

A Novel Self-tuning Zone PID Controller for Temperature Control via a PLC code

Abbas Rezaei Ashtiani¹, Majid Ghandchi²

1. Department of Mechatronics Engineering, Ahar Branch, Islamic Azad University, Ahar, Iran

Email: rezaei.a55@gmail.com (Corresponding author)

2. Department of Electrical & Electronics Engineering, Ahar Branch, Islamic Azad University, Ahar, Iran

Email: m-ghandchi@iauh.ac.ir

Abstract

S7-1200 with Tia Portal technology has become a Standard function of distributed control systems. Self-Tuning methods belong to Programmable Controllers (PLC) techniques. PLC techniques contain software packages for advanced control based on mathematical methods. S7-1200 tools are designed to increase the Process Capacity, yield, and quality of products. Most of the present time digital industry regulators and PLCs are provided with some kinds of the Self Tuning constant algorithms. Practical part of the present paper deals with the design of the control systems, which contain Self Tuning regulators and Zone PID features. A Control System with PID Self Tuner is designed by Siemens along with visualization in Tia Portal to do so. There is a description of an implementation of the PID regulator as a function block, which can also be used for extension Zone PID control functions. Control Systems for relay and moment Self Tuner with visualizations in Tia Portal are also designed and tested in ZAFAR Steel Plant, BONAB, IRAN.

Keywords: PLC S7-1200, Self Tuning PID, Zoning PID, Tia Portal, Simulation Furnace

1- Introduction

It is not possible to increase the demand for the performance, quality, and efficiency of industrial processes without subvention of the control systems. Control systems' expansion is influenced by many improvements in the area of the microelectronics and application of the mathematical methods developed in control theory. APC (Advanced Process Control) methods are modern control methods that were used in control systems. All the PLC producers offer APC functions in their portfolio. APC contains software

packages for advanced control based on mathematical methods. There are not only tools for self tuning, but also tools for manufacturing costs optimization and energy savings. APC tools are designed to increase the process capacity, yield, and quality of products and for costs' savings. These reasons have become very common nowadays, because of the worldwide economic depression. APC tools are implemented to DCS/PCS (Process Control Systems/Distributed Control Systems) based on the PLC (Programmable Logic Controller). Some of the methods are

implemented directly to the PLC but most of them are implemented as supervisory functions on higher levels of the control system. These functions offer diagnostic, monitoring and optimization functions for control implemented directly in the PLC.

2-SELF-TUNING METHODS WITH ZONE PID

The main part of the control systems are still the PID regulators. Control theory deals with definition of their parameters. The starting point of the PID regulator constants calculation is a simple experiment apart from the algebraic approach. A simplified system model is created on the basis of the experiment. The PID regulator constants are calculated regarding the parameters of the identified system model. The idea of the automation of this process is evident and most of the PID controllers are equipped by some self-tuning algorithms (self-tuner). There are several methods for tuning PID controllers. Generally it can be distinguished between model based self-tuner and non-model based self-tuner. The controlled system is generally identified as three or four parameters' model (a first-order system with time delay or a second order system with time delay) and the constants of the controller are calculated according to the identified model in the case of the model based self tuner. The constants of controllers are directly edited according to the parameters of the measured responses in the case of non-model based self-tuner. [1], [2]

The following text in this chapter describes the most common self tuning methods and their implementation in the commercial PLCs. Methods based on step response are

primarily used for the systems with slow response (e.g. temperature systems) and are based on measurement and evaluation of the step response of the controlled system as the reaction on the set point change. The controlled system is identified as a three or four parameters' model. [3]

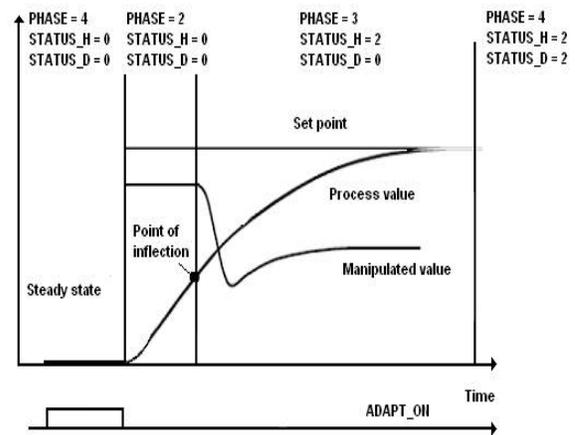


Fig.1. The step response method by PID Self Tuner [3]

Implementation of this method is very common. PID Self Tuner by Siemens (fig. 1), PIDE by Rockwell Automation, TUNE by ABB and Profit PID by Honeywell are based on a step response tuning method. Other PLC producers have also their own step response tuning methods (B&R, Omron, GE Fanuc, Mitsubishi etc.). The relay method comes out from the critical gain and the critical period of the controlled system. There are two modes: a setting mode and a control mode. The relay regulator is connected in the closed loop in setting phase.

The process value oscillates along set point in closed loop with relay. Controlled systems' behavior from the frequency point of view is generally a low pass filter. This fact is used

for approximate analysis of the controlled system. The PID regulator constants are calculated from identified critical gain and critical period of the system by some control laws, e.g. Ziegler-Nichols method.

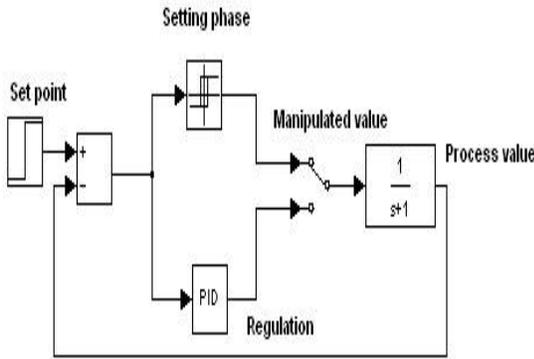


Fig.2.The principle of the relay method

Implementation of the relay method: Siemens S7 200, Bernecker & Rainer, Tecomat. Another self-tuning method is so called system moments method. This method was developed by Prof. Schlegel in ZCU Pilsen, Czech Republic. This method extends a control function of the standard PID regulator with a self-tuning method based on three characteristic process values determined by an identification procedure based on suitable rectangle pulse. [4]

An operator has to reach steady state of the controlled system, choose the type of the regulator and start the identification procedure. Rectangle manipulated signal is then applied to the controlled system. Three evaluated values have a clear physical meaning: static gain, delay and reaction time. They are obtained from the first three moments of the system response. The parameters of the PID regulator are then

calculated according to these evaluated values. [5], [6] There are other several self tuning methods implemented in PLCs.

This more recent addition to SELF TUNING METHODS of metrology furnaces offers an alternative for those who prefer 3-Zone furnaces to heat pipe technology that need high temperature operation.

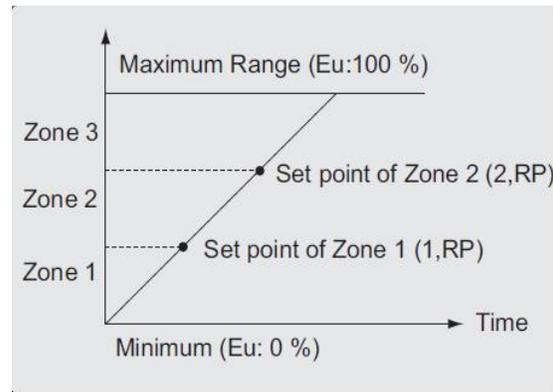


Fig.3.Zone PID Method for the varying of system function $H(s)$

In the innovated method, we divide the operating temperature range to three Zones: Zone 1, Zone 2, and Zone 3 as displayed in Figure 3. For each Zone the Self tuning method is used for finding P, I, D Parameters separately. Because of the varying system function, $H(s)$ by temperature zones of furnaces the PID parameter is different for these three zones. By PLC program and comparison instruction, PID parameter changes automatically as temperature rise of furnace. [7], [8]

3- Auto tuning PID via s7-1200 in Tia portal

A few weeks ago, I had the chance to experiment with serial communications using a Siemens S7-1200 PLC outfitted with a

CM1241 (RS-485 module). Due to the limited number of projects we have had, using the 1200 line of PLCs, this was an office first. And, while it was very straightforward, it was still a learning experience. [9]

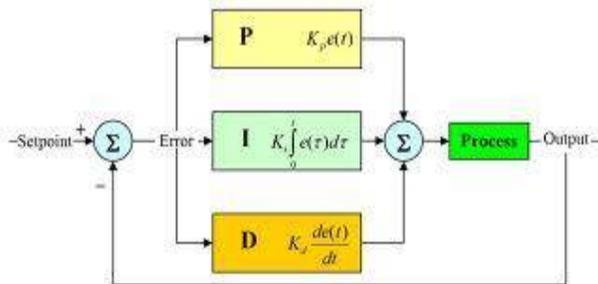


Fig.4. Block diagram of the PID controller

Recently I had another chance to be DMC's S7-1200 pioneer - this time using the S7-1200's PID blocks and luckily, not involving death by dysentery, cholera, or snakebite. Like serial communications, PID was also fairly straight forward. But, there were a few things I picked up along the way that may be helpful for the next person looking to close the loop. If you've thought about PID on an S7-1200, you probably know that Siemens is kind enough to provide two PID function blocks in STEP 7: PID_Compact and PID_3Step. For most purposes, I would think PID_Compact (for use with continuous input/output variables) would be sufficient, but if you're trying to control motor-actuated devices, such as valves that require digital open/close signals, the PID_3Step block could be very useful. In my case, I was controlling the temperature in a furnace by actuating a proportional valve - so PID_Compact fit the bill. So, if you're looking into using PID on your S7-1200, I hope the following tips may be useful in getting you up to speed. [10]

3.1 Where are the PID blocks?

One of the first questions I had after discovering that there were native PID blocks provided in STEP 7 was "Where are they?". If you are looking, expand the instructions tab on the right side of the development screen in STEP 7. This is the tab that contains all of your tools (timers, counters, math, etc). Under the basic instructions, there's another group called "Technology." Expand this and you should see a group of instructions called **PID Control**. You can click and drag a block from here directly to your code to add it to your project. But, before you add it; be careful where you're going to drop it. You don't want the PID block to execute every scan (it needs time to respond to changes in the control value), so add it to a cyclic OB. Choose how often your cyclic OB executes based on how fast your process is.

3.2 . I threw my PID block in, now what?

After you've added your block, it's up to you to configure it. Luckily, there's a very handy configuration tool provided. Just click on the 'toolbox' icon in the upper right hand corner of the block to open up the configuration window. This will guide you through setup. The first set of choices you have is "**Basic Settings**." This is where you define what kind of inputs/outputs you're using. In my case, I wanted to control temperature using a set point and input in degrees F, and an output in % that I could feed directly to an analog output to my valve. [11]

$$G_R(s) = K_P + \frac{1}{T_I \cdot s} + \frac{T_D \cdot s}{\alpha \cdot T_D \cdot s + 1} \quad (1)$$

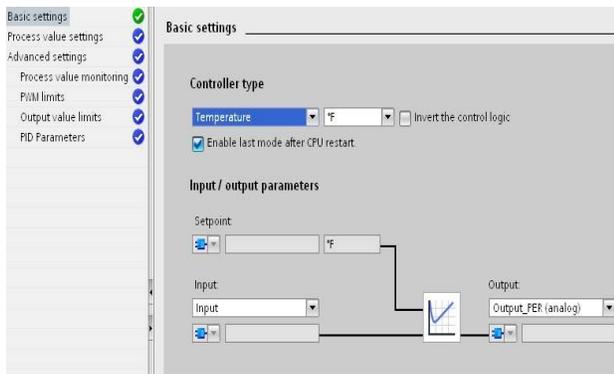


Fig.5.PID controller software settings

Next, if using Inper Per, you will need to define your "**Process Value Settings**." These options will define the scaling used to convert your percentage input to engineering units (for comparison to your set point). As I chose to use the input option (not Input_PER), I had nothing to do here. Finally, there are "**Advanced Settings**." This is where you can take advantage of the built-in warnings for process value monitoring, set minimum on/off times for a PWM output, output value limit values, and manually enter PID gains. It's worth noting here that the output options for the block are plentiful - engineering value, Output_PER, and PWM. And, you can use any or all of them simultaneously. So, basically this block is a one stop shop for any kind of PID control. [12]

3.3 .Is there an auto-tune function?

Now you've set up your block and have provided inputs/outputs, how do you get started? If you download your program to your PLC you will quickly see that the PID loop is not currently active. You will need to either manually start it, or walk through the tuning procedure. We'll get to manual tuning in a moment, so let's look at the auto-tune process first. Earlier, we used the "**Toolbox**"

icon on the block to configure the block. Right next to that icon is an icon with what looks like a pair of pliers and a yellow warning sign. This is how we enter the auto-tuning process. While the program is running on the PLC and you are online, you can use this icon to launch the tuning window. From here, you can start the auto-tuning process. I think this is fairly straightforward and doesn't need much comment, but do remember to be careful before starting auto-tune as the controller will take over and measure the response to different outputs. Once pre-tuning is complete, you can do further fine-tuning. Once the tuning process is complete, the PID loop will be activated and you're ready to go. That's it. [13], [14]

3.4 .Can I manually tune my PID loop?

Unfortunately, the process I was trying to control did not lend itself well to auto-tuning. In my case, the temperature response of the furnace was incredibly slow and the lag between output and response was great enough that auto-tuning wasn't a viable option. Thankfully, there are manual tuning options as well. I was lucky that I could schedule some time for offline tuning. I was able to heat the furnace up to steady-state, give a step-change in my output, and record the response. The tuning window is still helpful for this as it will record the process variable for you. Once I had my response, I was able to dig out some old college textbooks and calculate the gains for myself. While you can set gains through the configuration window, I preferred accessing the gains directly through the PID_Compact block's data block. Most of the important values you want are located in the data block

structure "sRet." There are pretty decent explanations (including a formula for how the output is determined) in the S7-1200 Easy Book for each parameter. You can also hover over each of the properties on the Advanced Settings page as well to get a short description and a link to the help file. One of the things that confused me initially was the Ctrl_A, _B, and _C, parameters, all three of which are 0 by default. You will most likely want to set B and C to 1 to enable traditional PID control. [15]

Finally, even after setting each of the gains, you will see that your PID block is still not functional. This is because you will need to manually set it into the correct operating mode. This is where sRet.i_Mode. This determines the mode into which the controller will go when it receives and enable command or comes out of manual control. You can set this to 3 (auto-mode) and then enable your controller and you're in business. While it takes a few minutes of exploring to familiarize yourself with the options, PID on an S7-1200 is pretty straightforward. [16]

4- Results in the Zafar Steel Plant, Bonab, Iran

Similar parameters of the PID regulator design are typical for the relay and the moment self tuner. The PID Self Tuner by Siemens has different proportional parts of the regulator due to the fact that the PID Self Tuner is primarily used for temperature controlled systems so that a bigger proportional part of the regulator is needed. It is possible to set up slow, normal or fast closed loop speed for the moment self tuner as a parameter of the function blocks of these

self tuners. Moment methods have small overshoots of process value in normal closed loop speed and no overshoots in slow closed loop speed. [17]

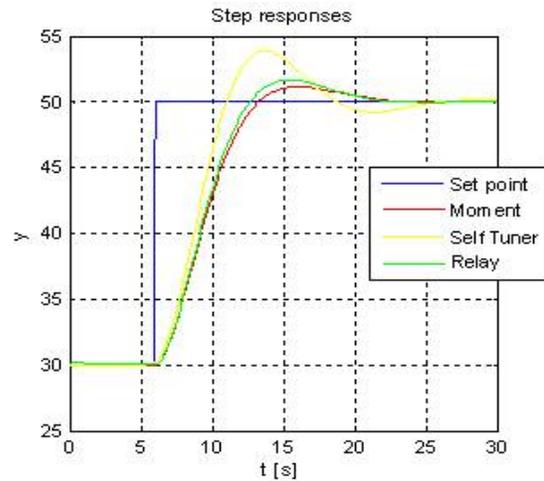


Fig.6. Control system block scheme

5. Conclusions

S7-1200 with Tia Portal technology has become Standard function of distributed control systems. Self Tuning methods belong to Programmable Controllers (PLC) techniques. PLC techniques contain software packages for advanced control based on mathematical methods. S7-1200 tools are designed to increase the Process Capacity, yield and quality of products. Because the varying of system functions, $H(s)$ by temperature zone of furnaces the PID parameter is different for temperature zones. By PLC program and comparison instruction, PID parameter changes automatically along with temperature rise of furnace. Our design of the control systems contains Self Tuning regulator and Zone PID feature. A Control System with PID Self Tuner by Siemens and with visualization in Tia Portal is programmed.

Acknowledgment

The authors would like to thank Ahar Branch , Islamic Azad University and the ZAFAR STEEL PLANT, BONAB, IRAN, for the financial support of this research, which is based on a MS thesis.

References

- [1] ASTROM, Karl, J., HAGGLUND, T., 1995. PID controllers: Theory, Design, and Tuning. Instrument Society of America. USA, 2nd Edition. 343 pages. ISBN: 1-55617-516-7.
- [2] BOBÁL, V., BOHM, J., PROKOP, R., FESSL, J., 1999. Practical Aspects of Self Tuning Regulators, Algorithms and Implementations. VUTIUM publisher. Brno, 242 pages. ISBN: 80-214-1299-2.
- [3] From Wikipedia, the free encyclopedia, https://en.wikipedia.org/wiki/Programmable_logic_controller, 2016.
- [4] KOCIAN, J., 2009. Self Tuning Controllers on PLC Background. Diploma thesis. VSB TU Ostrava 2009.
- [5] Kocian, J. and Koziorek, J. “Self-tuning techniques on PLC background and control system with self-tuning methods design”, AEEE CONTROL ENGINEERING, VOL. 8, No. 2, JUNE 2010, 40-47.
- [6] M. H. Refan, S. Bahmanpour and M. Nazari ,”Design of Turbo Compressor HMI system Based on LABVIEW” , 16th Iranian Student Conference on Electrical Engineering Islamic Azad University of Kazeroon, Kazeroon, Iran 3 – 5 September , 2013.
- [7] M. Kajan, J. Šovčík, F. Duchoň, L. Mraňko, M. Florek and P. Beňo ,”Sensoric subsystem of Automated guided vehicle,TCP communication between SIMATIC S7 PLC and Arduino”, Proceedings of the RAAD 2014,23rd International Conference on Robotics in Alpe-Adria-Danube Region ,September 3-5, Smolenice Castle, Slovakia, 2014.
- [8] O'DWYER, A., 2003. Handbook of PI and PID Controller Tuning Rules. Imperial College Press. London, 375 pages. ISBN: 1-86094-350-0.
- [9] Patel, J.,Patel, A. and Singh,R. “Development of PLC Based Process Loop Control for Bottle Washer Machine” , Procedia Technology 14, 2014, 365 – 371.
- [10] Pavković, D. , Polak, S. and Zorc, D. “PID controller auto-tuning based on process step response and damping optimum criterion”, ISA Transactions 53 ,2014,85–96.
- [11] PFEIFFER, B.-M, MOHR, D., 1998. Selbsteinstellender PIDRegler. Atp – Automatisierungstechnische Praxis. Oldenbourg Verlag. n. 11, pages 50-57.. ISSN: 3174171866.
- [12] Q. Chu, W. Li and X. Niu, “The Application of S7-200 PLC Reactive Compensation Device in Small Substation”, Fifth International Conference on Intelligent Human-Machine Systems and Cybernetics, Computer society, IEEE ,2013.
- [13] S. Tamboli, M. Rawale, R. Thoraiet and S. Agashe,” Implementation of Modbus RTU and Modbus TCP Communication using Siemens S7-1200 PLC for Batch Process”, 2015 International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM), Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Chennai, T.N., India. 6 - 8 May 2015. pp.258-263.
- [14] STEHLIK, P. , J. KOHOUTEK and V. JEBACEK . “SIMPLE MATHEMATICAL MODEL OF FURNACES AND ITS POSSIBLE APPLICATIONS” , Computers them. Engng Vol. 20. No. 11, pp. 1369-1372, 1996.
- [15] SCHLEGEL, M., BALDA, P., ŠTĚTINA, M.. Robust Autotuner: Moment Method. An article accessible(31.4.2010) on<http://www.rexcontrols.cz/downloads/clanky/PID_Autotuner-Momentova_metoda_CZ.pdf>.
- [16] T. O. Putri , Rahmadwati and B. Siswojo,”Temperature Control of Liquid Egg Pasteurization System Using PLC (Programmable Logic Controller) Siemens Simatic S7-200 and HMI (Human Machine Interface) Simatic HMI Panel”, Electrical Power,

Electronics, Communications, Controls, and Informatics Seminar (EECCIS), IEEE, 2014.

- [17] X. Zhang and D. Zhang, "Application of S7-200 PLC to the monitoring system of strip steel temperature", 978-1-4577-1600-3/12/\$26.00 ©, IEEE, 2012.



Abbas Rezaei Ashtiani received the B.S. degree in Electronics Engineering from Islamic Azad University Soufyan branch, Soufyan, Iran, in 2011, the M.S. degree in Mechatronics Engineering from Islamic Azad University Ahar branch, Ahar, Iran, in 2016. He joined the Ima Pardaz Tabriz Engineering Company in 2005, where he is currently Managing Director at Negin Tabriz Industrial Automation Company(LTD).



Majid Ghandchi received the B.S. degree in Electronics Engineering from Islamic Azad University Tabriz branch, Tabriz, Iran, in 1993, the M.S. degree in Semiconductor Devices from Islamic Azad University south Tehran branch, Tehran, Iran, in 1996. He joined the Islamic Azad University Tabriz and Ahar branch in 1997, where he is currently a lecturer with the college of Engineering.