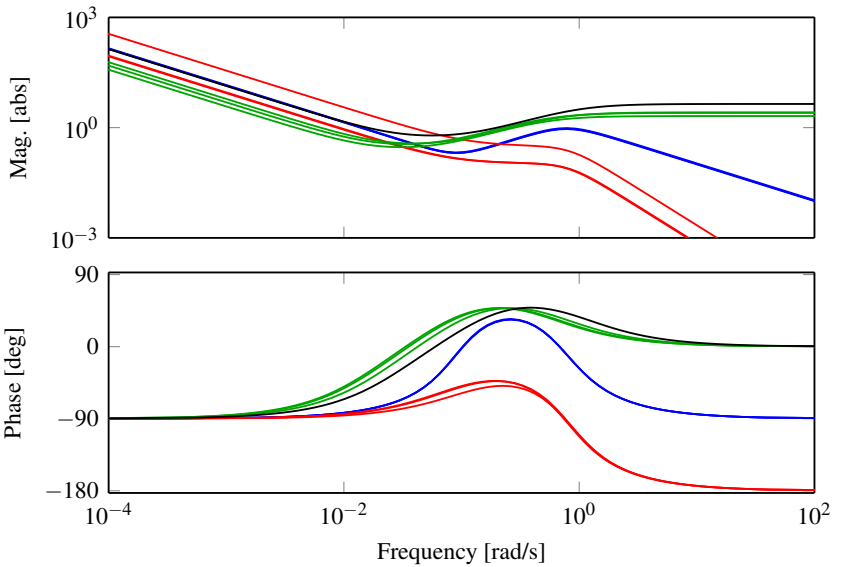


Figure 13. Bode plot of the obtained controllers for the level control of the delayed upper tank. The different autotuners are ECA (green), Honeywell (black), τ-tuner (red) and NOMAD (blue).

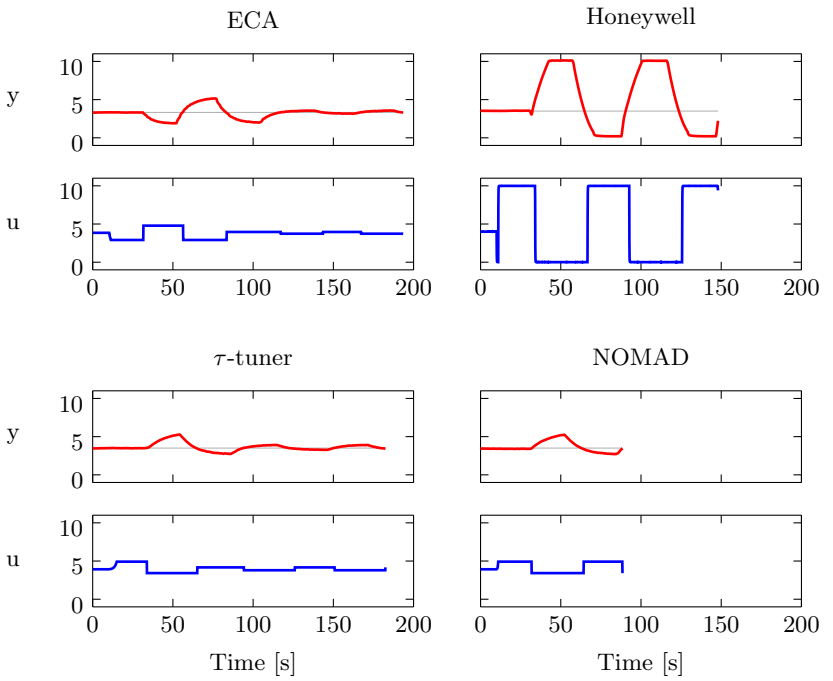


Figure 14. Relay experiments for the level control of the delayed tank. The Honeywell controller is overflowing and emptying the tank at every switch, which make its experiment results unreliable.

process including those with deadtimes the performance for that process is rather unsatisfactory. The autotuner feature is quite easy to use, and it is not as sensitive to starting in steady-state as the ECA since the experiment is the same no matter if the process value is at the setpoint or not. Since the relay always uses the maximum amplitudes the experiment becomes more or less asymmetric depending on the chosen setpoint. The need to put it in automatic before starting the experiment seems a bit strange since it does not have any appropriate controller parameters for the process yet. However, since the user is supposed to start the tuning procedure right after and it does not have to be steady it does not matter that much.

The τ-tuner has an experiment very similar to ECA, except for the asymmetry in the relay. They sometimes differ a few relay switches before convergence is reached, but both the process deviations, amplitude restrictions, and amplitude adjustments are similar. The way of finding models and controller parameters are different though. The results from the τ-tuner are slightly worse than ECA for the level control of the lower tank, (especially for the version where the process is classified as integrating),

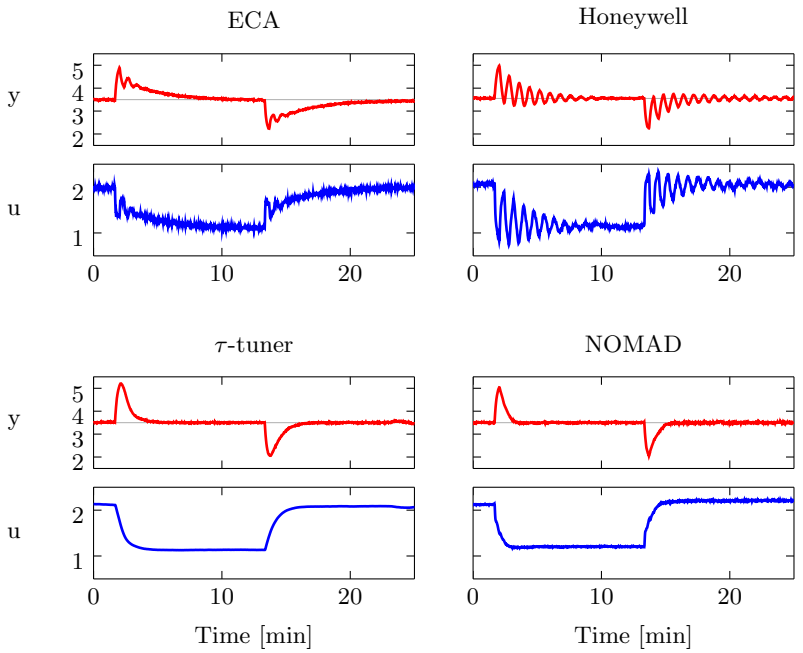


Figure 15. Controller performance for the delayed tank process. A step load disturbance is added to the input at time 100 s and then removed at time 800 s.

but better than Honeywell. For the temperature control the τ-tuner is worse than Honeywell but better than ECA, and for the time-delayed system the τ-tuner is much better than both ECA and Honeywell. Overall it can be argued that the τ-tuner gives a rather consistent performance. It may not be the best, but it gives acceptable controllers for all the tried processes. The τ-tuner, however, suffers the same problem as ECA by requiring the system to be in steady-state before the experiment starts. This is even more crucial for the τ-tuner since the asymmetry level of the relay will not be correct if the output is drifting, causing erroneous model estimations. This may be the reason why the controller parameters sometimes differ between consecutive experiments for the τ-tuner.

The NOMAD autotuner gives best controller performance for all tested processes. The increase in computing to get more accurate models allows for more aggressive controller designs, which is clearly seen for instance in Figure 9. Apart from the good performance, the short duration and low process deviations of the experiment are also great benefits of the method. The experiment can also be started with non-stationary initial conditions, so the problem encountered in ECA and the τ-tuner to ensure steady state before starting the autotuner is removed. The benefit

of short experiment duration is especially useful since it reduces the risk of disturbances entering during the experiment, which is one of the largest risks of failure for the relay autotuners. The experiment is similar to those of both ECA and the τ-tuner, but with approximately half the number of relay switches.

As can be seen in Figure 11, the number of relay switches are not always what is most significant for the experiment duration. If there would have been no hysteresis in the relays, they would all oscillate with the same frequency. With hysteresis the relay amplitudes and hysteresis amplitude will have a large impact on the oscillation period for certain processes. This could be understood by looking at the Nyquist curve in Figure 16. The approximative describing function method tells that a system under symmetric relay feedback will oscillate with a frequency decided by the intersection of the process' Nyquist curve with the horizontal line with imaginary part

$$Im = -\frac{\pi h}{4d}, \tag{8}$$

where h is the hysteresis of the relay and d is the relay amplitude. The hysteresis levels seem to be quite similar between the different autotuners, but since the relay amplitude is about five times larger for the Honeywell controller than the others it will oscillate with a much higher frequency and shorter period time. The ECA controller has a slightly higher amplitude than the τ-tuner and the NOMAD since it allows 10 % both up and down, while they have 10 % as their large amplitude. Hence ECA gets a faster oscillation than they do. To exemplify, consider the τ-tuner. The average relay amplitude is 7.5 % and the hysteresis 0.5 %. This would give an intersection with imaginary part $Im = -0.5\pi/(4 \cdot 7.5) \approx 0.05$. Looking in Figure 16 this implies an oscillation period a bit larger than 300 s, which seem to be a reasonable approximation. For processes that do not change so drastically in frequency in this area the difference between oscillation periods are barely noticeable and for the other experiments in this paper the oscillation periods are therefore more or less the same.

A fundamental benefit of the τ-tuner and NOMAD compared to the others is the provision of explicit process models. That makes it possible to change tuning method and filter design without having to perform a new experiment. However, that is mainly a benefit for experienced users or developers. Productified autotuners need to be user-friendly in order to be used. Decisions should be made automatically, maybe allowing, but not requiring, user inputs. Ideally, good results should be provided every time without any manual interaction. The τ-tuner and NOMAD-tuner aim for this by making both model and controller selections automatically.

Another feature we propose for the autotuners is that they provide a suitable filter in addition to the controller parameters, since they are strongly connected. ECA does that, and filter designs were included in the implementations of the τ-tuner and NOMAD. The autotuner could also propose set-point weighting constants, anti-windup tracking constants and other parameters connected to the PID controller.

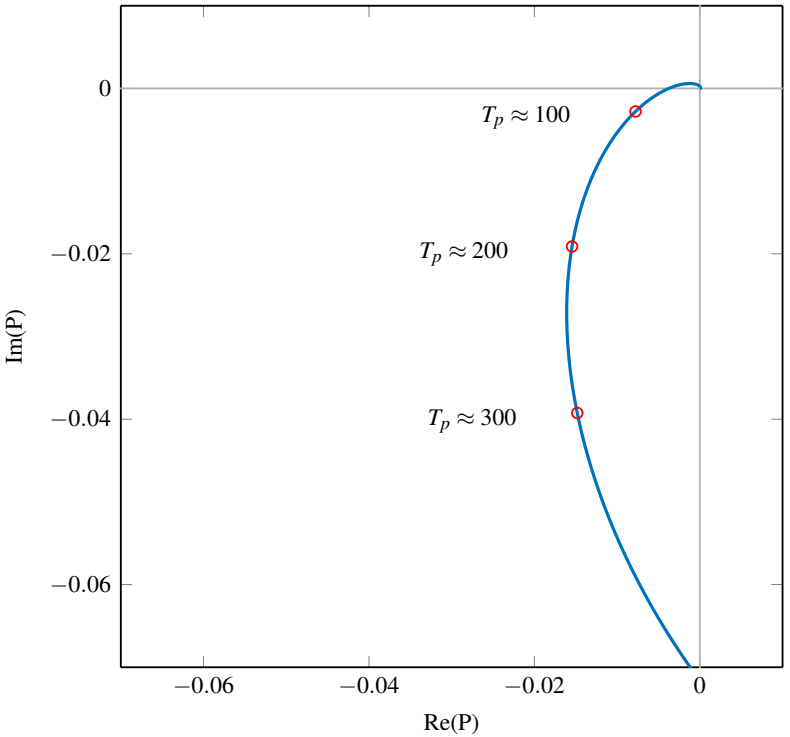


Figure 16. Nyquist plot of the model obtained from the NOMAD autotuner for the temperature control process. Depending on the ratio between relay amplitude and hysteresis, different points on the curve will be intersected by the relay describing function, causing different oscillation periods. Frequencies corresponding to oscillation periods T_p={100 s, 200 s, 300 s} are marked in the figure.

7. Conclusions and Future Work

It is time to update the industrial standard autotuners from the 1980's technology to the 21st century. The enormous increase in computing power and data storage provides the possibility to use much more data than just a handful of values from an experiment. By using optimization methods on the entire data set, the experiments can be made shorter and still provide more accurate models, resulting in controllers with better performance. Even though the ECA gives functioning controllers for all processes evaluated in this study, the results show that the NOMAD improves the performance significantly. The NOMAD autotuner also decreases the two main risks of failure for the ECA controller. The first risk is that the process is not in steady-state when the experiment is started. This is handled by the optimization method in NOMAD, that allows non-stationary initial conditions in its estimations. The

second main risk is that some disturbance enters the system while the experiment is ongoing, deteriorating the obtained controller. This risk is reduced significantly by the much shorter experiments used in the NOMAD. The short experiments are also very beneficial for the availability of the control loop.

The benefits of the NOMAD autotuner clearly motivates that procedures like it should be productified. The product should have good and fast implementations of the optimization algorithms, as well as a clear user-friendliness in mind. Because it does not matter how good the autotuner is if it is not used!

Acknowledgment

The authors are members of the LCCC Linnaeus Center and the ELLIIT Excellence Center at Lund University.

References

ABB (1999). *ECA06/60/600 EMA60 Operator's Manual*. Doc. number 493-0735-11 (4-1). ABB Satt AB.

Akaike, H. (1974). "A new look at the statistical model identification". *IEEE transactions on automatic control* **19**:6, pp. 716–723.

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.

Berner, J., T. Hägglund, and K. J. Åström (2016a). "Asymmetric relay autotuning–Practical features for industrial use". *Control Engineering Practice* **54**, pp. 231–245.

Berner, J., T. Hägglund, and K. J. Åström (2016b). "Improved Relay Autotuning using Normalized Time Delay". In: *2016 American Control Conference (ACC)*. IEEE, pp. 1869–1875.

Berner, J. and K. Soltesz (2017). "Short and robust experiments in relay autotuners". In: *2017 IEEE Conference on Emerging Technologies and Factory Automation (ETFA)*. Limassol, Cyprus.

Chidambaram, M. and V. Sathe (2014). *Relay Autotuning for Identification and Control*. Cambridge University Press.

De Keyser, R., O. L. Joita, and C. M. Ionescu (2012). "The next generation of relay-based PID autotuners (part 2): a simple relay-based PID autotuner with specified modulus margin". *IFAC Proceedings Volumes* **45**:3, pp. 128–133.

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

Ender, D. B. (1993). "Process control performance: not as good as you think". *Control Engineering* **40**:10, pp. 180–190.

Friman, M. and K. V. Waller (1997). "A Two-Channel Relay for Autotuning". *Industrial and Engineering Chemistry Research* **36**:7, pp. 2662–2671. DOI: 10.1021/ie970013u.

Hast, M., K. Åström, B. Bernhardsson, and S. Boyd (2013). "PID design by convex-concave optimization". In: *2013 European Control Conference (ECC)*. IEEE, pp. 4460–4465.

Honeywell (2012). *UDC3200 Universal Digital Controller Product Manual*. Doc. number 51-52-25-119 revision 5. Honeywell.

Johansson, K. H. (2000). "The quadruple-tank process: a multivariable laboratory process with an adjustable zero". *IEEE Transactions on control systems technology* **8**:3, pp. 456–465.

Kano, M. and M. Ogawa (2010). "The state of the art in chemical process control in Japan: Good practice and questionnaire survey". *Journal of Process Control* **20**:9, pp. 969–982.

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

Li, W., E. Eskinat, and W. L. Luyben (1991). "An improved autotune identification method". *Industrial and Engineering Chemistry Research* **30**:7, pp. 1530–1541. DOI: 10.1021/ie00055a019.

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

Luyben, W. L. (1987). "Derivation of transfer functions for highly nonlinear distillation columns". *Industrial and Engineering Chemistry Research* **26**:12, pp. 2490–2495. DOI: 10.1021/ie00072a017.

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

Yu, C.-C. (2006). *Autotuning of PID controllers: A relay feedback approach*. Springer Science & Business Media.

Paper V

Autotuner identification of TITO systems using a single relay feedback experiment

Josefin Berner Kristian Soltesz Tore Hägglund
Karl Johan Åström

Abstract

Relay autotuning has proven very successful for single-input single-output systems. This paper proposes an identification method for relay autotuning of systems with two inputs and two outputs (TITO systems). The combination of asymmetric relay feedback and output error identification admits short tuning time, without the need for limit cycle convergence. The method is successfully demonstrated on relevant system models, including the Wood-Berry distillation column.
Keywords: Multivariable autotuning, decentralized relay experiment, output error identification.

1. Introduction

Relay autotuning for single-input single-output (SISO) systems has been widely used since its introduction in the 1980's. The closed-loop identification automatically excites the system in the frequency interval relevant for PID control, and the amplitudes of the oscillations can be kept small as to not disturb the process more than necessary. The relay autotuner in [Åström and Hägglund, 1984] has been modified by e.g. [Luyben, 1987] where a first-order model with time delay (FOTD model) of the system was derived, by [Shen et al., 1996] where an asymmetric relay was used, and in more recent work as [Berner et al., 2016b] where the normalized time delay was used to classify the system during tuning and [Berner et al., 2016a] where parameter choices and other practical issues were discussed. A recent review of the advances in identification from relay experiments is given in [Liu et al., 2013]. Several books on relay autotuning, like [Yu, 2006; Chidambaram and Sathe, 2014], also give good overviews.

Even though much has been written about relay autotuning almost all of it considers SISO systems only. Since many industrial processes are of multi-input multi-output (MIMO) type, there is a need of finding multivariable models of them. These models can then be used to either tune a number of SISO PID controllers by picking a suitable input-output pairing or by decoupling the system. Or a multivariable PID controller could be tuned as in [Boyd et al., 2016]. The choice of tuning method is not the subject of this paper, but by obtaining a full transfer function matrix, many tuning options are possible. In this paper we will restrict ourselves to two-input two-output (TITO) systems. This is a common subclass of MIMO systems that shows up in many places both in literature and in industrial processes.

Relay autotuning of multivariable systems could be done in three different ways, [Wang et al., 1997]. The first way is to tune each loop independently while leaving the others in manual. This method does not take any of the cross-couplings in the system into account, and hence is not a good option for coupled systems. The second method is sequential tuning, where the first loop is tuned while the others are in manual, and then the next loop is tuned with the first one closed, and so on until all loops are tuned. With this method the controllers are tuned based on all the information up to that point, but the loops that are still open do not influence the behavior. Therefore the method is usually iterated a number of times with the loops closed, which results in a total of mk relay experiments for k iterations of an $m \times m$ process. The third option, which we will use in this paper, is decentralized relay feedback, where all the loops are tuned simultaneously. This is a completely closed-loop method where all cross-couplings will influence the result. In [Wang et al., 1997] m decentralized relay tests were needed in order to obtain the transfer functions. It was also needed to wait for convergence of the limit cycle oscillations. Another drawback of that method is the assumption that the systems are coupled strongly enough to oscillate with the same fundamental period. Attempts to find the conditions for when this happens has been made in e.g. [Loh and Vasnani, 1994], but to our knowledge no simple full

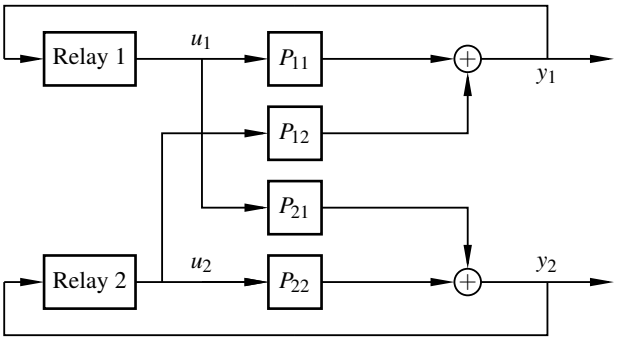


Figure 1. Setup of the decentralized relay experiment.

conditions have been published. In [Campestrini et al., 2006], problems with using the ultimate frequency as a tuning parameter for MIMO systems have been remarked upon. For MIMO systems there is not an ultimate point as in the SISO case, but rather an ultimate surface, and which point that will be found from the experiment depends on the settings. This makes simple methods like Ziegler-Nichols, based on the ultimate gain and frequency, inappropriate for the MIMO case.

What we want is a relay autotuner for MIMO systems that is fast, does not require numerous experiments, that works for both weakly and strongly coupled processes, and that does not rely on only an ultimate point to get the process models. The autotuner in this paper aims to identify the transfer function matrix for a TITO system from one single decentralized relay experiment. It does not need to wait for convergence, which makes it fast. The models are estimated by output error minimization, which does not require the loops to oscillate with the same frequency and does not use the ultimate frequency in any calculations. The method will be described and some examples demonstrating the potential of the method will be given. The controller tuning, and some practical issues of the experiments are left as future work.

2. Method

2.1 Experiment

The experimental setup is shown in Figure 1 and based on the decentralized relay feedback experiment proposed in [Wang et al., 1997]. Both loops are closed with relay feedback simultaneously. Even though it is a closed-loop experiment, the input-output noise correlation is negligable since the input signal is kept constant except at the switching instances of the relay. Hence, the system can be viewed as open-loop for identification purposes. The relays used are asymmetric, and implement most practical features from the one described in [Berner et al., 2016a]. To get as much

excitation as possible the asymmetry level γ (i.e., the ratio between the on and off amplitudes) is different for the two relays. In this paper the asymmetries were set to $\gamma = 1.5$ in relay 1 and $\gamma = 2$ in relay 2, but this choice should be investigated more in future work. The amplitudes are set automatically during the startup phase. The identification method that will be used in this paper does not require limit cycle convergence, hence the experiments can be made short. In this paper the experiments are stopped when both loops have undergone four relay switches. For the simulations in this paper, the control signal is set to its stationary level immediately when the experiment stops. If this is the best way of shutting down the experiment, and when to connect the new controller, needs to be investigated in future work.

2.2 Coupling level

As has been described in [Loh and Vasnani, 1994], there are three possible limit cycle behaviors of the decentralized relay feedback experiment of the TITO system. The separate behaviors depend on the strength of the cross-couplings in the process. An illustration of the possible outcomes is given by looking at the system

$$G_\epsilon(s) = \begin{pmatrix} \frac{1}{s+1}e^{-s} & \frac{\epsilon}{s+1}e^{-s} \\ \frac{\epsilon}{s+1}e^{-s} & \frac{1}{0.1s+1}e^{-s} \end{pmatrix} \tag{1}$$

with $\epsilon = \{0, 0.3, 0.8\}$ shown in Figure 2-4. For the purpose of this illustration the decentralized relay experiment has been run until convergence of the time periods of the limit cycles, or in the intermediately coupled case for 10 switches since it will not converge. If the coupling of the system is sufficiently strong, like in Figure 4, both loops will oscillate with the same frequency, the half-periods may, however, be separate. If the coupling is weak (or non-existent), as in Figure 2, the loops will oscillate with different frequencies like two separate SISO systems. If the coupling is intermediate, see Figure 3, the loops will have a more complex limit cycle with multiple relay switches within the fundamental period.

Most literature on decentralized relay autotuning like e.g. [Wang et al., 1997] assumes the strongly coupled case with equal fundamental periods. We would, however, like a method that can handle all coupling levels. The uncoupled system could just as well be treated as two SISO systems and use methods for that, but the possibility to handle it in the same framework as the others is desirable since the coupling level is not necessarily known on beforehand.

Figure 3 and Figure 4 show that the waveforms of the limit cycles can be quite complex, and to use simple equations from a few significant data points, as is usually done in the SISO autotuner, is not straightforward. Instead we use all data points to identify model parameters, as described in the following section.

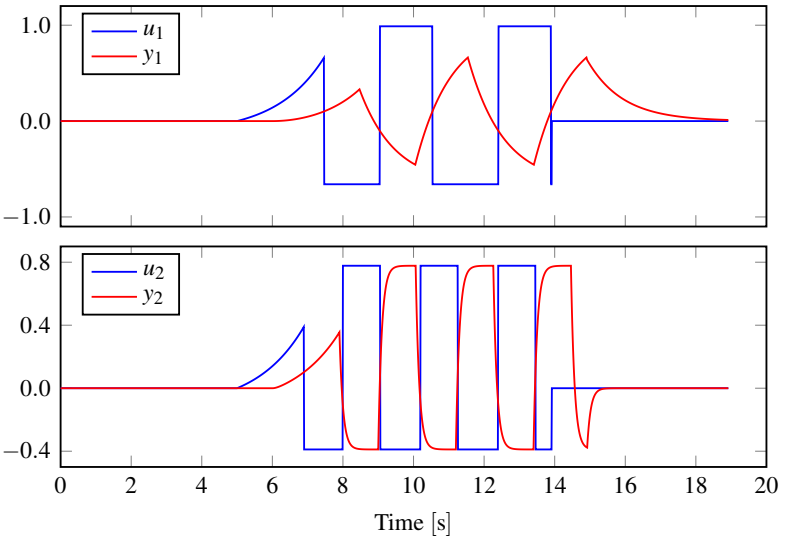


Figure 2. Decentralized relay experiment for the uncoupled system, $G_{\epsilon=0}$. The two loops converge to limit cycles with different periods $t_p = 3.26$ and $t_p = 2.19$.

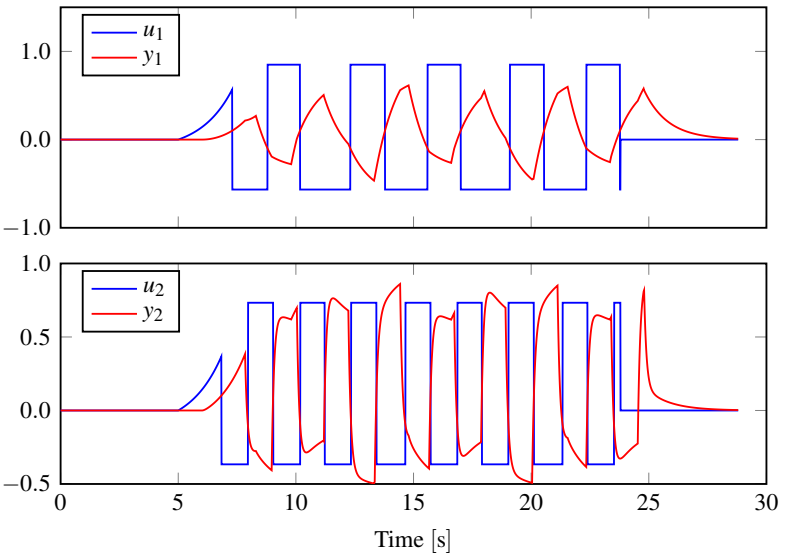


Figure 3. Decentralized relay experiment for the intermediately coupled system, $G_{\epsilon=0.3}$. The loops converge to limit cycles that contain multiple relay switches.

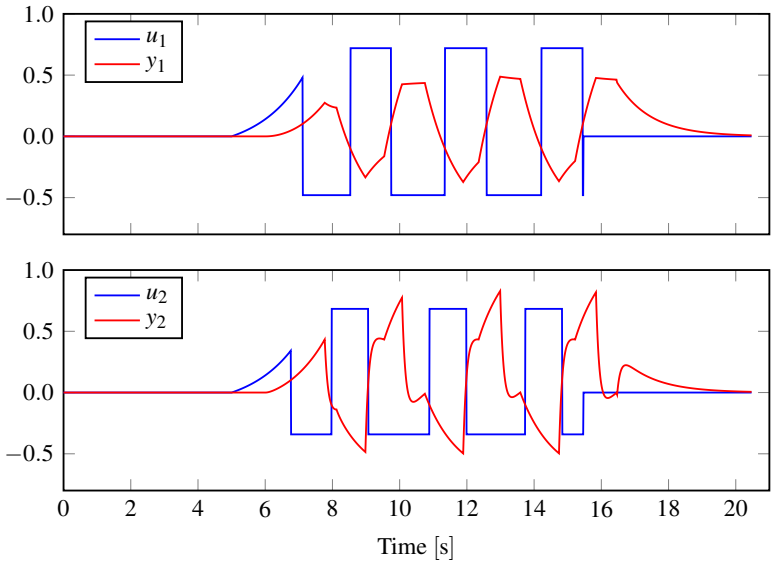


Figure 4. Decentralized relay experiment for the strongly coupled system, $G_{\epsilon=0.8}$. The loops converge to limit cycles with identical periods $t_p = 2.76$.

2.3 Identification method

The elements of the transfer function matrix in Figure 1 will be estimated on the form

$$P_{i,j}(s) = \frac{b_{i,j}}{s + a_{i,j}} e^{-sL_{i,j}}, \; i, j \in \{1, 2\}, \tag{2}$$

that is, first order systems with time delay (FOTD). The parameter vector defining each element (2) is $\theta_{i,j} = \begin{bmatrix} b_{i,j} & a_{i,j} & L_{i,j} \end{bmatrix}^\top$.

For our purposes the system can be decomposed into two parts $y_i = P_{i,1}u_1 + P_{i,2}u_2$, with parameter vectors $\theta_i = \begin{bmatrix} \theta_{i,1}^\top & \theta_{i,2}^\top \end{bmatrix}^\top$. Denoting by $\mathbf{u}_1, \mathbf{u}_2$, $\mathbf{y}_i$ and $\hat{\mathbf{y}}_i$, the sampled process inputs, output, and the model output, respectively, we pose the parameter identification problem as minimization of the $\mathcal{L}_2$ norm of the output error $\mathbf{e}_i = \mathbf{y}_i - \hat{\mathbf{y}}_i$

$$J(\theta_i) = \frac{t_s}{2} \mathbf{e}_i^\top \mathbf{e}_i, \tag{3}$$

where t_s is the sampling period of the signals. Since $\hat{\mathbf{y}}_i$ is not convex in θ_i, a local second-order method will be used. Its SISO counterpart for discrete time systems was described in [Åström and Bohlin, 1966], and it was adopted to relay identification of continuous time systems in [Soltesz et al., 2010]. In the SISO case, the essence of the method lies in constructing a continuous time state space system $\mathcal{S} : \{A, B, C, D\}$. The system $\mathcal{S}$ is parametrized in θ, and is designed to have a certain structure such

that its delayed output contains both the estimated output $\hat{\mathbf{y}}$ and its gradient $\nabla_\theta \hat{\mathbf{y}}$,

$$\text{Ш}_{t_s}\, C\mathbf{x}(t+L) = \begin{bmatrix} \hat{\mathbf{y}}^\top & \nabla_\theta \hat{\mathbf{y}}^\top \end{bmatrix}^\top,$$

where Ш_{t_s} denotes the sample operator of period t_s. The (un-delayed) output of $\mathcal{S}$ is cheap to compute by simulation of the discrete time system $\{\Phi, \Gamma, C, D\}$ obtained through zero-order hold sampling of $\mathcal{S}$, with period t_s. A subsequent shift by $\lfloor L/t_s \rfloor$ produces the desired output and associated gradient. For the FOTD SISO case, a minimal realization of $\mathcal{S}$ is given by

$$\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right] = \left[\begin{array}{cc|c} -a & 0 & 1 \\ b & -a & 0 \\ \hline b & 0 & 0 \\ 1 & 0 & 0 \\ 0 & -1 & 0 \\ ab & 0 & -b \end{array}\right]. \tag{4}$$

See [Soltesz et al., 2010] for the general order SISO case, and its derivation.

Use of (4) enables computation of the Jacobian

$$\nabla_\theta J = t_s \mathbf{e}^\top \nabla_\theta \hat{\mathbf{y}},$$

and an approximation of the associated Hessian

$$\Delta_\theta J \approx \nabla_\theta \hat{\mathbf{y}}^\top \nabla_\theta \hat{\mathbf{y}}.$$

The approximation consists in keeping quadratic terms $\partial^2/\partial^2\theta_i$, but discarding cross terms $\partial^2/(\partial\theta_i \partial\theta_j)$ – which are negligible unless $\mathbf{e}$ and $\nabla_\theta \hat{\mathbf{y}}$ are strongly correlated.

Using the Jacobian and Hessian information, the optimization problem can be solved (to a local minimum) by standard means. (The trust-region-reflective solver bundled with Matlab has been used here.) Each iteration only involves a simulation of $\mathcal{S}$ and a gradient descent step, making the identification method computationally cheap.

Extension to the TITO case is straightforward as the output $y_i = P_{i,1}u_1 + P_{i,2}u_2$ is the sum of outputs $y_{i,1} = P_{i,1}u_i$ and $y_{i,2} = P_{i,2}u_2$. Two instances of (4) are constructed for $y_{i,1}$ and $y_{i,2}$, respectively. From their outputs it is straightforward to assemble $\nabla_{\theta_i} \hat{\mathbf{y}}_i = \begin{bmatrix} \nabla_{\theta_{i,1}} \hat{\mathbf{y}}_i & \nabla_{\theta_{i,2}} \hat{\mathbf{y}}_i \end{bmatrix}$, which enables computation of the Jacobian and Hessian associated with (3).

It can be noted that the identification method is readily extendible to higher order systems, as put forth in [Soltesz et al., 2010]. It is also straightforward to apply it to transfer matrices of arbitrary input and output dimensionality. The ability of the proposed method to identify parameters under such conditions, depends on how well the identification input excites the dynamics, and will be investigated in future work.

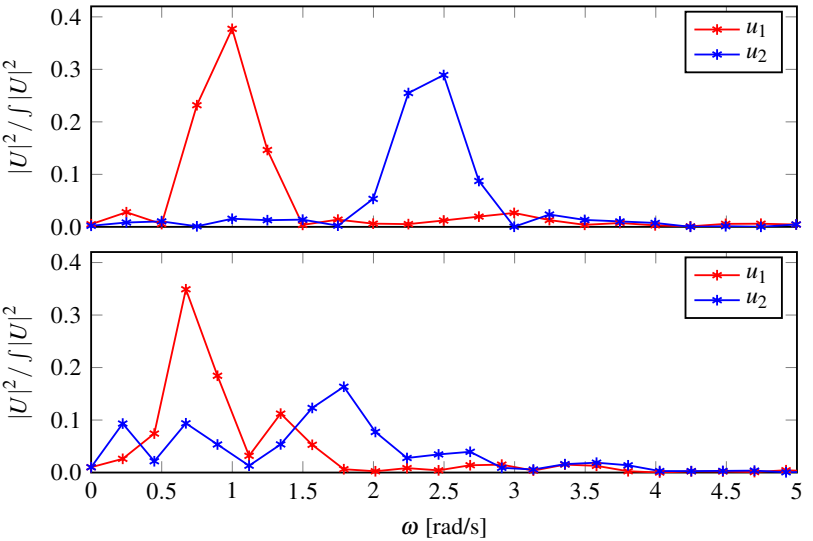


Figure 5. Power spectra of the relay signals in the decentralized relay experiment for the process $G_{\epsilon=0.3}$. The upper plot shows an experiment with symmetric relays. The lower plot shows an experiment with the asymmetry levels $\gamma_1 = 1.5$ and $\gamma_2 = 2$.

Excitation As was just stated, it is crucial to excite the process sufficiently to be able to find all unknown process parameters. In Figure 5 the power spectra of the input signals are shown for two decentralized relay feedback experiments. These specific experiments were done on $G_{\epsilon=0.3}$, as defined in (1), with symmetric relay functions in the upper plot and asymmetric relay functions in the lower plot. As can be seen the signal power is much more spread out for the asymmetric relay functions. The frequency plots of course depend on the processes. The large peaks in the frequency plots are from the main oscillation periods, so for strongly coupled systems, e.g. $G_{\epsilon=0.8}$, that oscillates with the same frequncy, the two curves will more or less be on top of each other.

Initialization The described identification method sometimes converges to a local minimum. Initialization is therefore important. Since the computations are cheap and fast, and this paper is a proof of concept rather than a complete algorithm, we chose to initialize in multiple points and then pick the best solution. The first attempts indicated that the time delay L is the most crucial parameter to have a good initial value of, therefore we start with L gridded between 0 and $L_{\max}$ that is here set to the time period of the oscillation. The initial values of a and b are set to 0. If the grid is very dense this methodology will require a lot of combinations which will make the overall computing time larger than necessary. Therefore a more clever way of initializing the system should be found in the future.

3. Results

To evaluate the method we explored the three G_ϵ in (1), representing different coupling levels, and the Wood-Berry distillation column, [Wood and Berry, 1973]. The dynamics of this common benchmark process is given by

$$G_{\text{WB}}(s) = \begin{pmatrix} \dfrac{12.8e^{-s}}{1+16.7s} & \dfrac{-18.9e^{-3s}}{1+21s} \\ \dfrac{6.6e^{-7s}}{1+10.9s} & \dfrac{-19.4e^{-3s}}{1+14.4s} \end{pmatrix}. \tag{5}$$

White noise with a peak-to-peak-amplitude of 0.9 was added to the simulations. The experiments for the three G_ϵ are shown in Figure 6-8, and the experiment for Wood-Berry is shown in Figure 9. The figures show that the black estimated output curves follow the true (noisy) red curves very well. The obtained model parameters are listed in Table 1. They are all close to their true values. The estimate of G_{WB} is comparable to the one obtained in [Chidambaram and Sathe, 2014]. Our parameters are slightly worse, but obtained from one noisy experiment instead of two noise-free. Since the method in [Chidambaram and Sathe, 2014] assumes strong coupling all G_ϵ cannot be compared.

To this point all examples have been processes that are of the model order we are estimating, and the good accordance of the estimated models is therefore possible. To explore what happens if the process is of higher order we investigated the second order TITO process

$$G_{\text{SOTD}}(s) = \begin{pmatrix} \dfrac{1}{(s+1)^2}e^{-s} & \dfrac{0.3}{(s+1)^2}e^{-s} \\ \dfrac{0.3}{(s+1)^2}e^{-s} & \dfrac{1}{(0.1s+1)^2}e^{-s} \end{pmatrix}. \tag{6}$$

The results for this process are shown in Figure 10 and the Bode plots for the true and estimated models are shown in Figure 11. There is a very good match between the estimated and the true Bode plots up to frequencies where the phase lag is $-180°$, which means that good PID controllers can be designed based on the FOTD models. The Bode plots naturally differ at high frequencies due to the differences in high frequency roll off. The estimated model parameters for G_{SOTD} are listed in the bottom of Table 1. Notice that the estimated time delays $\hat{L}$ are significantly larger than the true time delays L. This is because the extra time constant of the SOTD model is split between the time constant and the time delay of the estimated FOTD model. The large difference in true and estimated time delays indicates that, even though a good PID controller for the FOTD model could be obtained, the performance could be improved significantly by basing control design on a model of higher order. This since the time delay causes fundamental limitations on the achievable bandwidth of

Table 1. Estimated model parameters $\hat{K}$, $\hat{T}$, $\hat{L}$, as well as the true parameters K, T, L, for the five example processes. The rows are ordered P_{11}, P_{12}, P_{21}, and P_{22} for each process.

Process	$\hat{K}$	$\hat{T}$	$\hat{L}$	K	T	L
	0.99	1.00	1.00	1	1	1
$G_{\epsilon=0}$	0.01	0.99	1.15	0	-	-
	0.00	0.09	0.03	0	-	-
	1.00	0.10	1.00	1	0.1	1
	0.98	0.99	1.00	1	1	1
$G_{\epsilon=0.3}$	0.32	1.02	1.02	0.3	1	1
	0.25	0.68	1.12	0.3	1	1
	1.02	0.11	1.00	1	0.1	1
	0.97	0.96	0.99	1	1	1
$G_{\epsilon=0.8}$	0.85	1.17	1.00	0.8	1	1
	0.76	0.98	1.01	0.8	1	1
	1.03	0.13	0.99	1	0.1	1
	15.9	21.0	1.03	12.8	16.7	1
G_{WB}	-18.2	20.7	3.00	-18.9	21.0	3
	5.84	9.62	7.02	6.6	10.9	7
	-20.2	15.1	2.99	-19.4	14.4	3
	1.13	2.09	1.43	1	{1,1}	1
G_{SOTD}	0.32	2.17	1.33	0.3	{1,1}	1
	0.32	1.90	1.47	0.3	{1,1}	1
	1.02	0.17	1.04	1	{0.1,0.1}	1

the system, and a robust control design has a constant product of the time delay and gain crossover frequency.

4. Discussion

The proposed method shows that good process models of a TITO system of FOTD subprocesses can be identified by one single decentralized relay experiment. There is no need to wait for convergence of the limit cycle oscillations and neither to fully understand the conditions for when the different limit cycles occur. The method is shown to work well for the investigated FOTD examples. The result for the SOTD system is also satisfying in the sense that it provides good FOTD approximations of the processes. However, better control performance can be obtained if a higher order model was estimated instead. As stated in Section 2.3 the method can be extended to estimate higher order models, and a generalization of the method to both higher order

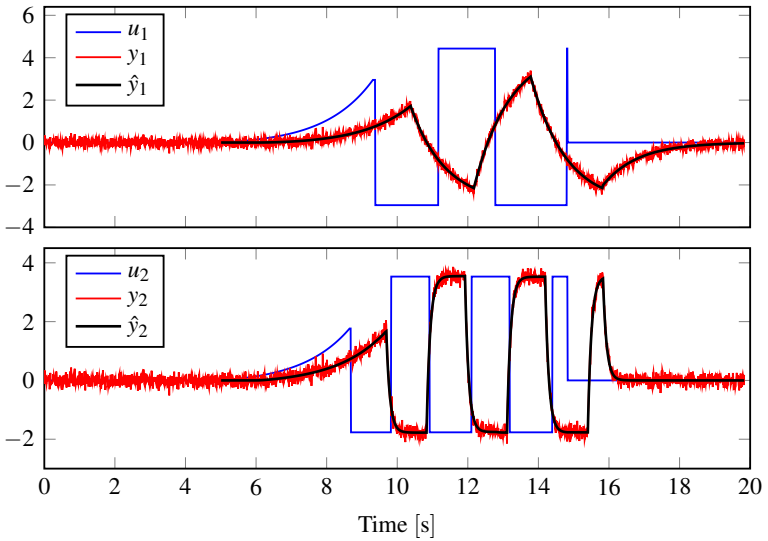


Figure 6. Experiment data and model fit for the weakly coupled example $G_{\epsilon=0}$.

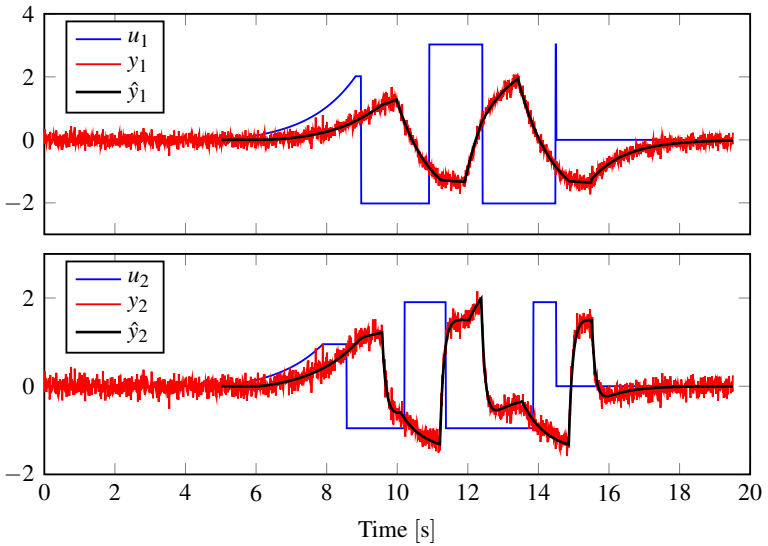


Figure 7. Experiment data and model fit for the intermediately coupled example $G_{\epsilon=0.3}$.

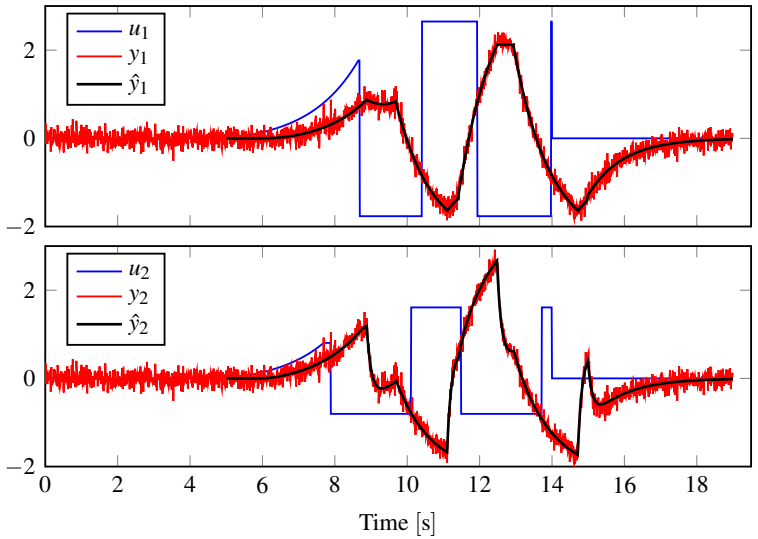


Figure 8. Experiment data and model fit for the strongly coupled example $G_{\epsilon=0.8}$.

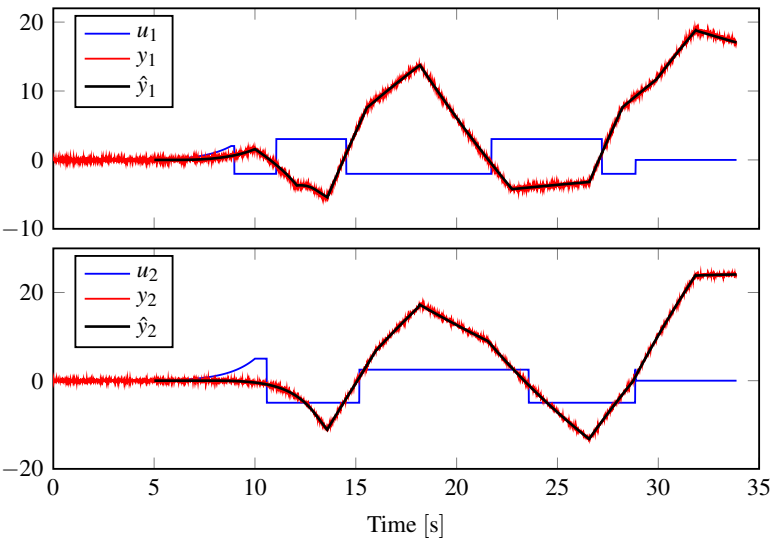


Figure 9. Experiment data and model fit for the Wood-Berry distillation column G_{WB}.

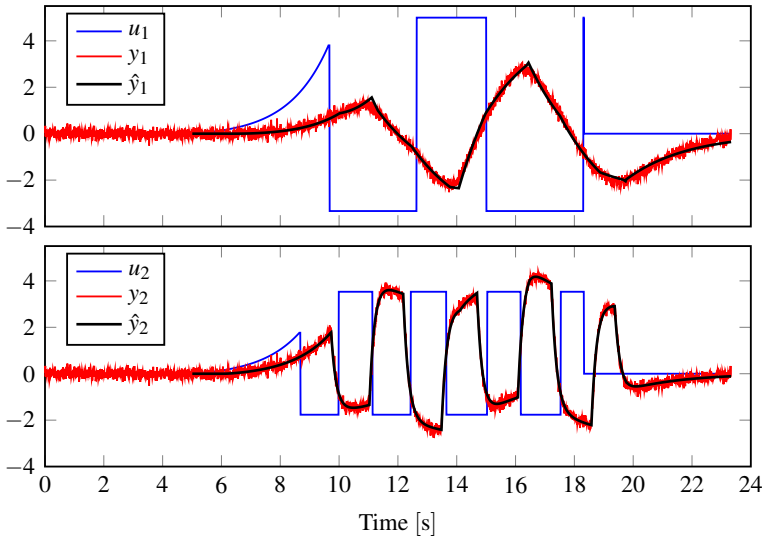


Figure 10. Experiment data and model fit for the second order time delayed process G_{SOTD}.

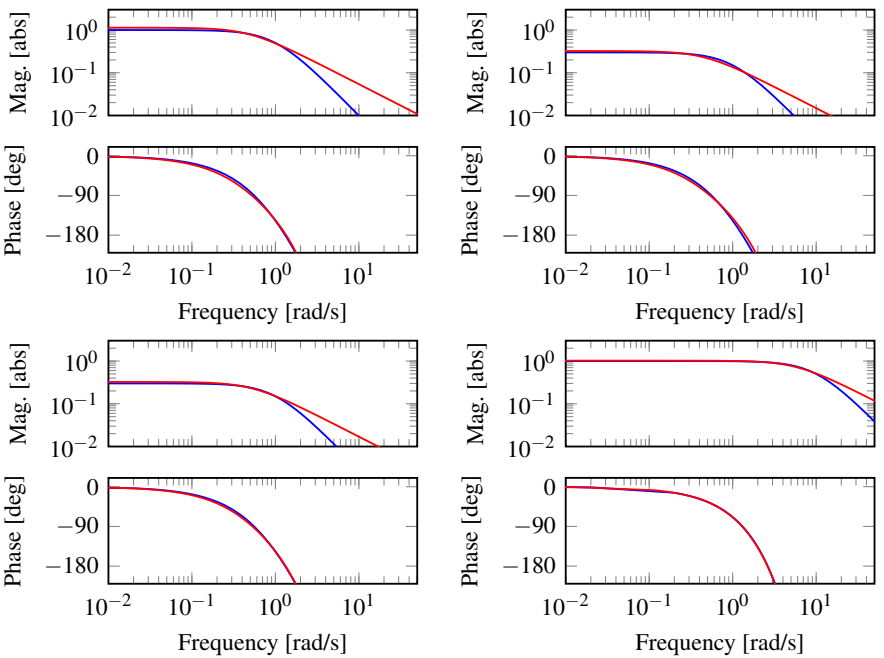


Figure 11. Bode plots for the true SOTD TITO process in blue, and the estimated FOTD TITO model in red.

models as well as arbitrary MIMO systems to see how it scales, and if the excitation from one single experiment is sufficient, are interesting continuations of this work. Practical issues like parameter choices, experiment start-up and termination, and of course how the controllers should be tuned from the obtained models, are other aspects that should be explored further. Noise has been considered in this paper, but the method's sensitivity to other disturbances has not yet been investigated.

Acknowledgment

The authors are members of the LCCC Linnaeus Center and the ELLIIT Excellence Center at Lund University.

References

Åström, K. J. and T. Bohlin (1966). "Numerical identification of linear dynamic systems from normal operating records". In: *Theory of self-adaptive control systems*. Springer, pp. 96–111.

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.

Berner, J., T. Hägglund, and K. J. Åström (2016a). "Asymmetric relay autotuning–Practical features for industrial use". *Control Engineering Practice* **54**, pp. 231–245.

Berner, J., T. Hägglund, and K. J. Åström (2016b). "Improved Relay Autotuning using Normalized Time Delay". In: *2016 American Control Conference (ACC)*. IEEE, pp. 1869–1875.

Boyd, S., M. Hast, and K. J. Åström (2016). "MIMO PID tuning via iterated LMI restriction". *International Journal of Robust and Nonlinear Control* **26**:8, pp. 1718–1731.

Campestrini, L., P. R. Barros, and A. S. Bazanella (2006). "Auto-tuning of PID controllers for MIMO processes by relay feedback". *IFAC Proceedings Vol.* **39**:2, pp. 451–456.

Chidambaram, M. and V. Sathe (2014). *Relay Autotuning for Identification and Control*. Cambridge University Press.

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

Loh, A. and V. Vasnani (1994). "Necessary conditions for limit cycles in multiloop relay systems". *IEE Proceedings-Control Theory and Applications* **141**:3, pp. 163–168.

Luyben, W. L. (1987). "Derivation of transfer functions for highly nonlinear distillation columns". *Industrial and Engineering Chemistry Research* **26**:12, pp. 2490–2495. DOI: 10.1021/ie00072a017.

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

Soltesz, K., T. Hägglund, and K. J. Åström (2010). "Transfer function parameter identification by modified relay feedback". In: *American Control Conference*. Baltimore, MD, pp. 2164–2169.

Wang, Q.-G., B. Zou, T.-H. Lee, and Q. Bi (1997). "Auto-tuning of multivariable PID controllers from decentralized relay feedback". *Automatica* **33**:3, pp. 319–330.

Wood, R. and M. Berry (1973). "Terminal composition control of a binary distillation column". *Chemical Engineering Science* **28**:9, pp. 1707–1717.

Yu, C.-C. (2006). *Autotuning of PID controllers: A relay feedback approach*. Springer Science & Business Media.

Paper VI

Practical Evaluation of a Novel Multivariable Relay Autotuner with Short and Efficient Excitation

Josefin Berner Kristian Soltesz Karl Johan Åström

Tore Hägglund

Abstract

In this paper we propose an autotuning method that combines a setup for decentralized relay autotuning of two-input two-output systems with an identification method that uses short experiments to estimate up to second-order time-delayed systems. A small modification of the experiment gives better low-frequency excitation and improved models. The method is successfully demonstrated in simulations and on a quadruple tank process.

1. Introduction

PID controllers are, and will probably remain, the most common controller type used in industry. Since the commonly used PID tuning rules rely on dynamic models of the processes to be controlled, automatic methods of obtaining these models are of practical value. For single-input single-output (SISO) systems, the introduction of the relay autotuner [Åström and Hägglund, 1984] proved to be a successful way to do this. The relay autotuner has since then been improved and modified in e.g. [Luyben, 1987; Shen et al., 1996; Kaya and Atherton, 2001; Berner et al., 2016]. A review of identification from relay experiments, written a few years ago, is found in [Liu et al., 2013].

Not all industrial systems are, or can be approximated by, SISO systems. This motivates an extension of automatic tuning methods to handle multivariable systems. The literature on this subject is not as extensive as for the SISO case, but there are some papers written on multivariable autotuning, of which [Wang et al., 1997] should be mentioned as it proposed the decentralized relay experiments that we will use in this paper. In [Chidambaram and Sathe, 2014] this method was the basis for a method to find FOTD or SOTD models from the experiments. Other papers have been published where each sub-loop is tuned individually, or where a sequential tuning procedure is used [Yu, 2006; Loh et al., 1993]. All these methods require several experiments, which is a drawback. A decentralized method that only needs one short experiment, without any assumptions on coupling-level of the subsystems, was proposed in [Berner et al., 2017]. In this paper we extend and improve that method. This is done by modifying the experiment to increase the low-frequency excitation, and by using an improved identification method from [Berner and Soltesz, 2017] where both first-order time-delayed (FOTD) and second-order time-delayed (SOTD) models are found, even in the presence of non-stationary starting conditions. The proposed method is evaluated by simulations, and is also tested on a physical two-input two-output (TITO) process, a quadruple tank process available in our control laboratory.

2. Method

The proposed method consists of combining the SISO-autotuner from [Berner and Soltesz, 2017] with the decentralized framework from [Berner et al., 2017] to get a well-functioning autotuner for TITO-systems that provides FOTD and SOTD models from a short experiment. The experiment is slightly modified to increase low-frequency excitation, as will be described in Section 4. From the obtained models, a centralized PID controller is designed, based on the optimization method described in [Boyd et al., 2016]. The obtained controller performance is evaluated in Section 5.

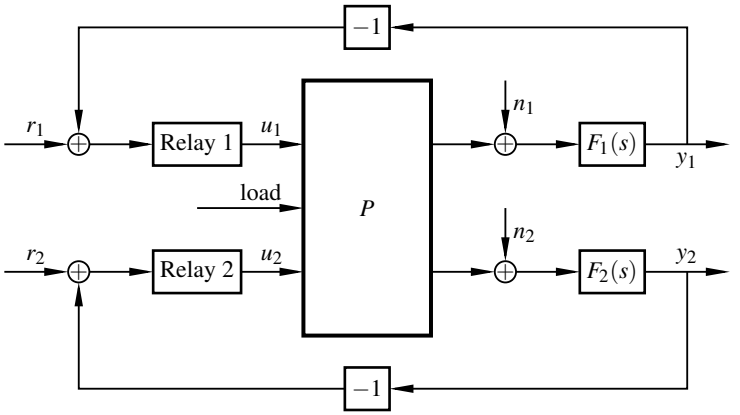


Figure 1. Setup of the relay experiment for the two-input two-output process P, with optional noise filters F_1 and F_2.

2.1 Decentralized asymmetric relay experiment

The experiment setup used in this paper is shown in Figure 1. In the decentralized relay experiment both loops are closed with relay feedback simultaneously. The relays used are asymmetric, with different asymmetry levels, and more extensively described in [Berner et al., 2016]. The relay feedback will cause both loops to oscillate, exciting the output signals in frequency intervals relevant for PID control. The original decentralized method proposed in [Wang et al., 1997] was modified in [Berner et al., 2017] to use an identification method that do not require limit cycle convergence of the experiment, which allowed a significantly shorter experiment time.

2.2 Identification of models

As was described in [Berner et al., 2017], each of the output ports consist of a sum of the subsystem outputs. For the TITO case this means that the output signals are $y_1 = P_{11}u_1 + P_{12}u_2$, and $y_2 = P_{21}u_1 + P_{22}u_2$. Thus, we can identify the subsystems related to one output separately from the subsystems related to the other output. The sub-models estimated in this work are FOTD and SOTD models of the form

$$P_{i,j}(s) = \frac{b}{s^n + a_1 s^{n-1} + \ldots + a_n} e^{-sL}, \tag{1}$$

where $1 \leq n \leq 2$. The model parameters are estimated along with the initial state(s) $\boldsymbol{x_0}$ to get the parameter vector $\boldsymbol{\theta} = \begin{bmatrix} b & \boldsymbol{a} & L & \boldsymbol{x_0} \end{bmatrix}$, from an output error minimization described in [Berner and Soltesz, 2017]. If desired, stability of the model can be enforced by introducing constraints on the parameters in the minimization method. Since the output data is the sum of the output of two subsystems it means that both $\boldsymbol{\theta_1}$ and $\boldsymbol{\theta_2}$ should be found simultaneously. The strategy used in this paper is to first

find an FOTD or SOTD model $\hat{P}_{11}$, assuming that $y_1 = P_{11}u_1$, and a model $\hat{P}_{12}$ assuming that $y_1 = P_{12}u_2$, where the model selection is done according to the Akaike Information Criteria [Akaike, 1974]. A second set of models is then estimated from the alternative assumption that $y_1 - \hat{P}_{12}u_2 = P_{11}u_1$ and $y_1 - \hat{P}_{11}u_1 = P_{12}u_2$. Combinations of these separate models are used as starting values for the estimation of the combined parameter vector $\boldsymbol{\theta} = \begin{bmatrix}\boldsymbol{\theta_1} & \boldsymbol{\theta_2}\end{bmatrix}$.

2.3 MIMO PID tuning

In this paper we use the multi-input multi-output (MIMO) PID tuning method described in [Boyd et al., 2016], with some small modifications. The idea of the method is to find controller matrices K_P, K_I and K_D by solving an optimization problem, where

$$||(P(0)K_I)^{-1}||_\infty \tag{2}$$

is minimized subject to constraints on the maximum sensitivities $||S||_\infty$ and $||T||_\infty$, as well as a constraint on the control signal. In the proposed version in the referred paper, a first order filter is applied to the derivative part of the controller. We have chosen to instead use one of the suggested modifications, and use a second order filter on the full controller. Hence our controller structure is

$$C(s) = \frac{1}{1 + sT_f + (sT_f)^2/2}\left(K_P + \frac{1}{s}K_I + sK_D\right), \tag{3}$$

where T_f is the filter time constant, in this paper chosen as

$$T_f = \frac{1}{k\max(\omega_c)}, \tag{4}$$

where k is a factor chosen to be 5, and $\max(\omega_c)$ is the largest crossover frequency of the obtained sub-models. By introducing this filter, and sending the filtered process output to the optimization method, we no longer need the restriction on the control signal, and that constraint is hence removed.

For a TITO (or general MIMO) system there could be a lot of questions asked about the controller structure. For example, should it be centralized like this? Should you really have integral parts on all entries in the controller matrix? Should the same filter be used for all input-output-pairings? Since the focus of this paper is not MIMO PID design, but rather how to obtain good enough models for PID design, these questions will not be answered here. We merely use the described MIMO PID design as an example of what results could be obtained from our autotuning method. The models we obtain could just as well be used to tune decentralized PID controllers with or without decoupling, or by any other method preferred by the user.

3. Example Processes

We will evaluate the proposed method on the Wood-Berry distillation column [Wood and Berry, 1973], commonly used as a benchmark process for TITO control, and a modified version of the quadruple tank, described in [Johansson, 2000], used in many control laboratories. The dynamics of the Wood-Berry distillation column are given by

$$G_{WB} = \begin{pmatrix} \dfrac{12.8e^{-s}}{1+16.7s} & \dfrac{-18.9e^{-3s}}{1+21s} \\ \dfrac{6.6e^{-7s}}{1+10.9s} & \dfrac{-19.4e^{-3s}}{1+14.4s} \end{pmatrix}. \tag{5}$$

The quadruple tank in our control lab, shown in Figure 2, has been modified from the one in [Johansson, 2000] to get faster dynamics. The linearized minimum-phase configuration for this tank process can be modeled by

$$G_{QT} = \begin{pmatrix} \dfrac{0.14e^{-s}}{s+0.043} & \dfrac{0.0088e^{-s}}{s^2+0.19s+0.0061} \\ \dfrac{0.0088e^{-s}}{s^2+0.19s+0.0061} & \dfrac{0.14e^{-s}}{s+0.043} \end{pmatrix}, \tag{6}$$

where the time delay is a simplification of the dynamics in the pumps and sensors.

Some characteristics of the example processes are worth mentioning. The Wood-Berry distillation column has only FOTD entries and should be well-estimated by the method in [Berner et al., 2017], while the quadruple tank is a mix of first-order and second-order systems, and could therefore be improved by the SOTD estimations from [Berner and Soltesz, 2017]. G_{WB} has high and negative gains, which puts requirements on the generality of the autotuner implementation. The quadruple tank model is linearized around a working point at half the tank height, so in order to do experiments it is first required to bring the system up to the operating point. This startup raises the question of when the system is in steady-state, and the estimation of initial states from [Berner and Soltesz, 2017] is useful to avoid that problem. The quadruple tank model is symmetric, which means that if the asymmetry levels in the two relays were the same, the input-output data from the two loops would be identical. That would reduce the information for the identification process, so the different asymmetry levels used by this method are very beneficial for this process.

3.1 Experiment settings

For the quadruple tank the working point was chosen as half-full tanks, and the nominal control signals were adjusted to achieve this. In all simulation plots, these working points has been subtracted from the result to get a system oscillating around zero. The Wood-Berry column was assumed to oscillate around a working point normalized to zero. The data sequences used by the identification method only

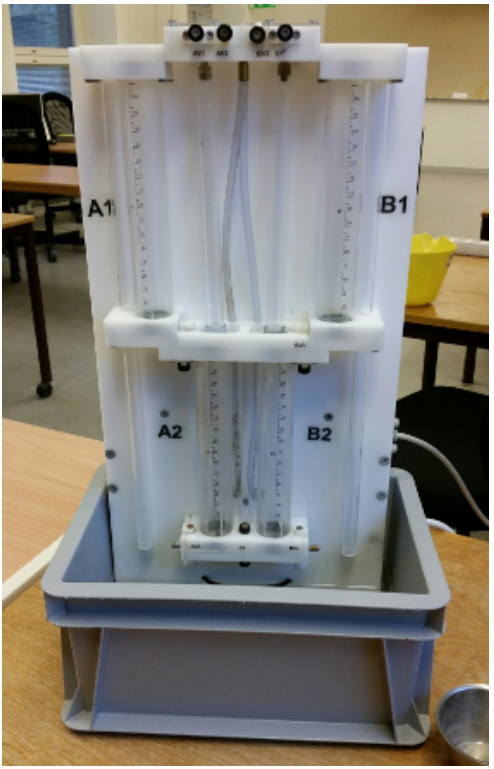
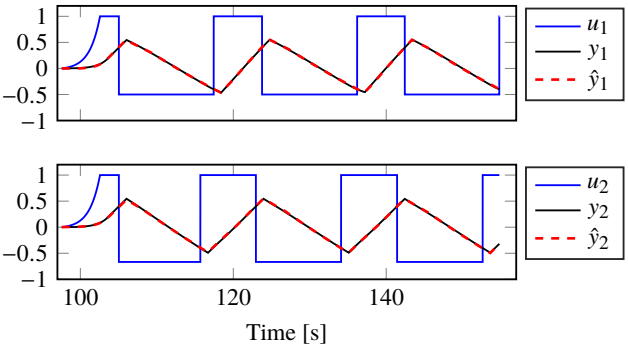


Figure 2. The quadruple tank used for experiments. The inputs are the signals to the two pumps that pump water to both the upper and the lower tanks in different proportions depending on the configuration. The output signals, y_1 and y_2, are the water levels in the two lower tanks.

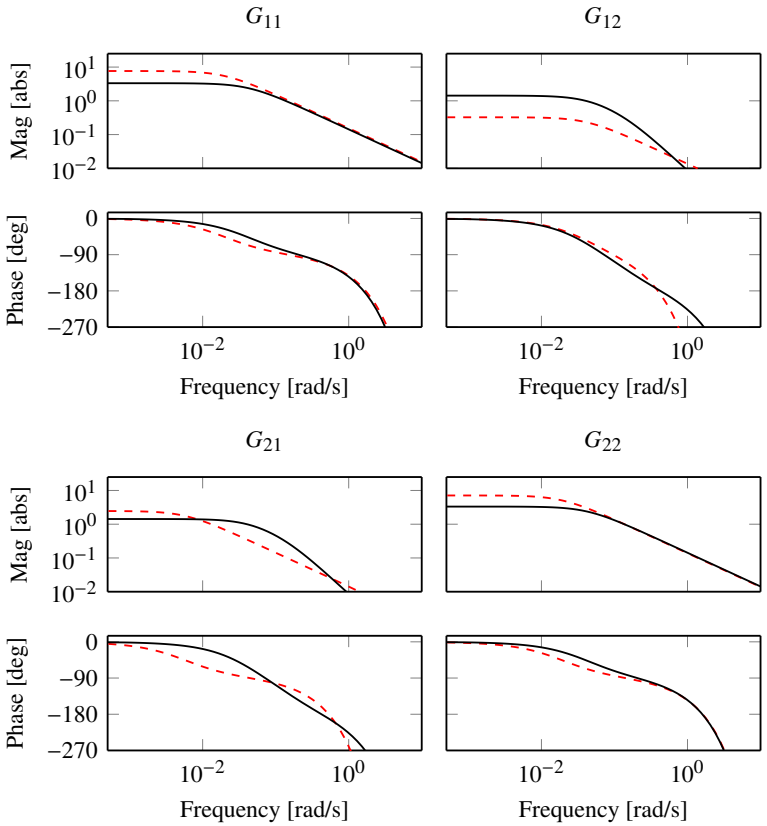
contain the actual experiments, and the plots in this paper only show this, and not the part where the system is brought to its working point or where the noise level is measured. The sample time was $t_s = 0.005$ s, and the hysteresis level in noise-free simulations was $h = 0.4$. The asymmetry levels of the two relays were set to $\gamma_1 = 2$ and $\gamma_2 = 1.5$.

4. Modifications to Experiment

The examples in [Berner et al., 2017] showed a very good fit of the model output data from short experiments consisting of three relay switches. However, additional experiments indicated that the obtained models sometimes gave poor estimates of the static gains. For the classic SISO relay autotuner the static gain is irrelevant since

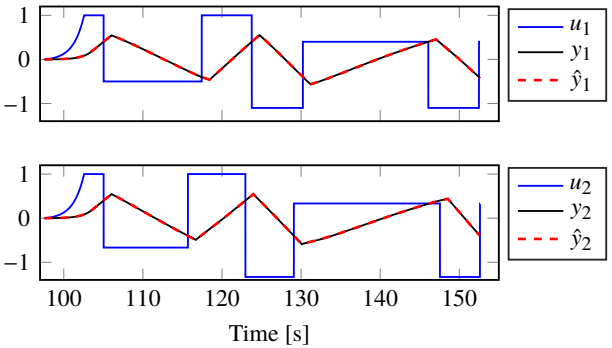


(a) Simulation data and obtained model output data $\hat{y}$.

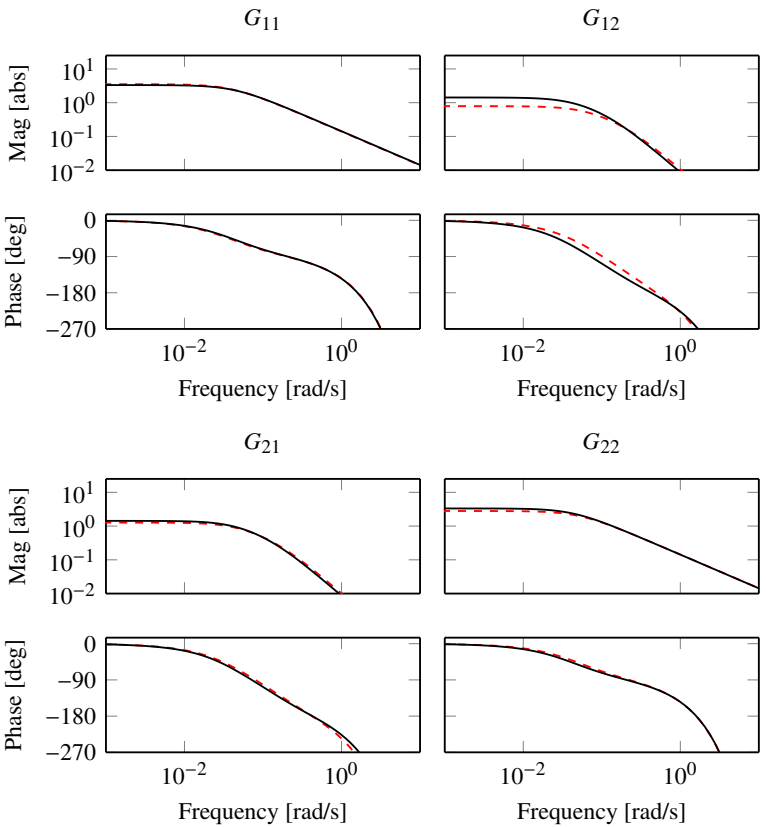


(b) Bode plots for the true model in solid black and estimated model in dashed red.

Figure 3. Simulation results for G_{QT} from original experiment.



(a) Simulation data and obtained model output data $\hat{y}$.



(b) Bode plots for the true model in solid black and estimated model in dashed red.

Figure 4. Simulation results for G_{QT} from modified experiment.

it only uses the critical point. However, a good estimate of the static gain matrix is needed for the multivariable design used in this paper, see (2). This motivated a slight modification of the experiment. Simply increasing the experiment length did not help much, as illustrated for the quadruple tank in Figure 3, where the obtained model output fits the data extremely well, but the models are not that satisfactory. Instead we decided to increase the low-frequency excitation of the experiment by changing the on and off amplitudes of the relay in order to induce a step in u_{ref}. By doing this small modification we get the results in Figure 4. As can be seen the experiment lengths are more or less the same, the process output still oscillates around the same level, but the obtained models are much better.

5. Results

5.1 Simulations

In the simulation study we compare the controller performance for the autotuner, to a PID controller tuned by the same method, but from the true process model. To make the simulations a bit more realistic we added band-limited white noise with the standard deviation $\sigma_n = 0.1$ for Wood-Berry, and $\sigma_n = 0.035$ for the quadruple tank. The reason for the different levels are that the processes are not normalized to the same scale and hence have very different gains.

The Wood-Berry Distillation Column The simulation data for G_{WB} is shown in Figure 5. The output data fit is very good, and the obtained model is essentially identical to the true model. Since the models are identical, so are the optimized controllers. The response from the controllers to setpoint changes is seen in Figure 6.

Quadruple tank model The simulation data for the quadruple tank model, G_{QT}, is shown in Figure 7. Bode plots of the obtained models are shown in Figure 8, the controller performance for a step in setpoint, and a step in input load disturbance, is shown in Figure 9 and Figure 10 respectively. The model is good, but the small difference in gain of G_{21} results in somewhat differing behavior to the controller for that subsystem. However, the performance is still satisfactory.

5.2 Quadruple tank experiment

The proposed method was also evaluated on a real quadruple tank process in the control lab at Lund University. The sensors were too noisy to perform the experiment in a good way, so a low-pass filter

$$F(s) = \frac{1}{0.1s + 1} \tag{7}$$

was added to each output, as in Figure 1. To get a model of the unfiltered dynamics from the identification, the input identification data was run through the same filter.

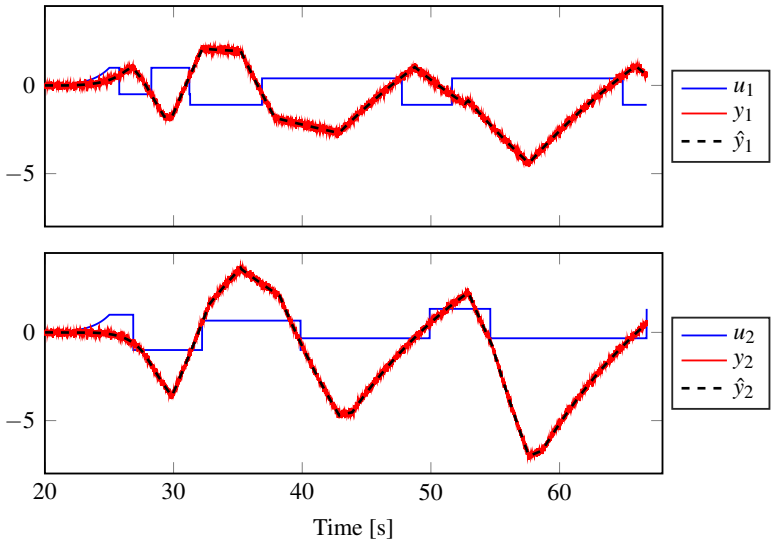


Figure 5. Simulation data for G_{WB}, along with the output data $\hat{y}$ resulting from the obtained model.

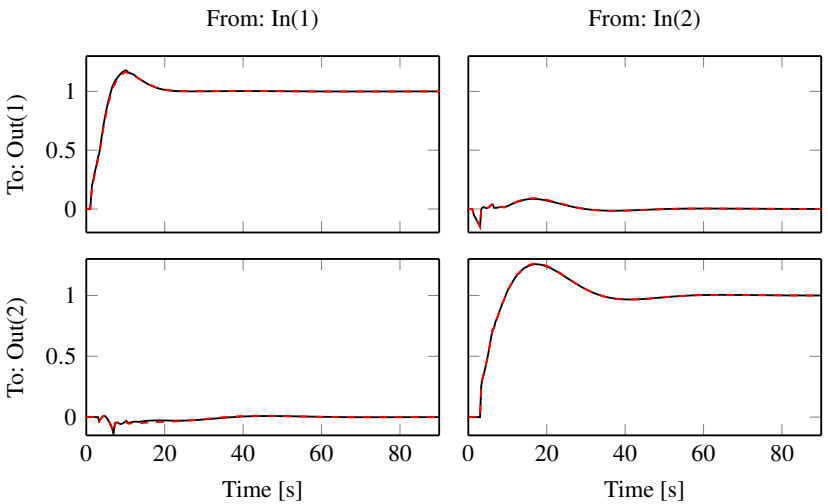


Figure 6. Controller performance for a step change in setpoint for G_{WB}. The controller tuned from the true model is shown in solid black, and the controller tuned for the estimated model in dashed red.

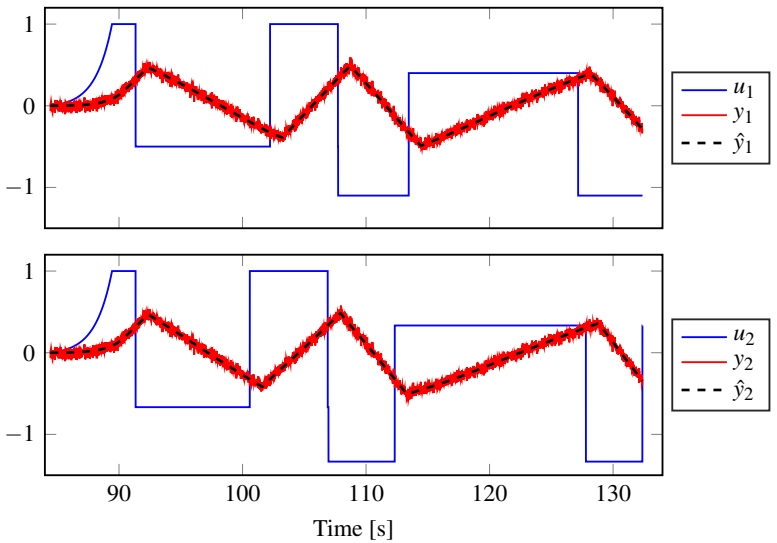


Figure 7. Simulation data for G_{QT}, along with the output data $\hat{y}$ resulting from the obtained model.

Hence, the filters do not affect the obtained model, but they slightly change the experiment excitation.

To evaluate the method we performed 10 experiments. Data (filtered) for one of these is shown in Figure 11, and Bode plots for the resulting models are shown for all of them in Figure 12. A MIMO PID controller was designed for each of the obtained models. The controller performance for the model obtained from the experiment in Figure 11 (the one showed by a thick red line in Figure 12) is shown in Figure 13. At time 120 s a step change in the setpoint for y_1 was made, followed by a step change for y_2 at 180 s. At time 250 s, a load disturbance was added by manually pouring 0.5 dl water to the tank with output y_1, and at 350 s a corresponding load was put on y_2.

6. Discussion

The simulation study shows that good models are obtained for the example systems, using the proposed method with the modified experiment, even with the addition of noise. The need for the extra step more or less doubles the experiment time compared to the SISO experiments in [Berner and Soltesz, 2017] and the first TITO tests in [Berner et al., 2017]. However, the experiment is still short since it does not have to wait for limit cycle convergence and only runs for a total of 5 switches.

The results were improved by the increased low-frequency excitation. This increased excitation can be achieved in many ways, but by changing the setpoint for u

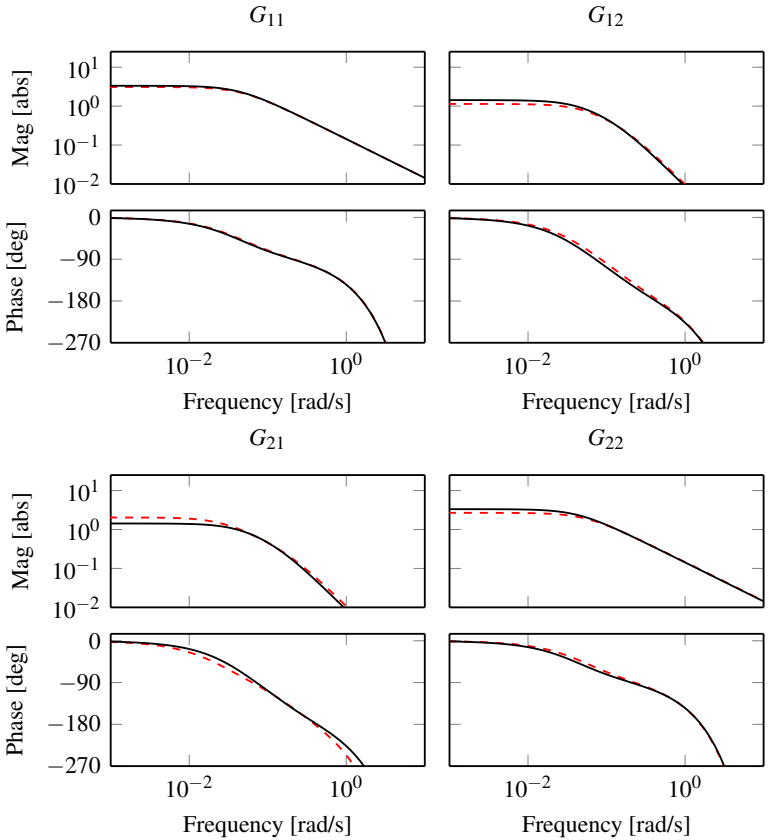


Figure 8. Bode plots of the obtained models for G_{QT}. The true model is shown in black, the estimated model in dashed red.

instead of taking a step in y, we do not cause the process to drift away from its working point, which is an advantage. The exact size of the step in u_{ref} could be further investigated, but by moving in the direction from the high relay amplitude towards the low relay amplitude, a larger step can be taken without risking to terminate the oscillations.

From the experiments on the quadruple tank it can be noted that the experiment seems to work very well. All obtained models are similar to each other, even if one or two differ slightly in static gain of G_{21}, or for large phase lags. The obtained controller shows satisfactory responses to both setpoint changes and load disturbances. The obtained model is not symmetric, and neither is therefore the controller. This is due to uneven wear in the process and can be seen by the somewhat different responses in y_1 and y_2. Another thing worth noting from the experiments, is the characteristics

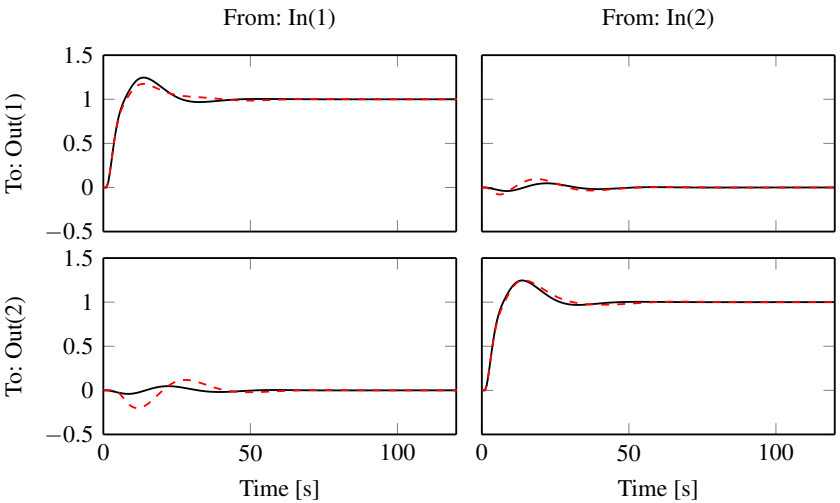


Figure 9. Controller performance for a step change in setpoint for G_{QT}. The controller tuned from the true model is shown in solid black, and the controller tuned for the estimated model in dashed red.

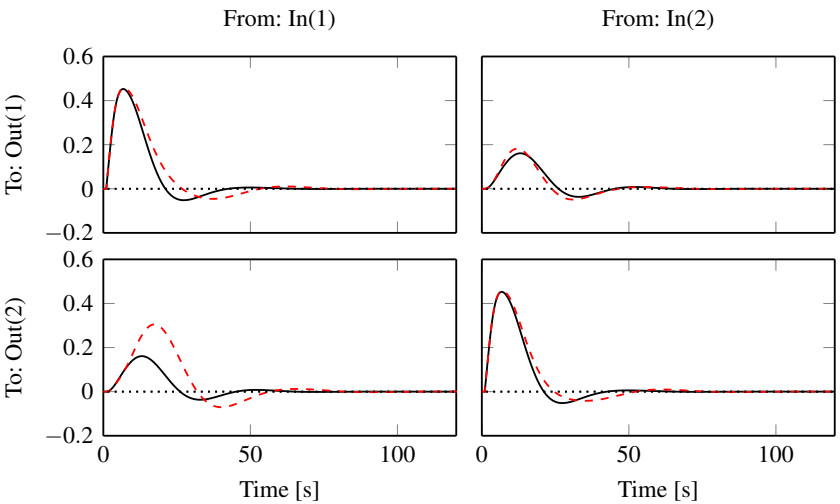


Figure 10. Controller performance for a step change in input load disturbance for G_{QT}. The controller tuned from the true model is shown in black, and the controller tuned for the estimated model in red.

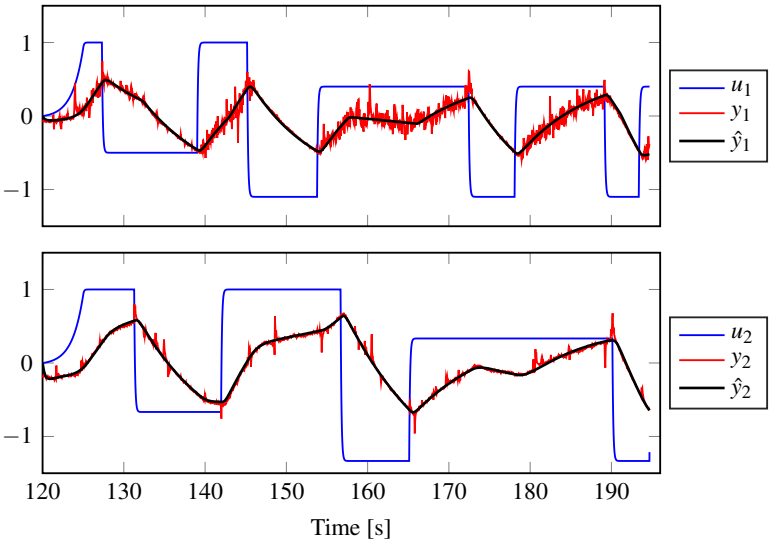


Figure 11. Experiment data from the quadruple tank process. Both the input and output have been filtered by $F(s)$, and the working point of approximately $u = 5$, $y = 5$, has been subtracted.

of the noise. It is clearly seen from y_1 in Figure 11 that the noise is much larger on the way up (that is, when more water is pumped in to the tank) than on the way down. It is also clear by comparing the noise in Figure 11 with the one in Figure 7 that the real noise is not at all as white and even as in the simulations. This does, however, not seem to deteriorate the results of the proposed autotuner.

7. Conclusion

We have proposed an autotuner for TITO systems that gives FOTD or SOTD models for each sub-model. The experiment is extended by a step in the relay amplitudes, which allows better models to be obtained. The method handles start from non-stationarity, and the experiment duration is short. The results are good for the evaluated simulation examples, as well as for the experiments on the quadruple tank process. This shows that the proposed extended version of the fast and simple relay autotuner can be succesfully used also for TITO systems, and that more advanced, time-consuming system identification methods are unnecessary in this case.

Acknowledgment

The authors are members of the LCCC Linnaeus Center and the ELLIIT Excellence Center at Lund University.

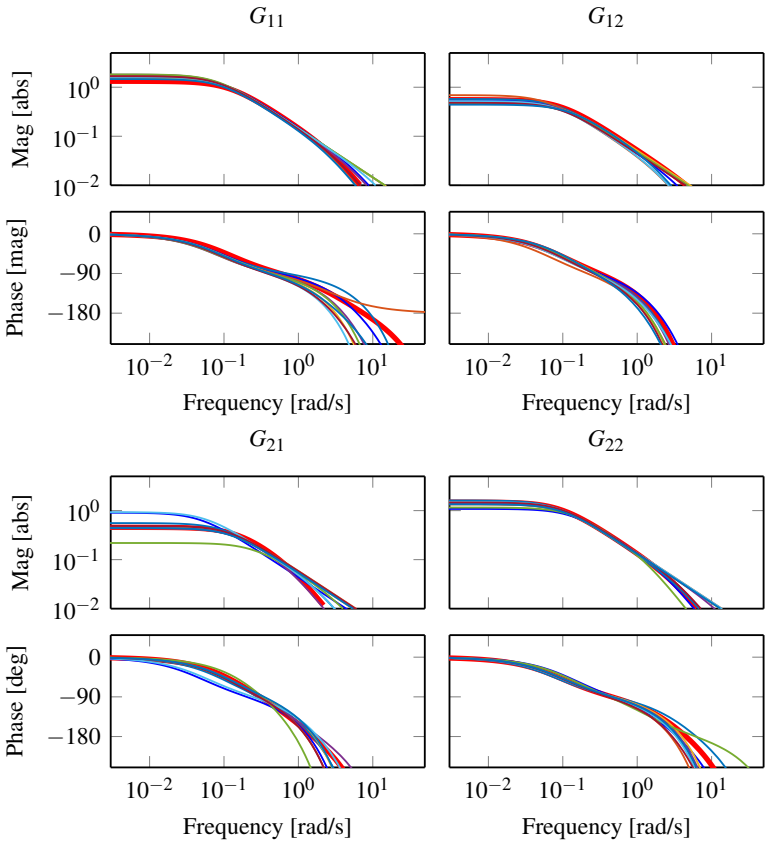


Figure 12. Bode plots for the models obtained from ten experiments on the quadruple tank. The thick red line is the model from the experiment illustrated in Figure 11 and used in Figure 13.

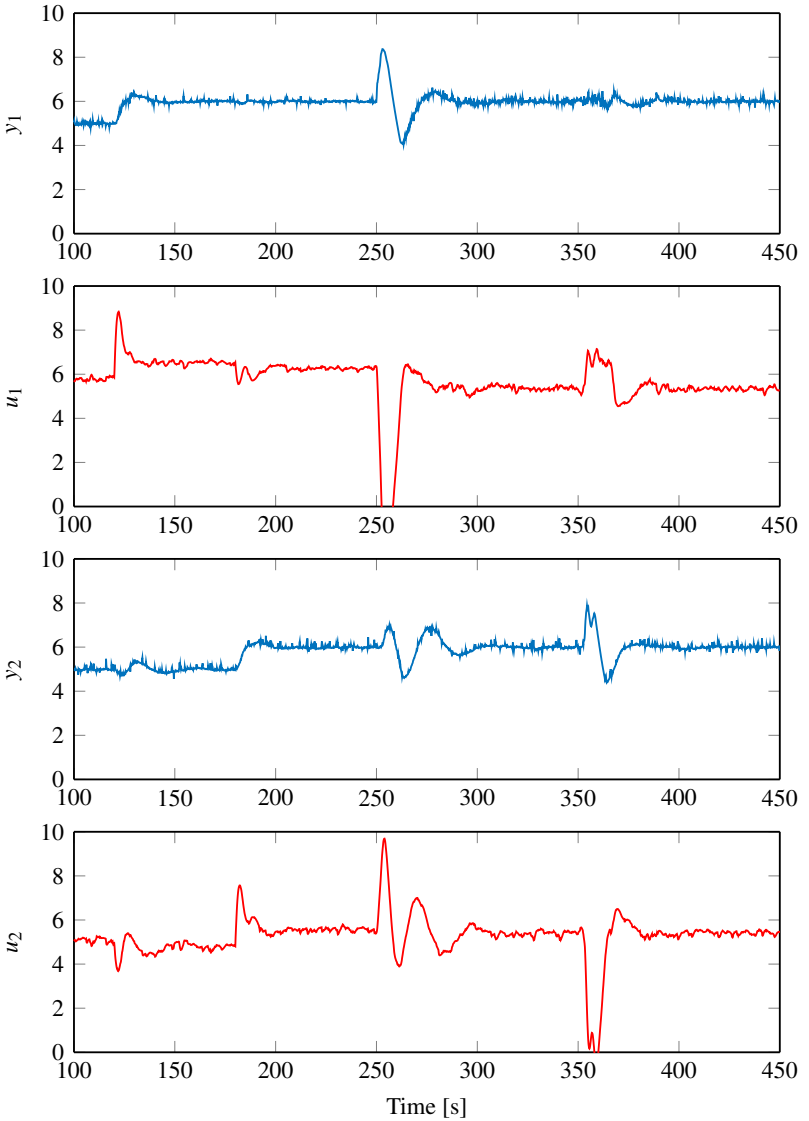


Figure 13. The quadruple tank controlled by our obtained controller. Step changes occur at time 120 s and 180 s, and load disturbances enter at time 250 s and 350 s for the two output tanks respectively. Here the working points have not been subtracted, and the plot starts at 100 s when the system has reached its stationary level.

References

Akaike, H. (1974). "A new look at the statistical model identification". *IEEE transactions on automatic control* **19**:6, pp. 716–723.

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.

Berner, J., T. Hägglund, and K. J. Åström (2016). "Asymmetric relay autotuning–Practical features for industrial use". *Control Engineering Practice* **54**, pp. 231–245.

Berner, J. and K. Soltesz (2017). "Short and robust experiments in relay autotuners". In: *2017 IEEE Conference on Emerging Technologies and Factory Automation (ETFA)*. Limassol, Cyprus.

Berner, J., K. Soltesz, T. Hägglund, and K. J. Åström (2017). "Autotuner identification of TITO systems using a single relay feedback experiment". In: *2017 IFAC World Congress*. Toulouse, France.

Boyd, S., M. Hast, and K. J. Åström (2016). "MIMO PID tuning via iterated LMI restriction". *International Journal of Robust and Nonlinear Control* **26**:8, pp. 1718–1731.

Chidambaram, M. and V. Sathe (2014). *Relay Autotuning for Identification and Control*. Cambridge University Press.

Johansson, K. H. (2000). "The quadruple-tank process: a multivariable laboratory process with an adjustable zero". *IEEE Transactions on control systems technology* **8**:3, pp. 456–465.

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

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

Loh, A. P., C. C. Hang, C. K. Quek, and V. U. Vasnani (1993). "Autotuning of multiloop proportional-integral controllers using relay feedback". *Industrial and Engineering Chemistry Research* **32**:6, pp. 1102–1107. ISSN: 0888-5885. DOI: 10.1021/ie00018a017.

Luyben, W. L. (1987). "Derivation of transfer functions for highly nonlinear distillation columns". *Industrial and Engineering Chemistry Research* **26**:12, pp. 2490–2495. DOI: 10.1021/ie00072a017.

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

Wang, Q.-G., B. Zou, T.-H. Lee, and Q. Bi (1997). "Auto-tuning of multivariable PID controllers from decentralized relay feedback". *Automatica* **33**:3, pp. 319–330.

Wood, R. and M. Berry (1973). "Terminal composition control of a binary distillation column". *Chemical Engineering Science* **28**:9, pp. 1707–1717.

Yu, C.-C. (2006). *Autotuning of PID controllers: A relay feedback approach*. Springer Science & Business Media.

Printed by Media-Tryck, Lund 2017 NORDIC ECOLABEL, 3041 0903

LUND
UNIVERSITY

Department of Automatic Control
P.O. Box 118, 221 00 Lund, Sweden
www.control.lth.se

PhD Thesis TFRT-1118
ISBN 978-91-7753-446-4
ISSN 0280–5316