Short and Robust Experiments in Relay Autotuners

Berner, Josefin; Soltesz, Kristian

Published in:
Proceedings of the 2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation; ETFA2017

DOI:
10.1109/ETFA.2017.8247696

2017

Document Version:
Peer reviewed version (aka post-print)

Short and Robust Experiments in Relay Autotuners

Josefin Berner
Dept. Automatic Control
Lund University
Lund, Sweden
Email: josefin.berner@control.lth.se

Kristian Soltesz
Dept. Automatic Control
Lund University
Lund, Sweden
Email: kristian.soltesz@control.lth.se

***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.**

I. Introduction

A. 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 [1]) on the topic, and several tuning rules, ranging from simple rules of thumb [2], to optimization-based alternatives like e.g. [3].

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 [4]. 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 have been proposed to identify FOTD models (and sometimes also SOTD models) in for instance [5], [6], [7]. 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 [8] 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.

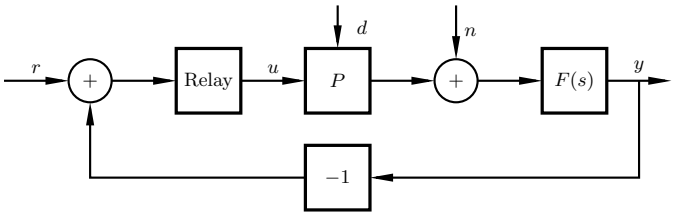

Fig. 1: Block diagram of the identification setting, showing process P, asymmetric relay, filter F, identification input u and output y, load disturbance d, and measurement noise n.

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.

B. 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.

II. Experiment

A. The relay

An asymmetric relay function, as defined in [9] 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 [9], 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 IV.

B. 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 [4] it has been assumed and required that the user starts the experiment when the system is in steady-state. In [9] 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 separate from zero is added to the identification method in this paper and described in Section III-D.

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 [9]. 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.

C. Experiment parameters

The parameters used for the experiments in Section IV are listed in Table I. 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 [9] but are reasonable for these experiments.

The initial state is set to

$$\boldsymbol{x_0} = -A^{-1}Bv_0\Delta u_{max}, \tag{1}$$

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.

III. Identification

A. Model Parametrization

The models we consider in this paper are of the form

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

TABLE I: 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

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 II-B, 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 $\boldsymbol{a} = \begin{bmatrix}a_1 & \dots & a_n\end{bmatrix}$, $\boldsymbol{b} = \begin{bmatrix}b_1 & \dots & b_m\end{bmatrix}$, and $\boldsymbol{x_0} = \begin{bmatrix}x_1(0) & \dots & x_n(0)\end{bmatrix}$.

B. Output Error Formulation

The output data from the experiment on the process P is collected in $\boldsymbol{y} = \begin{bmatrix}y_1 & \dots & y_N\end{bmatrix}^\top$, 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] [10] 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

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

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

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

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 [11] 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

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

C. 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

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

Our model is thus given by

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

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 II, 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 [12]. Results for the discrete time counterparts are found in [13]. In this paper, simplified expressions of those in [12], 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 II, 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

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

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 [12].

D. Initial State Sensitivity

By definition $\nabla_{\boldsymbol{x}_0}\boldsymbol{x}(0) = I_n$. The assumption made in Section III-C, 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

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

E. Calculating the gradient expressions

The expressions for the gradients have been previously derived in [12], 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

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

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

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

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

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

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

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

This gives the following expression for the gradient:

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

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 III-D.

F. 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 [14]. 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 [15]. 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.

G. 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) [16]. The chosen model is the one with lowest value of

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

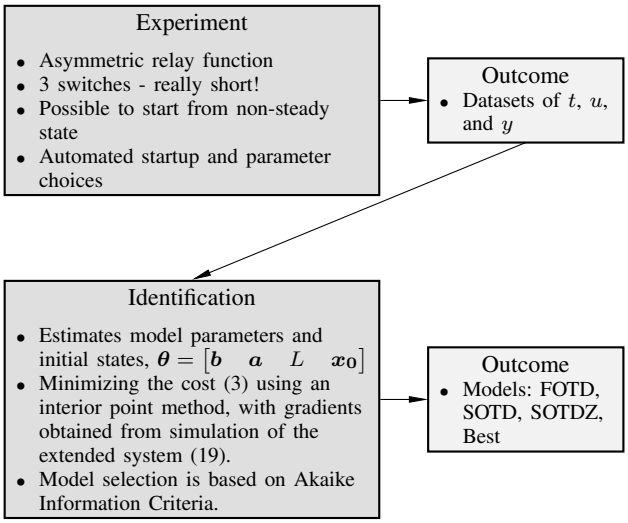

Fig. 2: Schematic summary of the proposed experiment and identification method.

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.

IV. Simulation study

The proposed method, briefly summarized in Figure 2, is evaluated by four example processes from the test batch in [1], namely:

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

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

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

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

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 [15].

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, while the best models seem to do. The "best model" was chosen using AIC, see Section III-G. 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

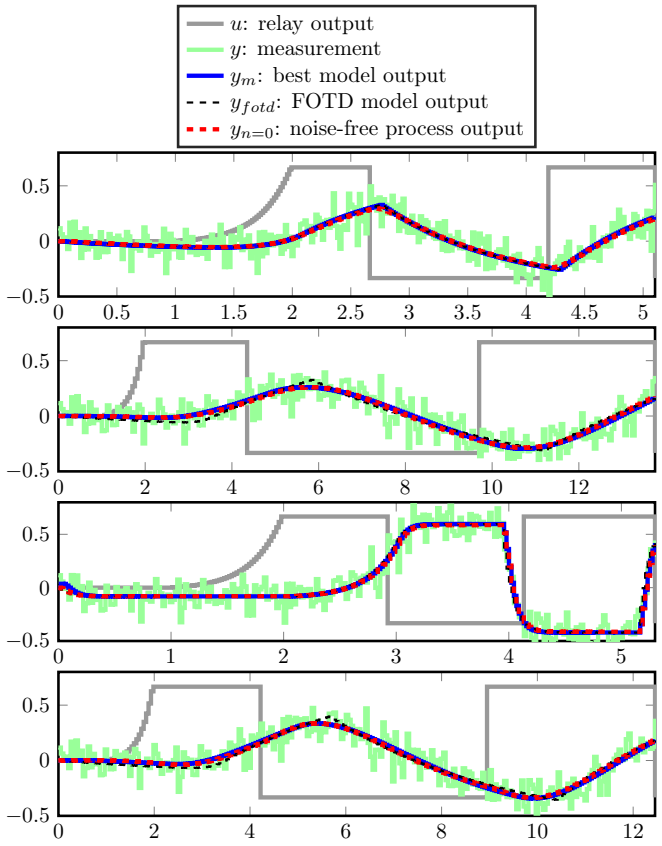

Fig. 3: Outputs from the experiments and identified models. The different subplots show P_1, P_2, P_3, and P_4 from top down.

model follows the magnitude curve much better. This issue will be discussed further in Section V.

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 $\boldsymbol{x_0} = \boldsymbol{0}$ 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.

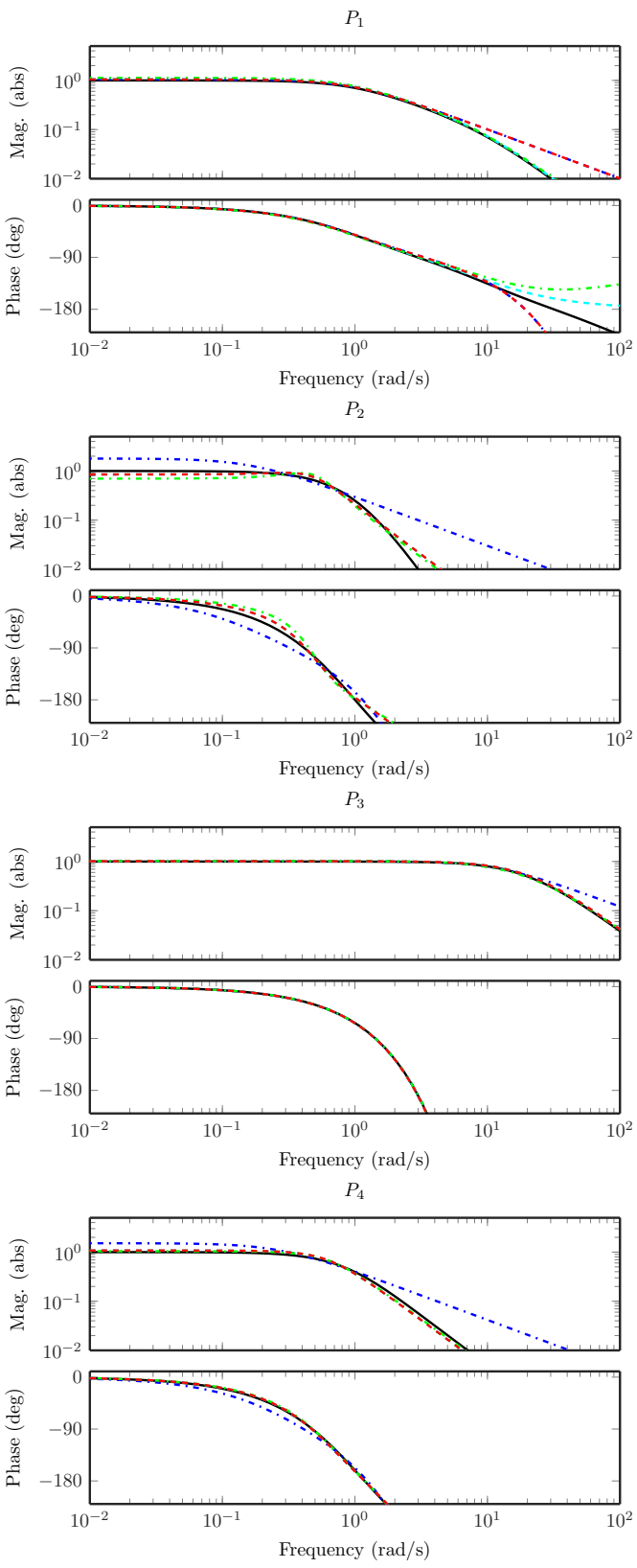

Fig. 4: Bode plots of the processes and identified models. The true process is shown in solid black, the FOTD model is shown in dashed-dotted blue, the SOTD model in dashed cyan, and the SOTDZ model in dashed-dotted green. The best model according to AIC is shown in dashed red.

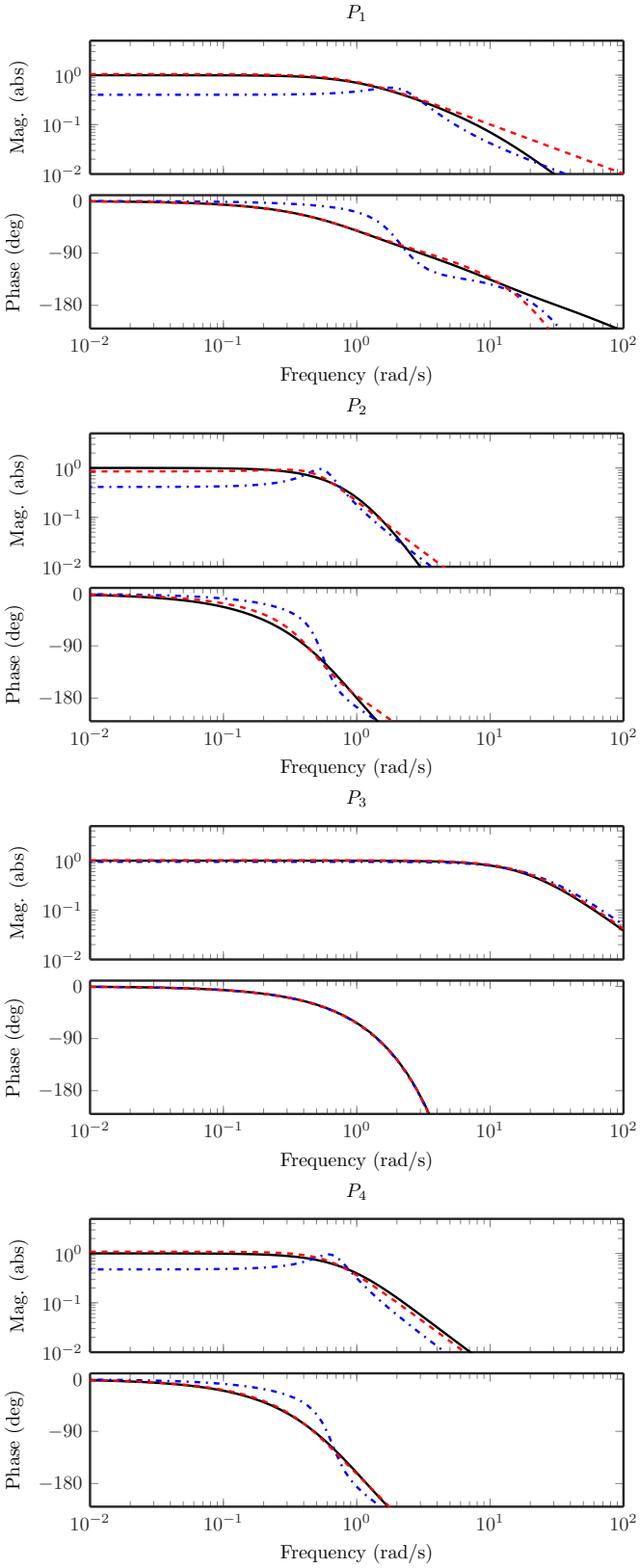

Fig. 5: Bode plots of obtained models for the example processes. The estimation of the initial state(s) $\boldsymbol{x_0}$ was active in the red dashed model and inactive in the blue dashed-dotted model. The true process is shown in solid black.

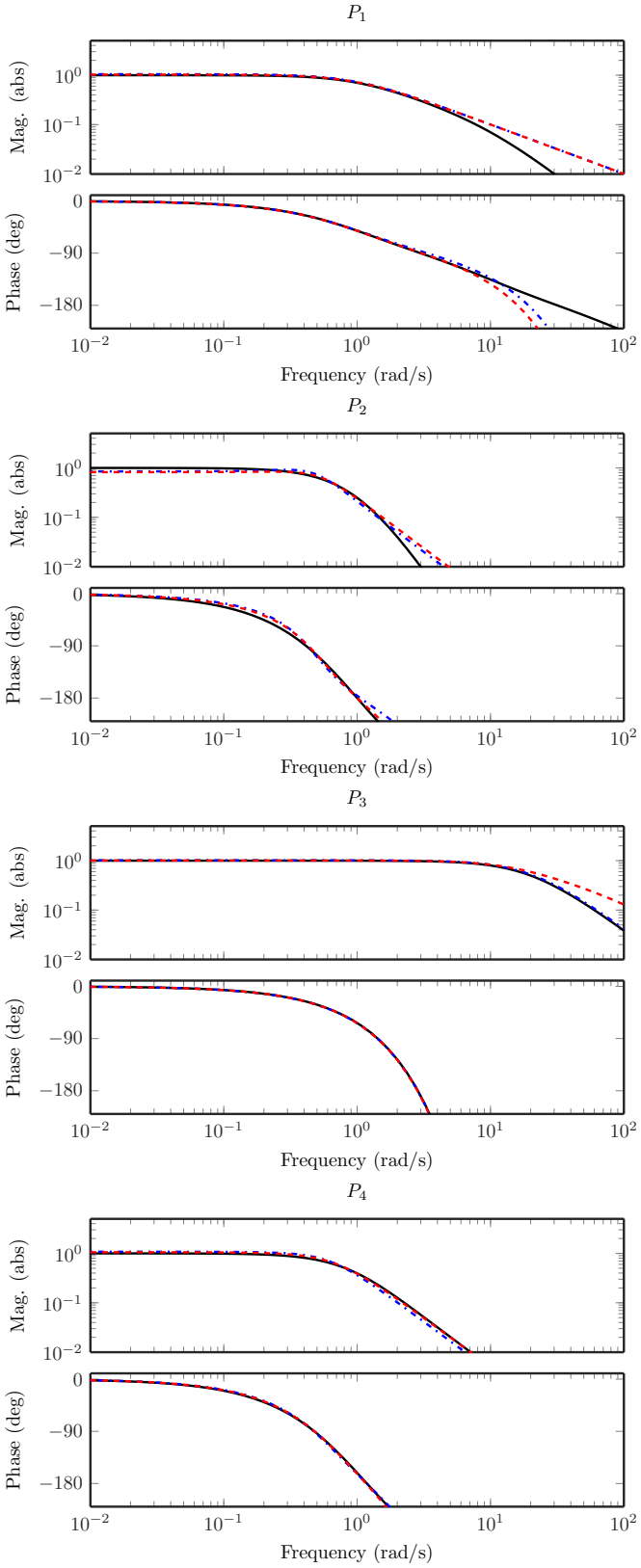

Fig. 6: Bode plots of the obtained models for the example processes. The best model from an experiment of 3 switches is shown in dashed-dotted blue, and 5 switches shown in dashed red. The true process is shown in solid black.

V. 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 [1] and [15], 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.

VI. 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

[1] K. Åström and T. Hägglund, *Advanced PID Control*. ISA - The Instrumentation, Systems, and Automation Society; Research Triangle Park, NC 27709, 2006.

[2] J. Ziegler and N. Nichols, "Optimum Settings for Automatic Controllers," *trans. ASME*, vol. 64, no. 11, 1942.

[3] O. Garpinger and T. Hägglund, "A Software Tool for Robust PID Design," in *Proc. 17th IFAC World Congr. Seoul, Korea*, 2008.

[4] K. Åström and T. Hägglund, "Automatic tuning of simple regulators with specifications on phase and amplitude margins," *Automatica*, vol. 20, no. 5, pp. 645–651, 1984.

[5] I. Kaya and D. Atherton, "Parameter estimation from relay autotuning with asymmetric limit cycle data," *J. Process Control*, vol. 11, no. 4, pp. 429–439, Aug. 2001.

[6] C. Lin, Q.-G. Wang, and T. Lee, "Relay Feedback: A Complete Analysis for First-Order Systems," *Ind. Eng. Chem. Res.*, vol. 43, no. 26, pp. 8400–8402, Dec. 2004.

[7] W. Luyben, "Derivation of transfer functions for highly nonlinear distillation columns," *Ind. Eng. Chem. Res.*, vol. 26, no. 12, pp. 2490–2495, Dec. 1987.

[8] K. Soltesz, P. Mercader, and A. Baños, "An automatic tuner with short experiment and probabilistic plant parameterization," *Int. J. of Robust and Nonlinear Control*, 2016.

[9] J. Berner, T. Hägglund, and K. Åström, "Asymmetric relay autotuning–Practical features for industrial use," *Contr. Eng. Prac.*, vol. 54, pp. 231–245, 2016.

[10] R. Byrd, J. Gilbert, and J. Nocedal, "A trust region method based on interior point techniques for nonlinear programming," *Mathematical Programming*, vol. 89, no. 1, pp. 149–185, 2000.

[11] K. Åström, "Maximum likelihood and prediction error methods," *Automatica*, vol. 16, no. 5, pp. 551–574, 1980.

[12] K. Soltesz, T. Hägglund, and K. Åström, "Transfer function parameter identification by modified relay feedback," in *2010 American Control Conference*, Baltimore, MD, USA, 2010.

[13] K. Åström and T. Bohlin, "Numerical identification of linear dynamic systems from normal operating conditions," IBM Nordic Laboratory, Tech. Rep. TP18.159, 1967.

[14] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *J. of Machine Learning Research*, vol. 13, no. Feb, pp. 281–305, 2012.

[15] J. Berner, T. Hägglund, and K. Åström, "Improved relay autotuning using normalized time delay," in *2016 American Control Conference (ACC)*. IEEE, 2016, pp. 1869–1875.

[16] H. Akaike, "A new look at the statistical model identification," *IEEE transactions on automatic control*, vol. 19, no. 6, pp. 716–723, 1974.