Journal of Technology and Exploitation in Mechanical Engineering
Vol. 2, no. 1, pp. 26–31, 2016
https://doi.org/10.35784/jteme.413 Research article

Submitted: 2016.11.11
Accepted: 2016.12.12
Published: 2016.12.26

AUTOTUNING OF PID CONTROLLER BY MEANS OF HUMAN MACHINE INTERFACE DEVICE

Michał Awtoniuk[1], Łukasz Ślizak[2], Marcin Daniun[3]

[1] Warsaw University of Life Sciences, Faculty of Production Engineering, email: michal_awtoniuk@sggw.pl
[2] Pro-face Poland, email: lukasz.slizak@proface.pl
[3] Warsaw University of Life Sciences, Faculty of Production Engineering, email: marcin.daniun@gmail.com

ABSTRACT
More and more control systems are based on industry microprocessors like PLC controllers (*Programmable Logic Controller*). The most commonly used control algorithm is PID (*Proportional-Integral-Derivative*) algorithm. Autotuning procedure is not available in every PLC. These controllers are typically used in cooperation with HMI (*Human Machine Interface*) devices. In the study two procedures of autotuning of the PID controller were implemented in the HMI device: step method and relay method. Six tuning rules for step methods and one for relay method were chosen. The autotuning procedures on simulated controlled object and PLC controller without build-in autotuning were tested. The object of control was first order system plus time delay.
KEYWORDS: PLC controller, autotuning, PID, HMIdevice

AUTOMATYCZNE STROJENIE REGULATORA PID ZA POMOCĄ PANELA DOTYKOWEGO HUMAN MACHINE INTERFACE

STRESZCZENIE
Coraz więcej układów sterowania opartych jest na przemysłowych mikroprocesorach takich jak sterowniki PLC (*Programmable Logic Controller*). W procesach regulacji najczęściej wykorzystywanym algorytmem jest algorytm PID (*Proporcjonalno-Całkująco-Różniczkujący*). Jednak nie w każdym sterowniku PLC dostępna jest procedura autotuningu. Zazwyczaj sterowniki PLC są wykorzystywane w połączeniu z panelami operatorskimi HMI (*Human Machine Interface*). W celu przeprowadzeniu badań zaimplementowano w panelu operatorskim HMI dwie metody automatycznego strojenia regulatora PID. Wykorzystano metodę skoku jednostkowego oraz przekaźnikową. Wybrano sześć reguł doboru nastaw dla metody skokowej oraz jedną dla przekaźnikowej. Procedurę autotuningu przetestowano z symulowanym komputerowo obiektem regulacji oraz sterownikiem PLC bez wbudowanej procedury autotuningu. Obiektem regulacji był człon inercyjny I rzędu z opóźnieniem.
SŁOWA KLUCZOWE: sterownik PLC, strojenie, PID, panel HMI

1. Introduction

PID algorithm is the most widely used control algorithm in the industry. The record shows near 90% control systems employ PID (*Proportional-Integral-Derivative*) algorithm [1], [2]. PID algorithm implemented in industry microprocessors, e.g. PLC, is called PID controller [3]. Currently, the most popular industrial microprocessor-based devices are PLC (*Programmable Logic Controller*). Many controller manufacturers (e.g. ABB, Allen-Bradley, General Electric, Honeywell, Omron, Siemens, Toshiba, Yokogawa, etc.) offer a PID algorithm as function block in their devices. Some of the manufacturers decide to introduce series of improvements of PID algorithm. Typical examples include

techniques for mode switches and anti-windup [3], [4]. Despite the popularity of the PID algorithm, not every controller manufacturers provide procedure to tune the controller parameters, i.e. autotuning. The main task of autotuner is to identify the controlled object. The controlled object is often called the plant. The identification process reveals the dynamic and static behaviour of the plant. This behaviour determines the controller parameters. There are two major autotuning procedures implemented in PLC. These are relay and step methods. If there is no build-in autotuning procedure, the operator has to do the identification sand election of controller parameters by himself.

To facilitate the handling of control system of single machine or technological process touch operating panels HMI (*Human Machine Interface*) are used. On the Polish automation market, the most popular HMI devices are Pro-face panels. At the beginning, the HMI panels had only two tasks: to display information that came from PLC (e.g. the state of discrete output or temperature measurement) and to transmit input information to PLC (e.g. set the state of discrete output or setpoint temperature). Modern HMI panels, in addition to tasks related to process control, perform following functions: the gateway that makes communication between different automation devices easier, generates alarms based on events, sampling data, reporting and historical data collection. They can not only be complementary to the PLC, but also they can replace the PLC in the realization of some functionality. Controller tuning is a good example.

The goal of our study was to implement in the HMI the automatic procedure of controller tuning. The assumption was that the program would identify controlled object and calculated controller parameters on the basis of identification results.

2. Materials and methods

Fig. 1. presents scheme of data collection. The key elements used in research were:

- PLC controller GE VersaMax (IC200CPU001) with analogue module (IC200ALG432E),
- HMI panel Pro-face AGP-3600 T,
- Data acquisition board National Instruments USB-6229,
- Software Matlab&Simulink.

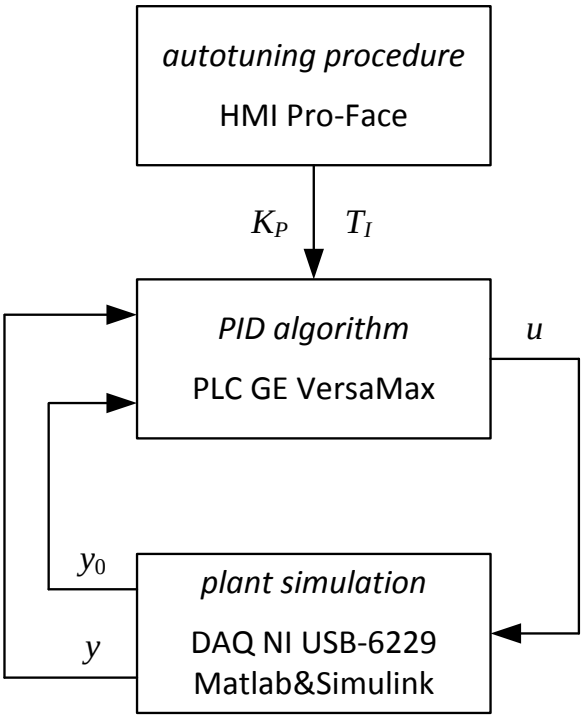

Fig. 1. Block diagram of data set collection

Controlled object, also called the plant, described by transfer function$G(s)$ (1) was simulated using Matlab&Simulink. It was the first order plus time delay system, which is typical example of thermal system [5]. It was assumed that value of parameters of controlled object were as follows: k=0,8, T=9s, and τ=3s.

$$G(s) = \frac{k}{Ts+1} e^{-\tau s} \tag{1}$$

where: k – plant proportional gain, T – time constant, τ – delay or lag time.

The PLC controller realized the PID algorithm described by equation (2). Which is the most commonly implemented form of PID. This form is labeled ISA, dependent or gain-dependent form. The ISA is used in numerous controllers, e.g., GE VersaMaxPLC (*PID ISA* function block), Siemens S7-300 (*FB41* function block), and Allen-Bradley PLC-5 (*PID* function block) [4], [6], [7]. In our research we use PID algorithm without derivative term, so T_D=0.

$$u = K_P \left(1 + \frac{1}{T_I s} + T_D s \right) \tag{2}$$

where: u – control signal, K_P – controller proportional gain, T_I – reset time, T_D – derivative time.

Matlab&Simulinkwas generating setpoint y_0 and plant output y signals, which were transmitted to the PLC through the DAQ board. On the basis of setpointy_0 and controlled object output y signals, the PLC was calculating the control signal u and sending it back to Matlab&Simulink.

The HMI panel supervised control process, displayed actual values of the measurements and archived those values in the internal memory. The autotuning procedure was initialized by operator through the HMI. We implemented two versions of procedure: relay and step.

Relay method, also labelled as sustained oscillation, was based on the procedure available at EFTRONIK XS controller [8]. The procedure had several stages:

- Operator (user) declares maximum u_{max} and minimum u_{min} permissible values of control signal u, setpointy_0, and time of experiment.
- Program switches controller to the manual mode.
- Program analyses value of plant output signal y. If $y \le y_0$ then u=u_{max}, else u=u_{min}.
- After identification stage, program determines two parameters: ultimate gain K_U and ultimate period T_U. These parameters are used to calculate the controller parameters according to *Aström-Hägglund* rule (3). Program sends parameters K_P and T_I to the controller and switches it to the automatic mode.

$$K_P = 0,45 \cdot K_U \quad T_I = \frac{T_U}{1,2} \tag{3}$$

The second procedure was based on step method. This method is used among others in SIPART DR 22 controller [9], [8]. The scheme of the procedure is:

- Operator declares the amplitude of excitation of control signal Δu.
- Program switches controller to the manual mode.
- Program changes the control signal u by the Δu value. Controlled object reacts on the excitation and trends to new steady state.
- Program analyses the changes of y signal and calculates lag time τ as the time counted from the start of experiment to the moment when the value of y will change by 5%.
- Program still analyses the changes of y signal and calculates time constant T as the time counted from the determination of lag time τ to the moment when plant output y will reach new steady state. New steady state is reached when the changes of the value of the next three samples of y will be smaller than 10%.

- Program calculates proportional gain k as the ratio of changes of plant output signal Δy to the changes of control signal Δu.
- Controller parameters are calculated according to the following tuning rules: *Chien 0% overshoot* (4), *Chien 20% overshoot* (5), *Cohen-Coon* (6), *Hey* (7), *Oppelt* (8), *Ziegler-Nichols* (9). We chose them from O'Dwyer's book [10].
- Program asks the operator to choose one of the six controller parameters sets. Program sends parameters K_P and T_I to the controller and switches it to the automatic mode.

$$K_P = \frac{0,6 \cdot T}{k \cdot \tau} \quad T_I = 4 \cdot \tau \tag{4}$$

$$K_P = \frac{0,7 \cdot T}{k \cdot \tau} \quad T_I = 2,33 \cdot \tau \tag{5}$$

$$K_P = \frac{1}{k}\left(0,9\frac{T}{\tau} + 0,083\right) \quad T_I = T\left[\frac{3,33\left(\frac{\tau}{T}\right) + 0,33\left(\frac{\tau}{T}\right)^2}{1 + 2,22\left(\frac{\tau}{T}\right)}\right] \tag{6}$$

$$K_P = \frac{x_1}{k} \quad T_I = x_2 \cdot \tau \tag{7}$$

where: x_1, x_2 – is the additional coefficients according to table 1.

Table 1. Determining of x_1 and x_2 coefficients values depending on the ratio of lag time and time constant

Ratio $\frac{\tau}{T}$	**Coefficient** x_1	**Coefficient** x_2
0,1	9,5	3,2
0,125	7	3,2
0,165	5	3
0,25	3,4	2,8
0,5	1,8	2

$$K_P = \frac{1}{k}\left(0,77\frac{T}{\tau} - 1\right) \quad T_I = 3,32 \cdot \tau \tag{8}$$

$$K_P = \frac{0,9 \cdot T}{k \cdot \tau} \quad T_I = 3,33 \cdot \tau \tag{9}$$

Both of two autotuning procedures were tested to evaluate control performance. Two control performance indexes: IAE (10) and ISE (11) were used. Smaller value of each index means better control performance.

$$\mathrm{IAE} = \int_0^t \left| y_0(t) - y(t) \right| dt \tag{10}$$

$$\mathrm{ISE} = \int_0^t \left[y_0(t) - y(t) \right]^2 dt \tag{11}$$

3. Results

In the relay procedure, dynamic and static behaviour of plant were described by two parameters: ultimate gain K_U, and ultimate period T_U. We received the following values: K_U =4,01 and T_U=7,54s.

In the step procedure, program determined three parameters that describe plant dynamics: plant proportional gain k, time constant T, and lag time τ. The following values: k=0,75, T=7,25s, and τ=4s were received. By comparing these estimated values with true values (i.e. k=0,8, T=9s, and τ=3s) error of estimation can be calculated: 6, 19, and 33% for plant proportional gain, time constant and lag time respectively.

On the basis of estimated plant parameters, program determined several sets of PI controller parameters, i.e. proportional gain K_P and reset time T_I according to (3-9). Values of controller parameters, as well as values of performance indexes for each tuning rule are presented in table 2.

Table 2. Results of autotuning procedures: values of controller parameters and values of performance indexes

Name of the tuning rule A: relay method B: step method	**Controller parameters**		**Performance indexes**	
	K_P	T_I	ISE	IAE
A: *Aström-Hägglund*	1,8	6,29	295,99	109,08
B: *Chien 0% overshoot*	1,46	16	319,03	161,05
B: *Chien 20% overshoot*	1,7	9,32	296,91	124,43
B: *Cohen-Coon*	2,3	6,29	279,08	101,96
B: *Hay*	1,34	8	323,38	127,18
B: *Oppelt*	0,53	13,28	587,88	267,17
B: *Ziegler-Nichols*	2,19	13,32	299,86	141,25

Relay procedure and step procedure were tested with the same experiment scheme. The setpoint value y_0 changed every 40s in the range of 32–78%. Fig. 2. shows the signals of setpoint y_0, plant output y, and control u measured during tests.

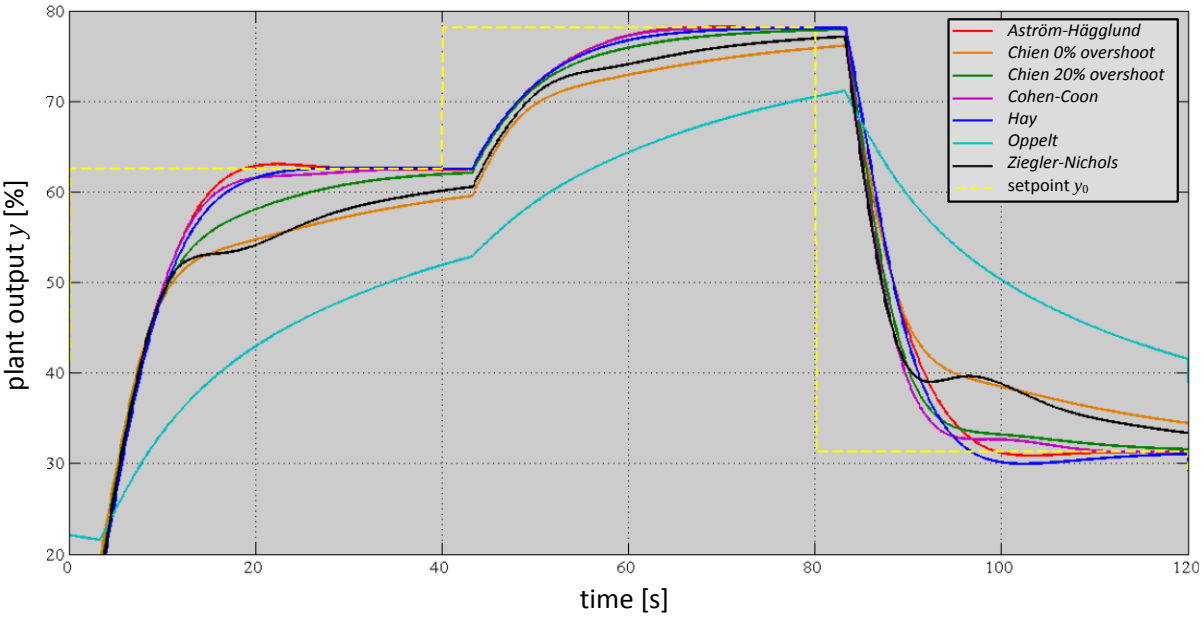

Fig. 2. Output of the plant for different controller parameters

Cohen-Coon was the best tuning rule among all six rules tested in step method (it has the smallest values of performance indexes). Tuning rules *Chien 20% overshoot*and *Hay* had IAE values 22 and 25% higher than the best rule. The greatest value of IAE index for the *Oppelt's* rule was observed and the value was 162% higher than the *Cohen-Coon's* rule. The rule of *Cohen-Coon* had also the lowest ISE index. Rules *Chien 20% overshoot*and*Ziegler-Nichols* were worse (i.e. they had higher value of ISE) by

approximately 7%. For *Chien 0% overshoot*and*Hay* rules performance index ISE was higher by approximately 15%. The worst control performance was related to the *Oppelt's* rule and the value of ISE was 110% higher than for the rule of *Cohen-Coon*. The rule of *Aström-Hägglund*, used in the relay method provided comparable control performance with the best step method. In this method control performance indexes were higher by 6% (IAE) and 7% (ISE).

The analysis of transient responses (fig. 2.) shows that the controller parameters obtained by rules of *Oppelt* and *Chien 0% overshoot* do not guarantee the achievement of setpoint within the assumed time of 40s. Therefore, they should be applied to processes where cannot be any overshoots and slow reaching of setpoint is permissible. This is caused by too slow effect of integral action of the controller. The reset time T_I was equal to 13,3 and 16s for rules of *Oppelt* and *Chien 0% overshoot*. The controller parameters obtained by rule of *Ziegler-Nichols*also had reset time equal to 13,3s, but higher value of proportional gain K_P made transient response faster.

4. Conclusion

In the study implementation of two PID autotuning procedures in HMI device was presented. HMI panel can play a role of universal tuning PID controllers tools, especially for those without built-in autotuning function. Six tuning rules for step method and one for relay method were used. The experiment exposed differences in the values of control performance indexes. Therefore, the effectiveness of the autotuning procedure depends not only on the accuracy of controlled object identification, but also on the tuning rule. The step procedure requires the development of more accurate way to estimate lag time and time constant. The problem may also be a long time of experiment in the case of identification of a plant with a long time constant. The relay method seems to be easier to implement, but offers only one set of control parameters. For simulated plant, with characteristic similar to thermal system, the control performance of control system tuned by relay method was slightly worse than the control performance of system tuned by step method. The difference between control performance indexes was less than 7%. For this reason, we suggest control engineers to try the relay method at the first place.

5. References

[1] Y. Li, K. H. Ang, and G. C. Y. Chong, "Patents, software and hardware for PID control: an overview and analysis of the current art," *IEEE Control Syst. Mag.*, vol. 26, no. 1, pp. 42–54, 2006.

[2] V. V. Denisenko, "Modifications of PID Regulators," *Autom. Remote Control*, vol. 71, no. 7, pp. 1465–1475, 2010.

[3] R. R. Rinehart, F. G. Shinskey, and H. L. Wade, "Control modes - PID variations," in *Process control and optimization*, B. G. Liptak, Ed. Boca Raton: Taylor and Francis, 2006.

[4] R. Salat and M. Awtoniuk, "Black box modeling of PIDs implemented in PLCs without structural information: a support vector regression approach," *Neural Comput. Appl.*, vol. 26, no. 3, pp. 723–734, 2015.

[5] J. F. Smuts, *Process Control for Practitioners*. League City: OptiControls Inc, 2011.

[6] M. Awtoniuk, "Dostrajanieregulatora z pominięciem eksperymentu identyfikacji obiektu przy użyciu Imperialist Competitive Algorithm," *Elektron. Konstr. Technol. Zastos.*, vol. 53, no. 12, pp. 46–49, 2012 [in Polish].

[7] J. Kwaśniewsk, *Programowalny sterownik SIMATIC S7-300 w praktyce inżynierskiej*. Legionowo: BTC, 2009 [in Polish].

[8] H. Kunert, "Metoda identyfikacjii doboru nastaw dla regulatorów z samonastrajaniem" *PomiaryAutom. Kontrola*, vol. 4, no. 6, pp. 14–18, 2000.

[9] J. Müller, *Controlling with SIMATIC: practice book for SIMATIC S7 and SIMATIC PCS7 control systems*. Berlin: Publicis Corporate Publishing, 2005.

[10] A. O'Dwyer, *Handbook of PI and PID controller tuning rules*. London: Imperial College Press, 2009.