Model optimization for autotuners in industrial control systems

Lundh, Magnus; Theorin, Alfred; Hägglund, Tore; Hansson, Jonas; Svensson, Magnus; Åström, Karl Johan; Soltesz, Kristian

Published in:
Proceedings of the 26th International Conference on Emerging Technologies and Factory Automation (ETFA)

DOI:
10.1109/ETFA45728.2021.9613187

2021

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

Model optimization for autotuners in industrial control systems

Magnus Lundh, Alfred Theorin, Tore Hägglund, Jonas Hansson, Magnus Svensson,
Karl Johan Åström, Kristian Soltesz

***Abstract*— Automatic tuning of PID controllers using relay feedback experiments has received attention on and off since it was first proposed and industrially implemented in a control system in the 1980s. While optimal experiment design and modern system identification easily outperform the original automatic tuner, they rely on computational resources that are not always available in industrial control systems. Here we present a combination of experiment and subsequent output-error identification of continuous-time first-order time-delayed (FOTD) system models, that requires very little in terms of computations and memory. The method has been extensively evaluated in simulation, and a prototype has been implemented for the ABB AC 800M controller family.**

I. Introduction

A. Background

The idea to use relay feedback to obtain a system response that self-oscillates at the critical $-180°$ phase shift angular frequency of the process dynamics was first presented in [1] and lay the ground for automatic PID tuning, as originally implemented in the NAF SDM20 system, and later also in systems by several other major vendors. The main strength of the original auto-tuner is that its experiment automatically detects the critical freqency ω_0 of the process dynamics. Its main weaknesses are that a relatively long experiment is required for convergence to a steady limit-cycle at ω_0, and the fact that the process model provided by the original autotuner comprised only of ω_0 and $|G(i\omega_0)|$.

Numerous extensions and improvements have been proposed, as exemplified by *e.g.* [2]–[6]. We have previously shown [7]–[9] how the combination of a short asymmetric relay experiment and optimization techniques outperform the original relay autotuner [1]. However, such improvements have not yet made it into the product lines of major vendors. We believe the relatively high complexity of the optimization-based model identification to be the main reason for this situation.

We would like to acknowledge Josefin Berner for prior work. LU-affiliated authors are members of the ELLIIT Strategic Research Area at Lund University. JH is supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

ML and AT are with ABB, Malmö, Sweden. Correspondence: alfred.theorin@se.abb.com

TH, JH, KJÅ and KS are with Dept. Automatic Control, Lund University, Sweden.

MS is with Dept. Energy Sciences, Lund University, Sweden.

B. The FOTD process model

The first-order time-delayed (FOTD) system

$$G(s) = \frac{K}{sT+1}e^{-sL} \tag{1}$$

is sufficient for modelling a vast majority of industrial processes, where the purpose of modelling is PID control. The phase lag introduced through the delay L makes the FOTD model a good approximation of higher-order dynamics in the phase-range of interest for PID control, and the normalized time-delay $\tau = L/(L+T)$ provides an important characterization of the dynamics. A small $\tau \gtrsim 0$ means that the dynamics are dominated by the lag $T \gg L$; a large $\tau \lesssim 1$ means that the dynamics are dominated by the delay $L \gg T$. Note also that (1) can be used to model integrating processes if $T \gg 1$.

C. Output error identification

If FOTD dynamics (1) are to be identified from a pair of time-aligned and equi-temporally spaced input samples $u = [u_1 \ \dots \ u_n]^\top$ and corresponding measurement samples $y = [y_1 \ \dots \ y_n]^\top$, where the measurements are corrupted by additive, uncorrelated and identically distributed Gaussian noise (a common noise model for many sensors), it is well-known that the FOTD model with output $\hat{y}$ that minimizes the output error $\|y - \hat{y}\|_2^2$ is optimal in the maximum likelihood sense.

Here we show how this model can be estimated from asymmetric relay experiment data in a lightweight way, that does not require software libraries for linear algebra, optimization, *etc.* This makes our approach readily applicable to industrial implementation, and we are currently evaluating it within the ABB AC 800M controller family.

II. Method

A. Experiment

A relay experiment, as the one shown in Fig. 1, is conducted using an asymmetric relay as described in [9], with two relay switches and no chirp. Since the relay automatically matches experiment duration to the process time scale, a dynamic buffer of length $128 \leq n \leq 256$ was used. If the buffer got full before the experiment terminated, half of its samples were discarded and the sampling period thus doubled.

B. Loss function evaluation

The objective is to find parameters $\theta = [\hat{K} \quad \hat{T} \quad \hat{L}]^\top$ of an FOTD model $\hat{G}$ that minimizes the output error norm

$$V(\theta|u,y) = \|y - \hat{y}(\theta)\|_2^2 = (y-\hat{y})^\top (y-\hat{y}). \tag{2}$$

To do so, we need to evaluate $\hat{y}(\theta)$ at different candidates θ. Since the control signal u is zero-order hold (ZOH) sampled in industrial control systems, we can use the exact ZOH discretization of (1) to simulate candidate systems. If the sampling period is h, we need to ZOH discretize the system

$$\tilde{G}(s) = \frac{K}{sT+1} e^{-\tilde{L}s}, \tag{3}$$

where $\tilde{L} = \text{mod}(L,h)$, and then delay its simulated output $d = \lfloor L/h \rfloor$ samples. ZOH sampling of the system with delay $\tilde{L}$ is conducted by writing (1) on state-space form (where subscript c denotes continuous time)

$$\dot{\tilde{y}}_c(t) = \underbrace{-\frac{1}{T}}_{A} \tilde{y}_c(t) + \underbrace{\frac{K}{T}}_{B} u_c(t-\tilde{L}), \tag{4}$$

and solving for

$$\tilde{y}_c(h) = \underbrace{e^{Ah}}_{\Phi} y_c(0) + \underbrace{\int_0^{\tilde{L}} e^{At} dt}_{\Gamma_1} u_c(0) + \underbrace{\int_0^{h-\tilde{L}} e^{At} dt B}_{\Gamma_0} u_c(h). \tag{5}$$

Inserting expressions for A and B from (4) into (5) and evaluating the integrals gives

$$\Phi = e^{-h/T}, \tag{6a}$$
$$\Gamma_1 = K\Phi(\gamma - 1), \tag{6b}$$
$$\Gamma_0 = K(1 - \Phi\gamma), \tag{6c}$$

where $\gamma = e^{\tilde{L}/h}$. We finally apply a time-shift to obtain

$$y(k) = \Phi y(k-1) + \Gamma_1 u(k-2-d) + \Gamma_0 u(k-1-d), \tag{7}$$

which we can use to simulate $\hat{G}$ to obtain $\hat{y}$ and the loss (2).

Here we limit ourselves to asymptotically stable systems with positive stationary gain and non-zero delay, $\theta \succ 0$. The cases where $\hat{T} = 0$ or $\hat{L} = 0$ are practically indistinguishable from cases where the two parameters are small but non-zero. Explicit treatment of $\hat{T} = 0$ is nonetheless straightforward, but relies on coefficients that differ from (6).

C. Loss function minimization

The approach taken in [10] was to compute the gradient $\partial V/\partial\theta$ and perform local optimization in search of the minimizer θ^o. There are no guarantees that V is convex in θ and examples where it is not can indeed be devised. This means that the optimization could get stuck in a local minimum other than θ^o, unless cleverly initialized. The optimization itself also relies on the computation of the cost gradient and possibly the associated Hessian, which alongside line-search, requires software libraries.

One alternative to the local (gradient-based) approach is to use a global (gradient-free) approach comprising of simply gridding the parameter space, evaluating V at every grid-point, and reporting the grid point that minimizes V.

Here we do something in between—an iterative grid search—inspired by simple bisection search. A continuous scalar function f with $f(a) < 0$ and $f(b) > 0$ implies $f(x) = 0$ for some x between a and b. In bisection search, this is used to bisect the search interval by evaluating the sign of $f(a+b)/2$ and discarding either half of the original interval based on it. This procedure is then repeated until desired accuracy has been obtained.

Here, we are not looking for zero-crossing of $V(\theta)$, but instead of its gradient $\partial V/\partial\theta$. The argument θ is not scalar, and furthermore, we cannot rule out the presence of multiple local minima of V, translating into equally many zero-crossings of $\partial V/\partial\theta$.

What we have done is to consider the marginal loss $V(\hat{\tau}|L,K)$. This is the loss associated with the free parameter $\hat{\tau} = \hat{L}/(\hat{L}+\hat{T})$, provided that $\hat{L} = L$ and $\hat{K} = K$. We hypothesize that this marginal loss is *almost* convex in the sense that any local minima are shallow compared to the global minimum $V(\tau, L, K)$ and partition the closed interval $[0,1]$ into $m-1$ equally spaced sub-intervals using m grid points $\hat{\tau}_1 = 0, \ldots, \hat{\tau}_m = 1$, at which we evaluate V using the method of Sec. II-B. If $V(\hat{\tau}_k) > V(\hat{\tau}_{k+1}) < V(\hat{\tau}_{k+2})$, we know that the marginal loss has a (local) minimum between τ_k and τ_{k+2}, and can confine the next iteration to search $[\tau_k, \tau_{k+2}]$. If there are multiple local minima, the one with lowest marginal loss is chosen; if there are no local minima the interval can be reduced down to $[\hat{\tau}_1, \hat{\tau}_2]$ or $[\hat{\tau}_{m-1}, \hat{\tau}_m]$, based on whether the marginal cost is largest at $\hat{\tau}_1$ or $\hat{\tau}_m$.

The same type of marginalization is also conducted with respect to $\hat{L}$, resulting in a nested identification algorithm, while optimization of $\hat{K}$ is explicitly handled as explained under Sec. II-D.1. The nested optimization algorithm can thus be summarized:

- perform an iterative grid search for $\hat{\tau}$, starting with $[\hat{\tau}_1 = 0, \hat{\tau}_m = 1]$;
- for each candidate $\hat{\tau}$, perform an iterative grid search for $\hat{L}$ staring with $[\hat{L}_1 = 0, \hat{L}_m = L_{\max}]$;
- optimize $\hat{K}$ at each candidate pair $\hat{\tau}, \hat{L}$.

D. Implementation aspects

1) Optimizing the gain parameter: Since $\hat{y}$ is linear in $\hat{K}$, finding the optimal $\hat{K}$ for any candidate pair $\hat{\tau}, \hat{L}$ is done through minimizing

$$V(\hat{K}|\hat{\tau},\hat{L}) = (y - \hat{K}\hat{y})^\top (y - \hat{K}\hat{y}) = y^\top y - 2\hat{K} y^\top \hat{y} + \hat{K}^2 \hat{y}^\top \hat{y}, \tag{8}$$

where $\hat{y}$ is the output of the model with $\hat{K} = 1$. The quadratic form (8) is minimized by $\hat{K}^o|\hat{\tau}, \hat{L} = \hat{y}^\top y/(\hat{y}^\top \hat{y})$.

2) Grid density: The number of grid points, $m(i)$, in iteration i of the grid search constitutes a trade-off between speed and safeguard against local minima. For $m \geq 4$ we discard $m-3$ or $m-2$ of the $m-1$ sub-intervals between neighbouring grid points. The worst-case for interval length reduction per loss evaluation is thus $g(m) = (m-3)/(m(m-1))$. Solving $dg/dx = 0$ and discretely maximizing g at all m with $|m-x| < 1$ suggests the use of either $m = 5$ or $m = 6$ grid points.

If we store evaluated V from iteration i to iteration $i+1$, the values of V at the new interval end-points do not need to be re-evaluated. Additionally, if m is odd, the mid-point can (sometimes) be re-used. The worst-case for interval length reduction per loss evaluation is then $g(m) = (m-3)/((m-1)(m-2-\mathrm{mod}\,(m,2)))$, which is maximized for $m = 5$. We therefore choose $m = 5$ grid points for all iterations.

3) Termination criterion: Since one or two sub-intervals are kept between consecutive iterations, N iterations provide a relative accuracy σ satisfying

$$\left(\frac{1}{m-1}\right)^N \leq \sigma \leq \left(\frac{2}{m-1}\right)^N. \tag{9}$$

The iterations required for a relative accuracy σ is thus

$$\left\lceil -\frac{\log\sigma}{\log(m-1)} \right\rceil \leq N \leq \left\lceil \frac{\log\sigma}{\log 2 - \log(m-1)} \right\rceil. \tag{10}$$

4) Iterative computations: There is no need to store the entire simulation output $\hat{y}$ to evaluate either of the loss V or to optimize the static gain $\hat{K}$. The loss can be written as the scalar product $V(\hat{y}) = (y-\hat{y})^\top(y-\hat{y})$. It can thus be computed iteratively as $V(k) = V(k-1) + (y_k - \hat{y}_k)^2$, with $V(0) = 0$, and eventually yielding $V = V(n)$. Similarly, the expression (8) only depends on similar scalar products, and it is sufficient to iteratively evaluate $y^\top\hat{y}$, $y^\top y$, and $\hat{y}^\top\hat{y}$.

III. Results

We evaluated the proposed identification algorithm across a previously published [11] batch of 134 well-damped models representative for industrial processes.

The method was executed using $m = 5$ grid points in all iterations, and terminated once a relative accuracy of $\sigma = 0.01$ had been reached, resulting in $4 \leq N \leq 7$ iterations in accordance with (10).

In interest of space, we have limited the presentation here to include only a few representative results[1]. The true (unknown) dynamics for these examples are an FOTD process (11a); an integrating process (11b); a process with higher-order dynamics (11c):

$$G_1(s) = \frac{1}{10s+1}e^{-s}, \tag{11a}$$

$$G_2(s) = \frac{0.1}{s(0.1s+1)}e^{-0.9s}, \tag{11b}$$

$$G_3(s) = \frac{1}{(s+1)(0.5s+1)(0.25s+1)(0.125s+1)}. \tag{11c}$$

The outcome of the relay experiments and output of the obtained FOTD models are shown in Fig. 1–3. The noise-corrupted output y resulting from the input u is shown in grey; the noise-free (unknown) output is shown in blue; the output of the identified model is shown in red. (Both u and y denote deviations from a stationary working point $[u^0,\ y^0] = [0\ 0]$, explaining their occasionally negative values.) The dashed black lines indicate the relay hysteresis level, explained in [1].

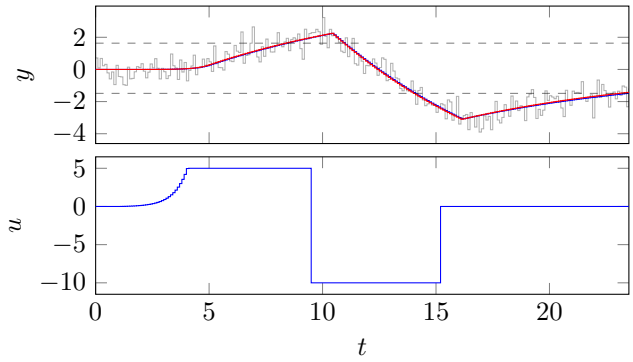

Fig. 1. Experiment and model fit for the FOTD process (11a).

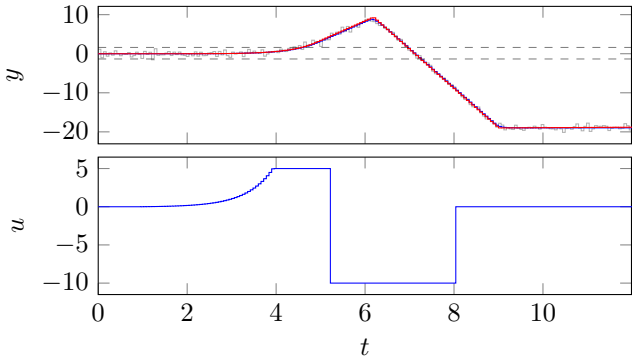

Fig. 2. Experiment and model fit for the integrating process (11b).

Identifying FOTD models this way involved median (min, max) 236 (193, 238) unique simulations (loss evaluations) per experiment, to be compared with 10^4 unique simulations required to obtain the same accuracy using dense gridding. For each model, we performed the expensive grid computation, as illustrated in Fig. 4, to verify that $m = 5$ is sufficient when identifying representative process industrial dynamics.

Good agreement between $\hat{y}$ and y results in a small loss V, but not necessarily in a useful model. For PID tuning purposes, the latter additionally requires a sufficiently high signal-to-noise ratio close to ω_0, necessary

[1]Additional results and further description of the methodology are found in [12].

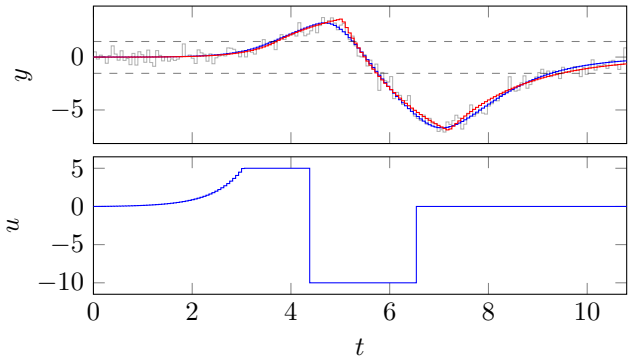

Fig. 3. Experiment and model fit for the higher-order process (11c).

for good model fit close to the $-180°$ phase shift of the process dynamics. To investigate this, we also plotted the Bode diagrams of the process model dynamics for each batch process, together with that of the identified FOTD model. A representative example is shown in Fig. 5 and all plots are available in [12].

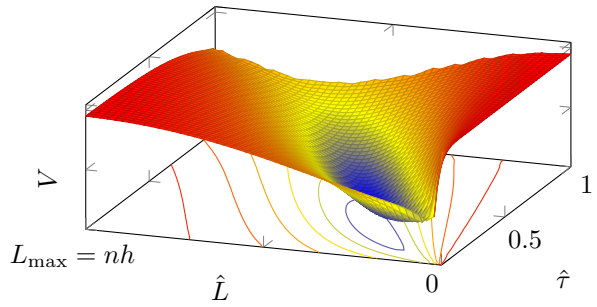

Fig. 4. Representative loss $V(\hat{\tau}, \hat{L})$ with $\hat{K}$ optimized to minimize V at each grid point.

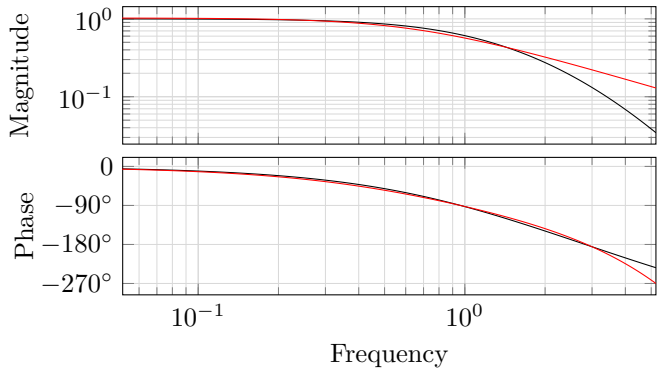

Fig. 5. Bode diagram for the higher-order process (11c) (black) and identified model (red).

IV. Discussion

We have demonstrated how output error identification of continuous-time FOTD models from asymmetric relay experiment data can be conducted in a lightweight manner. The method requires no software libraries, and is straightforward to implement within an embedded industrial controller—even one with very limited computational power and memory. The use of a very short experiment reduces the risk of the experiment being corrupted by severe sporadic load disturbances.

The obtained FOTD models can be used for PID controller tuning, but also for other purposes. The method was initially implemented in MATLAB, where it has undergone thorough investigation and validation. A prototype for the ABB AC 800M industrial controller family has also been implemented and successfully evaluated in combination with existing PID tuning rules, and thus proven to be a capable autotuner suitable for implementation in industrial control systems.

References

[1] 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. DOI: 10.1016/0005-1098(84)90014-1.

[2] M. Friman and K. Waller, "A two-channel relay for autotuning," *Industrial and Engineering Chemistry Research*, vol. 36, no. 7, pp. 2662–2671, 1997. DOI: 10.1021/ie970013u.

[3] I. Kaya and D. Atherton, "Parameter estimation from relay autotuning with asymmetric limit cycle data," *Journal of process control*, vol. 11, no. 4, pp. 429–439, 2001. DOI: 10.1016/S0959-1524(99)00073-6.

[4] R. De Keyser, C. Muresan, and C. Ionescu, "A novel auto-tuning method for fractional order PI/PD controllers," *ISA Transactions*, vol. 62, pp. 268–275, 2016. DOI: 10.1016/j.isatra.2016.01.021.

[5] W. Li, E. Eskinat, and W. Luyben, "An improved autotune identification method," *Industrial and Engineering Chemistry Research*, vol. 30, no. 7, pp. 1530–1541, 1991. DOI: 10.1021/ie00055a019.

[6] S. Shen, J. Wu, and C. Yu, "Use of biased-relay feedback for system identification," *AIChE Journal*, vol. 42, no. 4, pp. 1174–1180, 1996. DOI: 10.1002/aic.690420431.

[7] K. Soltesz and T. Hägglund, "Extending the relay feedback experiment," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 13 173–13 178, 2011. DOI: 10.3182/20110828-6-IT-1002.00066, Presented at IFAC World Congress, Milan, Italy.

[8] J. Berner, K. Soltesz, T. Hägglund, *et al.*, "An experimental comparison of PID autotuners," *Control Engineering Practice*, vol. 73, pp. 124–133, 2018. DOI: 10.1016/j.conengprac.2018.01.006.

[9] J. Hansson, M. Svensson, A. Theorin, *et al.*, "Next generation relay autotuners: Analysis and implementation," in *Proceedings of the IEEE conference on control technology and applications (CCTA)*, Accepted manuscript, San Diego, CA, 2021.

[10] J. Berner and K. Soltesz, "Short and robust experiments in relay autotuners," in *Proceedings of the 21st IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Limassol, Cyprus, 2017. DOI: 10.1109/ETFA.2017.8247696.

[11] T. Hägglund and K. Åström, "Revisiting the Ziegler-Nichols step response method for PID control," *Journal of Process Control*, vol. 14, no. 6, pp. 635–650, 2004. DOI: 10.1016/j.jprocont.2004.01.002.

[12] M. Lundh, *A new, fast, and efficient automatic tuner for the ABB AC 800M family of controllers*, 2021. [Online]. Available: https://lup.lub.lu.se/student-papers/search/publication/9061708.