ITM Web of Conferences **15**, 05009 (2017) DOI: 10.1051/itmconf/20171505009
CMES'17

Implementation of PID autotuning procedure in PLC controller

Marcin Daniun[1], *Michał* Awtoniuk[1,*], and *Robert* Sałat[1]

[1]Warsaw University of Life Sciences, Faculty of Production Engineering, Nowoursynowska 164, 02-787 Warsaw, Poland

Abstract. In this paper, we present the automatic PID tuning procedure based on the Method of Moments and AMIGO tuning rules. The advantage of the Method of Moments is that the time constant and transport delay are estimated at the areas rather than on the individual points. This results in high resistance to the measurement noises. The sensitivity to measurement noises is a serious problem in other autotuning methods. The second advantage of this method is that it approximates plant during identification process to first order model with time delay. We combined the Method of Moments with the AMIGO tuning rules and implemented this combination as a stand-alone autotuning procedure in Siemens S7-1200 PLC controller. Next, we compared this method with two built-in PID autotuning procedures which were available in Siemens S7-1200 PLC controller. The procedure was tested for three types of plant models: with lag-dominated, balanced, and delay-dominated dynamics. We simulated the plants on a PC in Matlab R2013a. The connection between the PC and PLC was maintained through a National Instruments data acquisition board, NI PCI-6229. We conducted tests for step change in the set point, trajectory tracking, and load disturbances. To assess control quality, we used IAE index. We limited our research to PI algorithm. The results prove that proposed method was better than two built-in tuning methods provided by Siemens, oscillating between a few and even a dozen percent in most cases. The proposed method is universal and can be implemented in any PLC controller.

1 Introduction

A proportional-integral-derivative (PID) controller is still the most widely used algorithm in industrial applications [1]. Individual terms can be activated or deactivated to create P, PI, PD, and PID controller type but the most common amongst them is PI. The biggest problem with commissioning PID algorithm is a controller tuning process. Manual tuning can be time consuming. It also requires specialised knowledge and experience in the field of control engineering. The research conducted in the 1990's found that 80% of PID applications were badly tuned [2]. For this reason, more and more PID controllers are equipped with autotuning function. Many controller manufacturers as well as scientists are trying to develop new, more accurate and efficient autotuning procedures.

In this paper, we present the procedure for automatic tuning of the PID controller based on the Method of Moments, as a method of system identification, and the AMIGO tuning rules. The advantage of the Method of Moments is that the parameters of model are estimated at the areas rather than on the individual points. This results in high resistance to the measurement noises. We implemented Method of Moments + AMIGO as a stand-alone autotuning procedure in Siemens S7-1200 PLC controller. Next, we compared the method with two built-in PID autotuning procedures which were available in Siemens S7-1200 PLC controller.

2 PID autotuning

The tuning process of the PID controller is based on determining the value of the parameters of the controller. Selection of controller parameters is a difficult task; therefore, any manufacturers decide to implement special procedures to make tuning process automatically. This is called autotuning.

There are many different autotuning procedures but generally autotuning has two major phases: control object identification and controller parameters calculation. The most common approach in the first phase is to build a model of control object. Model should be as simple as it is possible, e.g. first order model with time delay (FOTD). Equations used to calculate the controller parameters, based on model parameters, are called tuning rules. The literature is abundant in different tuning rules, e.g. O'Dwyer in [3] shows 649 controller tuning rules dedicated for FOTD model only.

Autotuning methods can be divided into two groups. The first group uses step response, and the second is based on relay method. In the PLCs both methods are available. Siemens S7-1200 PLC controller can be an example with two autotuning procedures: Pretuning and Fine Tuning implemented in it.

The Pretuning procedure is based on the step response. Firstly, the controller is turned into manual mode and controller output u is forced to a value set by

* Corresponding author: michal_awtoniuk@sggw.pl

DOI: 10.1051/itmconf/20171505009

the user. Secondly, value of plant output y is observed until it reaches the point of inflection. Finally, the Chien-Hrones-Reswick tuning rules are used to calculate the parameters of the controller [4], [5].

The Fine Tuning procedure uses the relay method developed by Åström and Hägglund, which is based on the Ziegler-Nichols Ultimate Cycling method [6], [7]. The method involves a series of oscillations of the plant output y. Oscillations are forced by switching the controller output u between two values. The controller parameters are calculated with Ziegler-Nichols tuning rules. The relay method is still being developed. Currently, researchers are focused on asymmetric relay autotuning and adaptation the method to plants with more input and output signals such as TITO [8].

In this paper, we compare two build-in autotuning procedures with combination of Method of Moments and AMIGO tuning rules. The Method of Moments is a system identification technic based on the step response characteristic. In this method, the plant is modelled by FOTD model (1).

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

where: K -- static gain, T -- time constant, L -- time delay.

FOTD model has tree parameters: static gain K, time constant T, and time delay L. In the Method of Moments parameters of model are estimated at the areas rather than on the individual points. In the fig. 1 graphical interpretation of the method are presented.

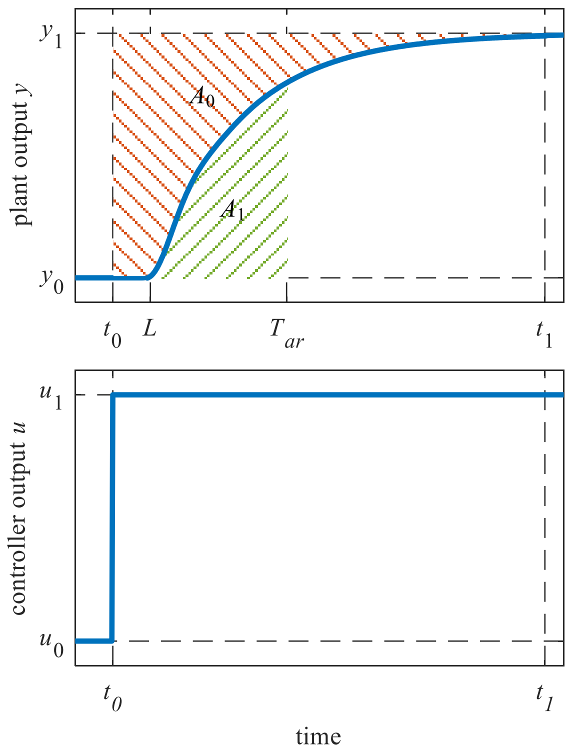

Fig. 1. Graphical interpretation of parameters calculated in the Method of Moments.

The static gain is calculated by equation

$$K = \frac{\Delta y}{\Delta u} = \frac{y_1 - y_0}{u_1 - u_0} \tag{2}$$

where: Δy -- increase of plant output, Δu -- increase of controller output.

Calculation of time constant T and time delay L is not as simple as static gain K and requires estimation of additional variables: area A_0 (integrated distance between steady-state of step response y_1 and plant output y), time T_{ar} (average residence time), and area A_1 (area under the step response up to time T_{ar}) according to equations (3-7)

$$A_0 = \int_{t_0}^{t_1} [y_1(t) - y(t)]dt \tag{3}$$

where: y_1 -- steady-state of plant step response, y -- plant output.

$$T_{ar} = \frac{A_0}{K} \tag{4}$$

$$A_1 = \int_{t_0}^{T_{ar}} y(t)dt \tag{5}$$

$$T = \frac{eA_1}{K} \tag{6}$$

where: e – Euler's number.

$$L = T_{ar} - T \tag{7}$$

After FODT model parameters estimation, calculation of controller parameters is performed with AMIGO tuning rules. These rules are well described in [7]. We used PI algorithm thus our controller had two parameters: proportional gain K_p and reset time T_i.

$$K_p = \frac{0.15}{K} + \left[0.35 - \frac{LT}{(L+T)^2}\right]\frac{T}{KL} \tag{8}$$

where: K_p – proportional gain.

$$T_i = 0.35L + \frac{13LT^2}{T^2 + 12LT + 7L^2} \tag{9}$$

where: T_i – reset time.

3 Materials and methods

Fig. 2 presents scheme of data collection. Key elements used in our research were:
– PLC controller Siemens SIMATIC S7-1200,
– Data acquisition board National Instruments NI-6226,
– Matlab&Simulink software.

All autotuning procedures, i.e. Pretuning, Fine Tuning, and Method of Moments + AMIGO, were implemented in PLC controller. The first and the second procedure are available in S7-1200 PLC controller as a build-in functions. We implemented the third

 DOI: 10.1051/itmconf/20171505009

procedure by ourselves in the form of function block. We have programmed the PLC controller in ladder diagram language in TIA Portal V13 software. To calculate the integrals from equations (3) and (5) we used rectangle rule. The sampling period was 20 ms.

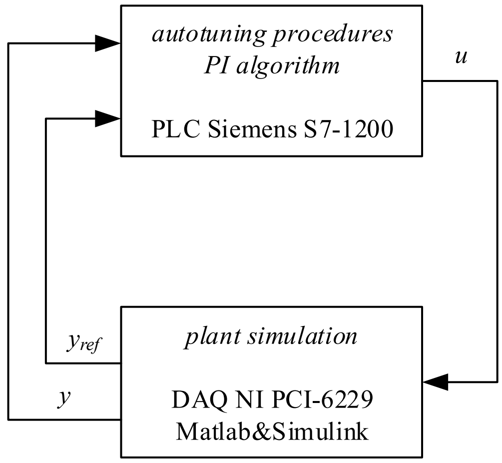

Fig. 2. Block diagram of data collection.

PID algorithm, a build-in function, is implemented in S7-1200 PLC controller as a technological function available in a standard library. The function block name is PID_Compact. According to manual PID algorithm is described as [4]:

$$u(s) = K_p\left[\left(by_0 - y\right) + \frac{1}{T_i s}\left(y_0 - y\right) + \frac{T_d s}{aT_d s + 1}\left(cy_0 - y\right)\right] \quad (10)$$

where: b -- proportional term weighting, y_{ref} -- set point, a -- derivative delay coefficient, T_d -- derivative time, c -- derivative term weighting.

The values of coefficients a, b, and c were estimated automatically during Pretuning and Fine Tuning process. In Method of Moments + AMIGO autotuning procedure we set them to a constant value equal 1. We limited our research to PI algorithm so T_d=0.

We simulated the plants in Matlab&Simulink. Communication between PLC and Matlab&Simulink was maintained throughout data acquisition board in the way depicted in the fig. 2. All signals, i.e. set point y_{ref}, plant output y, and controller output u, were voltage signals from the range 0-10V.

Autotuning procedures were tested with tree plants described by transfer functions (11-13). P_1 is a lag-dominated plant, P_2 is balanced, while P_3 is delay-dominated plant [9].

$$P_1(s) = \frac{1}{(s+1)(0.1s+1)(0.01s+1)(0.001s+1)} \quad (11)$$

$$P_2(s) = \frac{1}{(s+1)^4} \quad (12)$$

$$P_3(s) = \frac{e^{-s}}{(0.05s+1)^2} \quad (13)$$

To compare results of autotuning procedures three scenarios were prepared: step change in the set point, trajectory tracking, and load disturbances. Fig. 3 shows changes of set point signal in all scenarios. In the first case, step change of set point was applied in the 10th second to a value of 7V and the next step was applied in the 110th second to a value of 3V. Test lasted 210 s. In the second scenario, the set point was changed according to trajectory of response of the first order system $(5s+1)^{-1}$ to step change to value of 7V. In the 100th second another trajectory was applied. It was response of the first order system $(25s+1)^{-1}$ to step change to value of 3V. Test lasted 250 s. In the last scenario step change of set point was applied in the 5th second to a value of 5V, remaining constant to the end of experiment. Next, in the 155th second the load disturbance of value of 1V was applied to a plant input, i.e. controller output signal u. Test lasted 305 s.

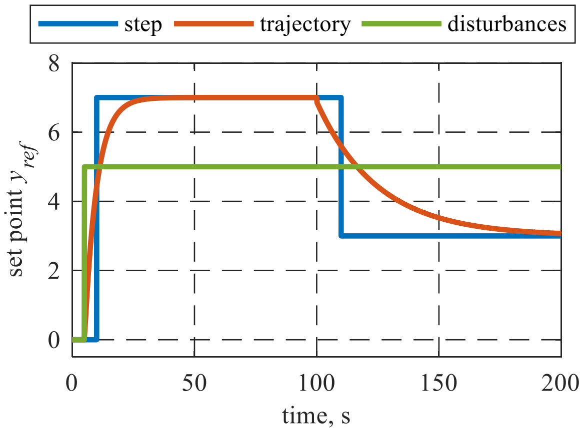

Fig. 3. Changes of set point signal in tree scenarios.

We compared all autotuning procedures from control performance point of view. To assess control quality, we used Integral Absolute Error (IAE) index:

$$IAE = \int \left| y_{ref}(t) - y(t) \right| dt \quad (14)$$

4 Results

Tables 1, 2, and 3 show results for lag-dominated plant P_1, balanced plant P_2, and delay-dominated plant P_3, respectively. The tables include controller parameters calculated by each autotuning procedures and values of control quality index IAE for step change, trajectory tracking, and load disturbances scenario.

After system identification process, by means of Method of Moments, plants P_1, P_2, and P_3 were modelled by M_1, M_2, and M_3 models described by following transfer functions

$$M_1(s) = \frac{e^{-0.12s}}{1.119s + 1} \quad (15)$$

$$M_2(s) = \frac{e^{-1.962s}}{2.168s + 1} \quad (16)$$

DOI: 10.1051/itmconf/20171505009

$$M_3(s) = \frac{e^{-1.052s}}{0.117s+1} \quad (17)$$

Fig. 4-6 show plant output and controller output signals during experiments in all three scenarios.

Table 1. Controller parameters and control performance assessed for plant P_1.

	Pretuning	**Fine Tuning**	**Method of Moments + AMIGO**
controller parameters			
K_p T_i [s]	6.31 0.49	11.1 0.37	4.4 0.49
control performance: IAE index			
step trajectory disturbances	7.33 2.22 3.38	7.44 1.84 3.82	7.09 1.54 3.14

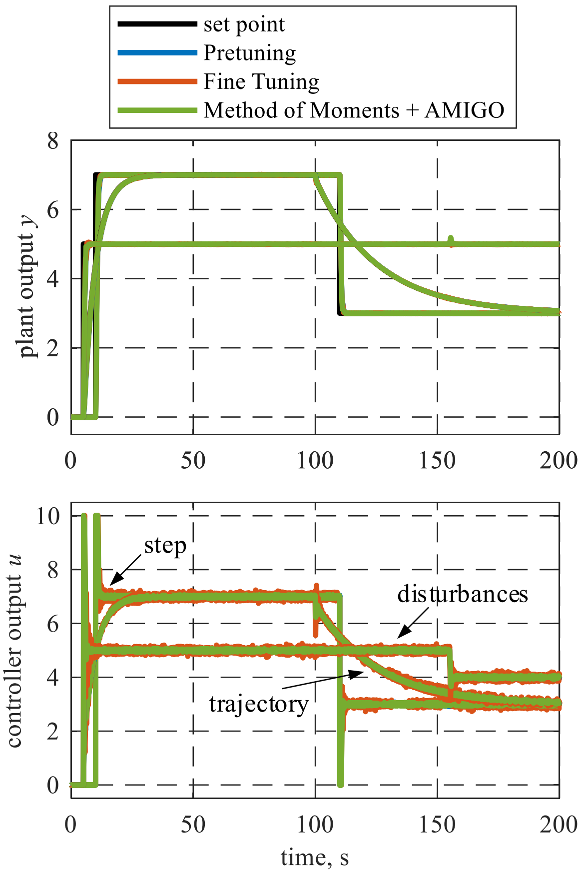

Fig. 4. Plant output and controller output for plant P_1 (curves are very close to each other).

Table 2. Controller parameters and control performance assessed for plant P_2.

	Pretuning	**Fine Tuning**	**Method of Moments + AMIGO**
controller parameters			
K_p T_i [s]	1.92 5.72	1.6 5.48	0.7 2.27
control performance: IAE index			
step trajectory disturbances	60.3 44.8 30.2	54.5 49.4 27.4	49.5 35.5 26.5

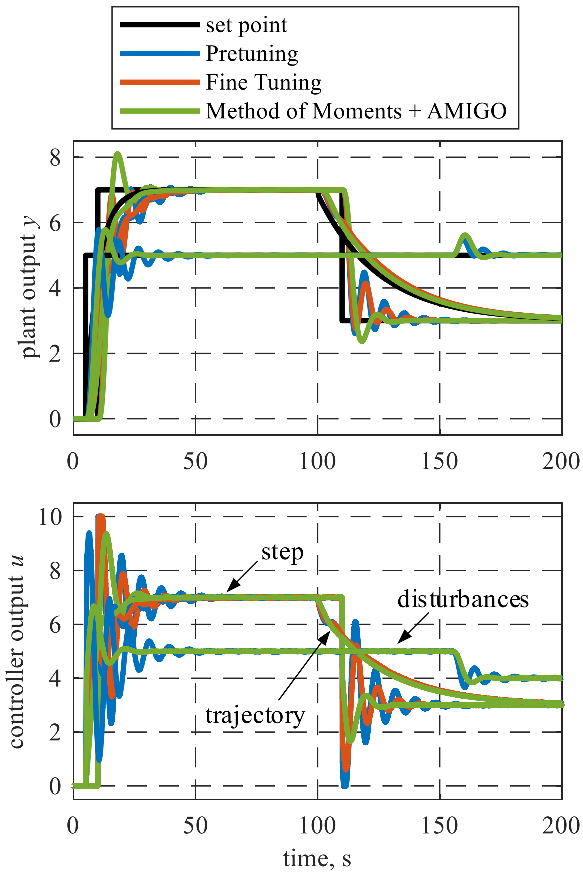

Fig. 5. Plant output and controller output for plant P_2.

DOI: 10.1051/itmconf/20171505009

Table 3. Controller parameters and control performance assessed for plant P_3.

	Pretuning	**Fine Tuning**	**Method of Moments + AMIGO**
controller parameters			
K_p T_i [s]	0.26 4.09	0.57 1.89	0.25 0.51
control performance: IAE index			
step trajectory disturbances	181 179 98.7	41.2 40.8 22.6	22.9 22.5 13

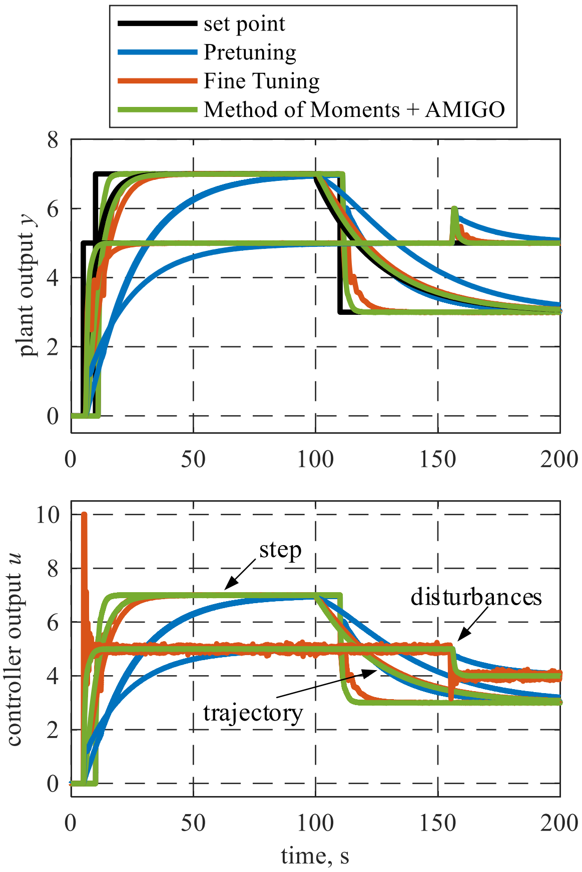

Fig. 6. Plant output and controller output for plant P_3.

All tested methods allow to tune the PID controller so that in each testing scenario tuned systems reached the steady state after set point change or load disturbance. There was no case of unstable system and the overshoots were small.

Results for lag-dominated plant (P_1) show the value of IAE index for Method of Moments + AMIGO was smaller by 3.3%, 30.6%, and 7.1% for step change in the set point, trajectory tracking, and load disturbances, respectively compare to Pretuning procedure. In the case of balanced plant (P_2) proposed procedure allows to achieve control performance index smaller by 9.2%, 28.2%, and 3.25% for step change in the set point, trajectory tracking, and load disturbances respectively compare to Fine Tuning procedure. However, the greatest advantage of described procedure can be seen for delay-dominated plant (P_3). The value of IAE index was smaller by 44.4% (step change in the set point), 44.7% (trajectory tracking), and 42.7% (load disturbances) in comparison to Fine Tuning procedure.

5 Conclusions

Our goal was to implement combination of Method of Moments and AMIGO as a new autotuning procedure to PLC controller. We also compared this method with two built-in PID autotuning procedures which were available in Siemens S7-1200 PLC controller. The results show that control loops tuned by proposed method has lower values of control quality index.

In future works we will focus on minimising the errors of integrals calculation caused by the rectangle rule. The trapezoidal rule should give better approximation. The source code in ladder diagram language should be optimised to reduce the controller memory occupation.

References

1. L. Desborough, R. Miller, AIChE Symp. Ser., **98**, 326, 169–189 (2002)
2. P. Van Overschee, B. De Moor, IFAC Proc. Vol., **33**, 4, 595–600 (2000)
3. A. O'Dwyer, *Handbook of PI and PID controller tuning rules*, London: Imperial College Press (2009)
4. Siemens, *SIMATIC S7-1200, S7-1500 PID control function manual* (2017)
5. J. Müller, *Controlling with SIMATIC: practice book for SIMATIC S7 and SIMATIC PCS7 control systems*, Berlin: Publicis Corporate Publishing (2005)
6. K. J. Åström, T. Hägglund, *PID Controllers: Theory, Design, and Tuning*, ISA (1995)
7. K. J. Åström, T. Hägglund, J. Process Control, **14**, 6, 635–650 (2004)
8. J. Berner, K. Soltesz, T. Hägglund, K. J. Åström, IFAC-PapersOnLine, **50**, 1, 5332–5337 (2017)
9. J. Berner, K. J. Åström, T. Hägglund, IFAC Proc. Vol., **47**, 3, 11288–11293 (2014)