Improved Relay Autotuning using Normalized Time Delay

Berner, Josefin; Hägglund, Tore; Åström, Karl Johan

Published in:
American Control Conference (ACC), 2016

DOI:
10.1109/ACC.2016.7525191

2016

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

Link to publication

Citation for published version (APA):
Berner, J., Hägglund, T., & Åström, K. J. (2016). Improved Relay Autotuning using Normalized Time Delay. In *American Control Conference (ACC), 2016* https://doi.org/10.1109/ACC.2016.7525191

Total number of authors:
3

Improved Relay Autotuning using Normalized Time Delay*

Josefin Berner[1], 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.**

I. 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 [1], 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 [2], where the static gain was assumed to be known and in [3], 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 [4], and later on investigated in e.g. [5], [6] and [7]. For a more thorough review of the advances in modeling from relay feedback experiments, see [8].

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.

II. BACKGROUND

A. PID Tuning

There are many methods for tuning of PID controllers, ranging from the classic rules proposed in [9], to advanced optimization programs. Examples of existing tuning rules based on a low-order model of the process are λ-tuning [10], the SIMC [11], [12] and AMIGO [13]. 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

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

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 [14], where IAE is minimized with constraints on M_{ST}, are the two methods used. Modification to another tuning method is straightforward.

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

[1] All authors are with Department of Automatic Control, Lund University, Sweden. Contact `josefin.berner@control.lth.se`

B. 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.

C. 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 [13] 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. This since 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 Sec. VI.

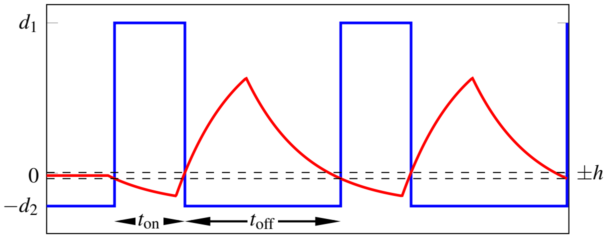

Fig. 1. An example of the signals from the asymmetric relay feedback experiment. The relay output u is shown in blue, the process output y is shown in red. The black dashed lines show the hysteresis levels, $\pm h$. The relay output switches between u_{on} and u_{off} every time the process output leaves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

The idea of using information from τ in a relay autotuning procedure is not new. In [15], 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 Sec. IV.

III. 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

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

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

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

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 Fig. 1. The half-periods t_{on} and t_{off} are defined as the time intervals where $u(t) = u_{\text{on}}$ and $u(t) = u_{\text{off}}$ respectively.

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

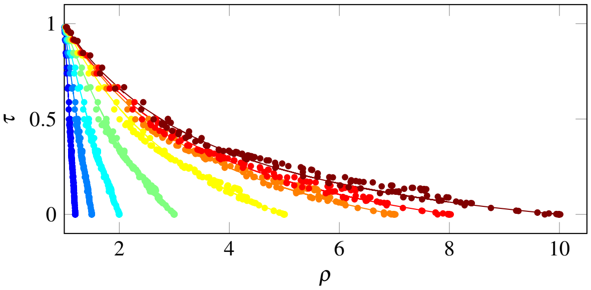

Fig. 2. Validation results of the equation for τ, stated in (12). The figure shows the results for $\gamma = \{1.2, 1.5, 2, 3, 5, 7, 8, 10\}$ in different colors from left to right. The solid lines show τ-values calculated from (12), while the dots show the relation between ρ and the true τ-values for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

adaptive relay amplitude. Details about the implementation and parameter choices are found in [16].

IV. 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 [7]. 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. Fig. 2 shows simulation results for a test batch [13] 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 Fig. 2.

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

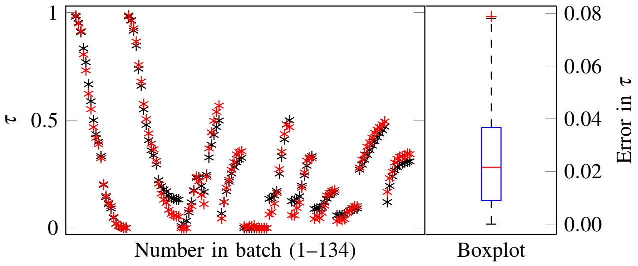

Fig. 3. Results of τ-estimation for the processes in the test batch. The left plot shows estimated τ in red, and the true values in black. The right plot shows a boxplot of the absolute errors of the τ-estimation.

V. MODELING

Once the experiment is performed we want to find the parameter values for the model structures listed in Sec. II-B. Modeling from an asymmetric relay experiment can be done in many ways. Some examples are by using the describing function as in [4], by using the A-locus method as in [5], by using the relation between the Fourier series coefficients and the model parameters as in e.g. [17] or by using a curve fitting approach as in e.g. [18]. For additional relevant references on different modeling strategies, see [8].

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 Sec. IV, 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

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

A. 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

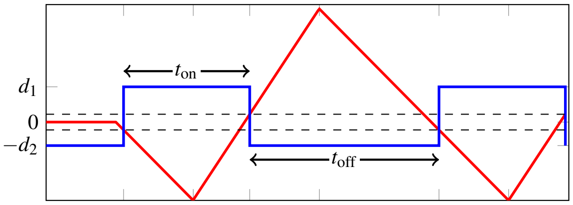

Fig. 4. An example of the signals from a relay experiment with an ITD process. The blue line shows the relay output u, the red line shows the process output y. The dashed black lines show the hysteresis. Note the triangular shape of y that is characteristic for an ITD process.

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}

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

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

given in [16]. 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 Sec. IV, 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).

B. 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 Fig. 4. By considering the output curves, equations for k_v and L can be obtained, see [16] for full derivation. The equations are

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

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

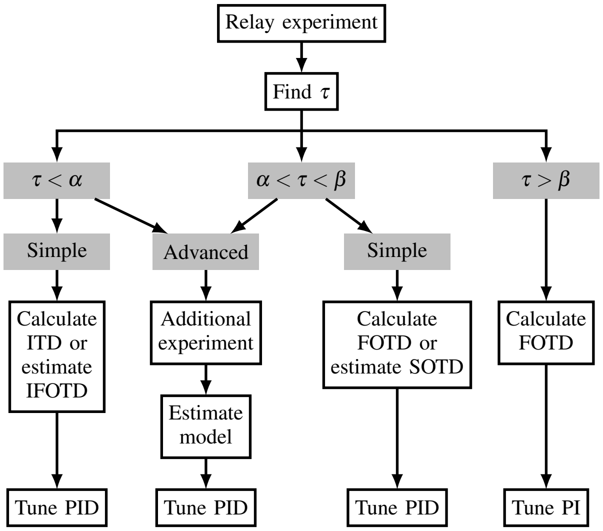

Fig. 5. Decision scheme based on the estimated normalized time delay.

C. 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 [7]. To assure convergence in the iterative method, appropriate initial parameter values are needed. These are obtained from the calculated FOTD or ITD model.

VI. Tuning Procedure

A. 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 Sec. II-B. The choice of model structure is based on the normalized time delay, τ. The resulting decision scheme is shown in Fig. 5. If τ is close to one, it has been shown [13], 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 Fig. 5, are $\alpha = 0.1$ and $\beta = 0.6$.

The low-order models defined in Sec. II-B are obtained as described in Sec. V. 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 Fig. 5. Information from the relay experiment already performed, can be used to design the additional experiment.

B. Controller Design

The choice of controller design is restricted to PID controllers. The low-order models in Sec. II-B were chosen since

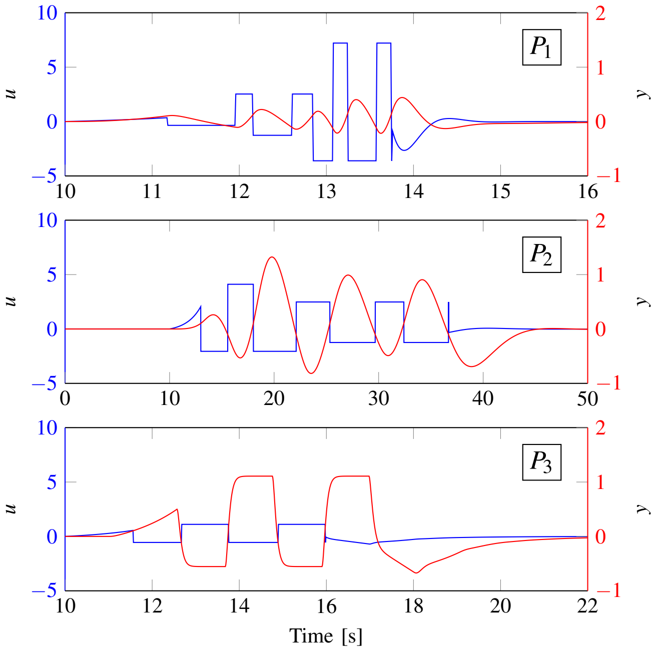

Fig. 6. Signals from the relay experiment for P_1 (top), P_2 (middle) and P_3 (bottom). The blue lines show the relay output u, and the red lines show the process output y. Note the different scales on the axes.

there exists simple tuning rules for them. This implementation of the autotuner uses the AMIGO tuning rules [13], 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 [14].

In [13], 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 Fig. 5.

VII. 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 [16].

The experiment output for the three processes are shown in Fig. 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 Fig. 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 Tab. I.

TABLE I

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

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 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 [14] where IAE is minimized with the constraint that $M_{ST} \leq 1.4$. The obtained controller parameters, performance and robustness measures are listed in Tab. II. Control performances for a step in load disturbance are illustrated in Fig. 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 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.

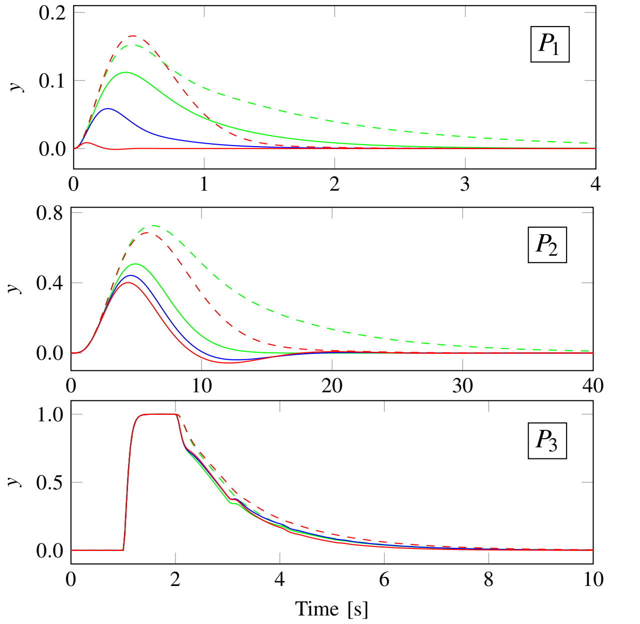

Fig. 7. Step responses from a load disturbance on the process input, shown for the closed-loop systems containing the true processes and the different obtained controllers. The green curves show controllers tuned for the ITD/FOTD models, blue shows controllers tuned for IFOTD/SOTD models, red shows controllers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

TABLE II

CONTROLLER PARAMETERS.

	Controller	K	T_i	T_d	M_{ST}	IAE
P_1	ITD PI	5.48	1.18		1.34	0.215
	ITD PID	7.04	0.70	0.04	1.15	0.100
	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
P_2	FOTD PI	0.36	3.02		1.24	8.487
	FOTD PID	0.98	2.85	0.80	1.35	2.906
	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
P_3	FOTD PI	0.17	0.37		1.44	2.158
	FOTD PID	0.24	0.48	0.11	1.41	2.014
	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

VIII. 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 full speed of the fan, while the pressure was measured in Pascal.

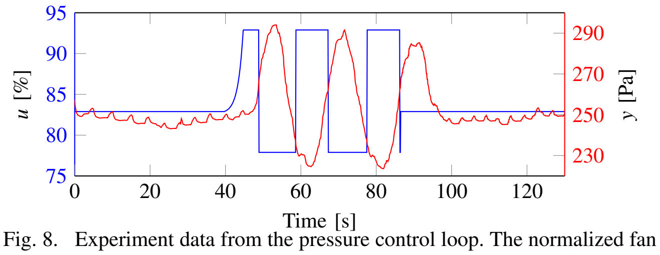

Fig. 8. Experiment data from the pressure control loop. The normalized fan speed is shown in blue, the pressure measurement in red. Note the different scales and units on the axes.

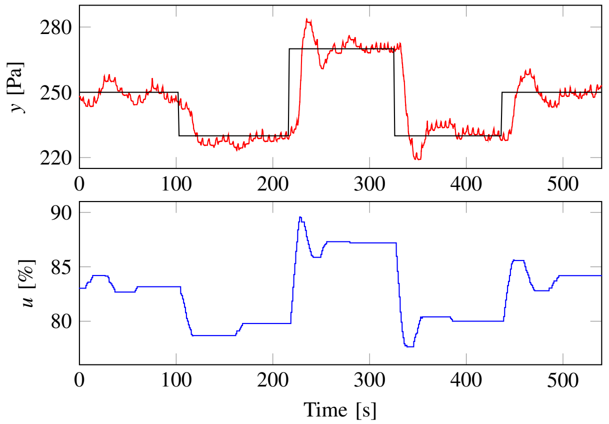

Fig. 9. Response to setpoint changes for the system with the controller tuned from the experiment. The upper plot shows the measured pressure in red, and the setpoint in black. The lower plot shows the control signal.

The reference value of the pressure was set to 250 Pa. A relay experiment performed on the system is shown in Fig. 8. The asymmetry level, convergence limit, maximum and minimum deviations and the maximum relay deviation, were set according to the default values in [16]. 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 Sec. V, 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 Fig. 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 Fig. 10. This also shows satisfactory results. The

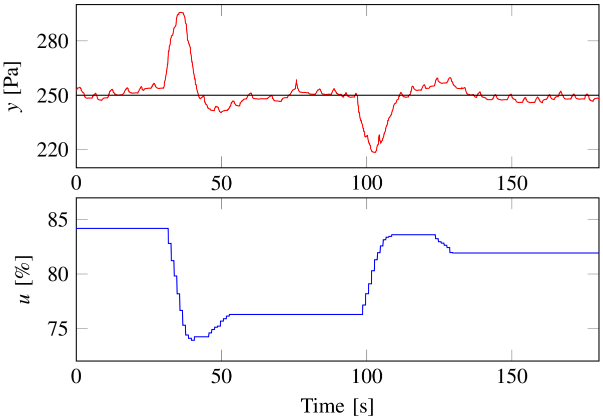

Fig. 10. Response to load disturbances. The upper plot shows the measured pressure in red, and the setpoint in black. The lower plot shows the control signal.

effect of the load disturbances is removed completely in approximately 20-25 s with rather small overshoots.

IX. 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.

REFERENCES

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

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

[3] W. Li, E. Eskinat, and W. L. Luyben, "An improved autotune identification method," *Ind. Eng. Chem. Res.*, vol. 30, no. 7, pp. 1530–1541, July 1991.

[4] S.-H. Shen, J.-S. Wu, and C.-C. Yu, "Use of biased-relay feedback for system identification," *AIChE J.*, vol. 42, no. 4, pp. 1174–1180, Apr. 1996.

[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. H. Lee, "Relay Feedback: A Complete Analysis for First-Order Systems," *Ind. Eng. Chem. Res.*, vol. 43, no. 26, pp. 8400–8402, Dec. 2004.

[7] J. Berner, K. J. Åström, and T. Hägglund, "Towards a New Generation of Relay Autotuners," in *19th IFAC World Congr.*, 2014.

[8] T. Liu, Q.-G. Wang, and H.-P. Huang, "A tutorial review on process identification from step or relay feedback test," *J. Process Control*, vol. 23, no. 10, pp. 1597–1623, Nov. 2013.

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

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

[11] S. Skogestad, "Simple analytic rules for model reduction and PID controller tuning," *J. Process Control*, vol. 13, no. 4, pp. 291–309, June 2003.

[12] ——, "Tuning for smooth PID control with acceptable disturbance rejection," *Ind. & Eng. Chem. Res.*, vol. 45, no. 23, pp. 7817–7822, 2006.

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

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

[15] W. L. Luyben, "Getting More Information from Relay-Feedback Tests," *Ind. Eng. Chem. Res.*, vol. 40, no. 20, pp. 4391–4402, Oct. 2001.

[16] J. Berner, "Automatic Tuning of PID Controllers based on Asymmetric Relay Feedback," Dept. Automatic Control, Lund University, Sweden, Licentiate Thesis ISRN LUTFD2/TFRT--3267--SE, May 2015.

[17] K. Srinivasan and M. Chidambaram, "Modified relay feedback method for improved system identification," *Computers & chemical engineering*, vol. 27, no. 5, pp. 727–732, 2003.

[18] T. Liu and F. Gao, "Alternative Identification Algorithms for Obtaining a First-Order Stable/Unstable Process Model from a Single Relay Feedback Test," *Ind. Eng. Chem. Res.*, vol. 47, no. 4, pp. 1140–1149, Feb. 2008.