Self-Tuning PID Controller Using Ziegler-Nichols Method for Programmable Logic Controllers

T. Yucelen [1], O. Kaymakci [2] and S. Kurtulan [3]

[1 2 3] Electrical and Electronic Faculty,
Istanbul Technical University,
Maslak, Istanbul, Turkey

e-mail: yucelent@itu.edu.tr [1], kaymakci@elk.itu.edu.tr [2], kurtulan@elk.itu.edu.tr [3]

Abstract

In this paper a modified PID controller is presented as a dynamic system controller. Here Ziegler-Nichols process reaction method is clarified to designate self-tuning, and advantages of self-tuning are also explained in detail. Moreover, simulation results of self-tuning PID controller using Ziegler-Nichols are acquired from programmable logic controller (PLC), and then are given in related topics. Finally, the complete algorithm is tested in an industrial thermal system, and results of this test are offered and explained.

Keywords: PID Control; Ziegler Nichols Process Reaction Method; Robustness, Self-Tuning Control

1 Introduction

In spite of developed modern control techniques like fuzzy logic controllers or neural networks controllers, PID controllers constitute an important part at industrial control systems so any improvement in PID design and implementation methodology has a serious potential to be used at industrial engineering applications. At industrial applications the PID controllers are preferred widespread due to its robust characteristics against changes at the system model. From the other side at industry, the exact plant models can not be obtained due to too much nonlinear parts and uncertainties so at practice engineers usually find an appropriate model for the dynamic system. For example, when a thermal system is taken into consideration, the system's overall gain changes from season to season. Changes in dynamic system parameters and unknown system variables directly affect the performance of the system. So for obtaining a better performance the controller parameters have to be renewed in some time interval.

A lot of methods have been developed over the last forty years for setting the parameters of a PID controller. Some of these methods are based on characterizing the dynamic response of the dynamic system to be controlled with a first-order model or second-order model with a time delay [4]. All general methods for control design can be applied to PID control. A number of special methods that are tailor made for PID control have also been developed, these methods are often called tuning methods. The most well known tuning methods are those that are stated by Ziegler and Nichols. These methods do not need any mathematical calculation to find PID parameters. The Ziegler-Nichols Oscillation Method, Ziegler-Nichol Process Reaction Method and Frequency Response method, and Cohen-Coon Reaction Curve Method are basic Self-Tuning methods. Oscillation method is based on system gain, in other words, system gain is redounded until the system makes oscillation, then PID parameters can be found from system response graphic. Practically, this method is useless for too many sort of real systems, because oscillation at the output of the system can easily damage the system. Frequency response uses frequency domain rules to find PID parameters. Cohen-Coon method uses system step response for an open loop system to find PID parameters. Also Ziegler and Nichols proposed PID parameters for a group of system due to its system parameter values [1]. In this paper, Ziegler-Nichols process reaction method (PRM) is used to determine PID controller parameters; Kc, Ti and Td.

Industrial systems are controlled by microcontroller based systems in recent years and widely used microcontroller based systems are programmable logic controllers (PLCs). These controllers are more capable than the other micro controller based ones at the design phase of automation systems so the time consumption over the project decreases. Also the elasticity at hardware level and software level let the modifications be done very easily. At the other side nowadays most of PLCs support the popular communication protocols like Profibus, Modbus, Industrial Ethernet, etc. Within this, the integration to SCADA systems gets easy. [8, 12, 13]

In this paper, a self-tuning PID Controller method is offered using Ziegler-Nichols process reaction method for programmable logic controllers. The paper is organized in 6 sections. Section 2 briefly gives a general knowledge about PID controllers. Section 3 describes Ziegler-Nichols process reaction method and the method is formulated here. At section 4 the self tuning PID Controller is presented and the implementation of it within a PLC and the results over this implementation is shown at section 5. Finally the paper is summarized and concluded at section 6.

2 PID Controller

A Proportional-Integral-Derivative controller or PID controller is a common used controller in industrial control applications. The controller compares the measured process output value (y) with the reference setpoint (r) value. The difference or error signal (e) is then processed to calculate the control signal for the manipulated process inputs so the system output reaches the desired reference value. Unlike simpler control algorithms, the PID controller can adjust process inputs based on the history and rate of change of the error signal, which gives more accurate and stable control. In this paper, a different structure of a PID controller is used.

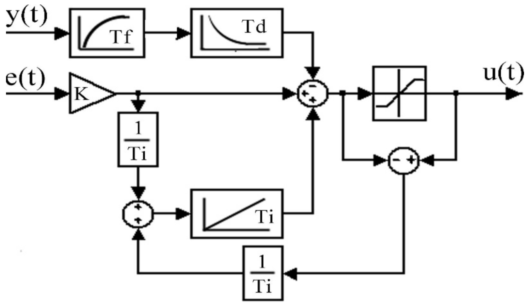

Figure 1: Structure of PID Controller

2.1 The Structure of the PID Controller in Figure 1

As known, the derivative can be computed or obtained if the error varying slowly. Since the response of the derivative to high-frequency inputs is much higher than its response to slowly varying signals [13]. So the derivative output in Figure 1 is smoothened for high-frequency noises by using first order filter, and it uses output of the system (y). The derivative which uses error signal can form high derivative output when the error signal has high frequency components. Thus, in this paper the derivative input uses the filtered output of the system. Here the filter smoothens the signal and suppresses the high-frequency noise due to filter time (Tf) constant (Figure 2). In application, the Tf should be bigger than Ts sampling period [6].

In figure 1, the integral signal is formed by the error multiplied by gain (K) and divided by integral time, and saturation difference divided by integral time. PID controller is a robust controller and this structure puts forward a more robust controller. The saturation component is necessary for discrete time controllers [8]. As said before, this structure is used in a programmable logic controller, and this controller has maximum and minimum borders. The saturation component supplies not to reach any other point except the limit of maximum and minimum borders. Thus, the control signal (u) is limited.

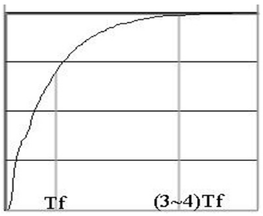

Figure 2: Smoothen of System Output Response

2.2 Mathematics of PID Controller in Figure 1

A commercial PID controller discrete time equation can be given like in (1). Although this type of PID structure is using in many industrial applications, it is sensitive to disturbances and system uncertainties.

$$G_{PID} = K_P + K_I \frac{Ts}{1-z^{-1}} + K_D \frac{1-z^{-1}}{Ts} \tag{1}$$

More robust PID parameters can be given separate to use in figure 1 like in (2). This z-domain equation is obtained form s-domain equation with Tustin transform and trapezoid integral approach for integral term [7].

$$\begin{aligned} G_P &= K_C \\ G_I &= \frac{K_C.Ts.(1+z^{-1})}{2.T_I.(1-z^{-1})} \\ G_D &= \frac{K.T_D.(1-z^{-1})}{Ts} \end{aligned} \tag{2}$$

Discrete time equations of PID can be given to use in Figure 1 like in (3). In these equations, Ts is sampling period, and it can be chosen form 30 times of band width frequency, $w_s = 2\pi / Ts > 30\, w_{BW}$ [5].

$$\begin{aligned} G_P(k) &= K_C.e(kT) \\ G_I(k) &= u(k-1) + \frac{K_C.Ts}{2.T_I}[e(kT) + e(kT-T)] \\ G_D(k) &= \frac{K_C.T_D}{Ts}[e(kT) - e(kT-T)] \end{aligned} \tag{3}$$

These equations are directly adapted to Figure 1 and it can be used in a microcontroller or PLC [9].

3 Ziegler-Nichols Process Reaction Method

Process reaction method is an experimental open-loop tuning method and is only applicable to open-loop stable systems. This method presented by Ziegler and Nichols is based on process information in the form of the open loop step response obtained from a bump test. This method can be viewed as a traditional method based on modeling and control. The Ziegler-Nichols tuning rules were developed to give closed loop systems with good attenuation of load disturbances. The design criterion was quarter amplitude decay ratio, which means that the amplitude of an oscillation should be reduced by a factor of four over a whole period. This corresponds to closed loop poles with a relative damping of about $\zeta = 0.2$, which is too small [1].

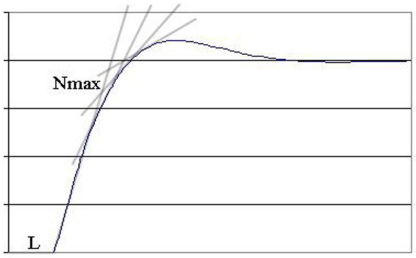

Figure 3: Ziegler-Nichols PRM

3.1 Calculations of PID Parameters Using Ziegler-Nichols Process Reaction Method

This method firstly characterizes the plant by two parameters Nmax and L for first and second order dead time systems and then calculates PID parameters (4). In this paper, Nmax and L are calculated by a PLC algorithm as seen in Figure 5. Here N max is the point of maximum slope and L is the dead time (Figure 3).

$$K_C = 1.2/(L.N_{MAX}) \qquad T_I = 2.L \qquad T_D = L/2 \tag{4}$$

First a step signal is applied to the system and program starts to search the dead time. The dead time is the time when system gives no response to reference signal. In program, a tolerance is given for measuring the dead time (Figure 4), because there are always some high frequencies measuring noises at system output. As shown in Figure 4, these signals and distributions change in an interval defined tolerance. After the dynamic system starts to follow reference and reaches outside the tolerance border, dead time is calculated by PLC program.

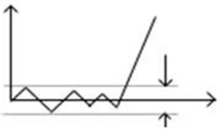

Figure 4: Tolerance Limit

If the dead time is finished or calculated, the program starts to search maximum slope. It collects all slopes and after collecting them, it selects the biggest slope. Every slope is calculated with equation (5).

$$N = \frac{Y(k) - Y(k-1)}{Ts} \tag{5}$$

It memorizes the output value of previous period and takes the output value of the recent period and divides their difference by sampling period [3, 5]. Then the program constitutes data of all slopes and selects the biggest slope. When the maximum slope is calculated, the program waits steady state because the parameters of system are stable in steady state. Finally, the program calculates PID parameters.

To sum up, to calculate PID parameters using Ziegler-Nichols PRM; first gather data from open-loop plant response to unit step input, then examine data set to find the maximum slope (Figure 3), after then determine the parameters needed for Ziegler Nichols PRM, finally, use tuning relations to generate PID constants [12]. The diagram of this process is given in Figure 5.

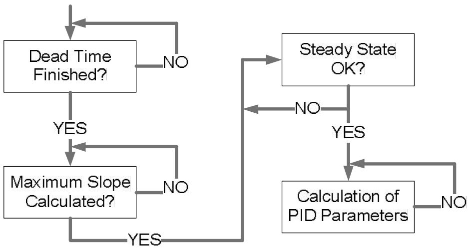

Figure 5: Calculation of PID Parameters

3.2 Robustness of Ziegler-Nichols Method

A good PID controller design should exhibit robustness with respect to small perturbations in the controller coefficients. Thus, the range of L/N_{MAX} values that ensures robustness was determined for Ziegler-Nichols PRM in (6), where τ is system's time constant (without controller) for first order deadtime systems (FODS), and T_{SET} is settling time (without controller) for second order dead time systems (SODS) [3].

$$\begin{aligned} &0 < \frac{L}{\tau} < 1.07 \quad \mathit{for} \quad \mathit{FODS} \\ &0 < \frac{4.L}{T_{SET}} < 1.07 \quad \mathit{for} \quad \mathit{SODS} \end{aligned} \tag{6}$$

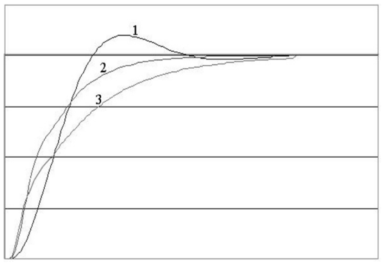

Figure 6: FODSs to the Step Response

$$\begin{aligned} &1^{st}\ \mathit{System}:\ L/\tau = 0.3/1.0 = 0.3 \\ &2^{nd}\ \mathit{System}:\ L/\tau = 0.3/0.6 = 0.5 \\ &3^{rd}\ \mathit{System}:\ L/\tau = 0.3/0.4 = 0.75 \end{aligned} \tag{7}$$

As seen in equations in (7) and figure 6, 2^{nd} system is a more robust system than do 1^{st} and 3^{rd} systems due to L/τ ratio. When L/τ ratio increases from $1.07/2$, system settling time is decreasing and when L/τ ratio decreases from $1.07/2$ system makes overshoot like a second order system, and when L/τ ratio is approximately zero, systems make oscillation [2].

In equations in (8) and Figure 7, 2^{nd} system is a more robust system than do 1^{st} and 3^{rd} systems due to $4L/T_{SET}$ ratio.

As resembling to figure 6, 2^{nd} system has a good performance due to $4L/T_{SET}$ ratio is approximately $1.07/2$.

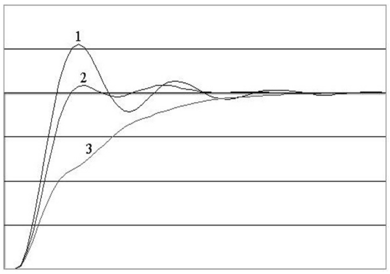

Figure 7: SODSs to the Step Response

$$1^{st}\ System:\ 4.L/T_{SET} = (4)(0.20)/3.05 = 0.26$$
$$2^{nd}\ System:\ 4.L/T_{SET} = (4)(0.20)/1.85 = 0.43 \quad (8)$$
$$3^{rd}\ System:\ 4.L/T_{SET} = (4)(0.20)/1.08 = 0.74$$

From figures 6 and 7, Ziegler-Nichols process reaction method (PRM) always provides a responsible proportional gain for PID controller. This method not only gives good performance but also is robust with respect to controller parameter perturbations [11].

4 Self-Tuning PID Controller

PID parameters must be determined from dynamic system. As said before, system parameters change because of various reasons. If PID controller parameters remain the same for a long time, the dynamic system could not be controlled by PID efficiently. Root locus method, bode-frequency analysis method and some methods like this can be used for this calculation. But these methods have complex mathematical calculations, and also system feedback and system's disturbations can not be measured momentary without any error. In addition, system parameters (like system gain) change due to environmental change. For these reasons, a self-tuning PID controller is a necessity because this type of a controller can be used in different type of systems and environmental situations. Moreover, a self-tuning PID is a robust controller for systems' uncertain parts. Also for changing at system dynamics the controller adopts itself. Thus, using a self-tuning PID is reasonable rather than using any other PID controller which has constant parameters [6].

4.1 Algorithm of Self-Tuning PID Controller and Process

Program algorithm for PLCs is given in Figure 8. The algorithm consists of two start options: one is working with recent parameters which are calculated before; other option is working with new parameters. In this option, program finds new PID parameters for system. Because of Ziegler-Nichols method is applicable for open-loop systems, program first cancels system feedback and waits the system response to settle. When the system output is reset, program records system's momentary input and Then program applies a step signal to system input. It should be said that this step signal is at least 10% bigger than the systems current input (reference) value [11].

If the step signal smaller than 10%, system parameters can not be determined reasonable. After applying the step signal, program waits until the system output to settle at the output value. When the system output is stable, program calculates PID parameters using Ziegler-Nichols process reaction method and sends them to PID parameter input. When PID parameters are loaded, program attaches system feedback and PID controller. Thus, system starts to work with PID controller.

To clarify, necessary steps are given in a sequence below:

- Run the system in open-loop mode, then wait until the system output becomes stable
- Record system input and output, then apply a step input to system (larger than 10% of recent input)
- Wait until the system output becomes stable, then calculate PID parameters and connect feedback and PID controller to the system

4.2 Simulation of Self-Tuning PID Controller in a Programmable Logic Controller

Siemens S7-400 CPU 412-2 DP PLC is used for testing this algorithm, Within the program Simatic Manager Professional a simulation toolbox is given. This simulation toolbox is good for testing algorithm before applying it to a real system. Also a data collection program is written for S7-400 CPU 412-2 DP which collects datas and transfers them to Microsoft Excel [10]. In addition, the Figures 2, 6 and 7 are obtained from this data collection program.

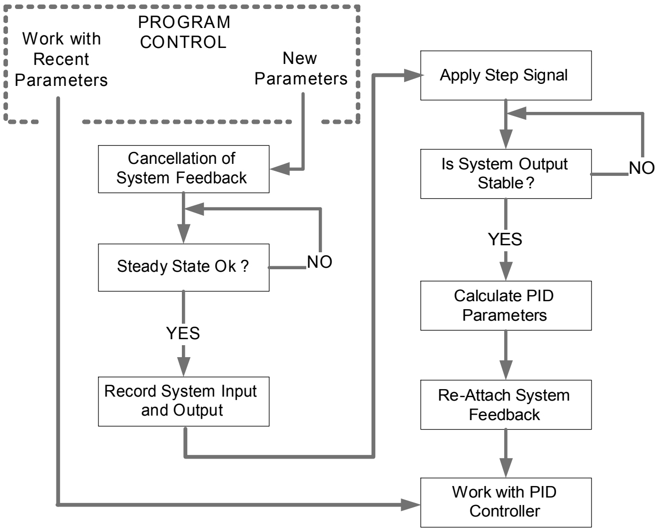

Figure 8: Algorithm of Main PLC Program

Figure 9 explains self-tuning operation for a first order dead-time system (FODS). In this system, L/τ ratio is 0.4 where dead time (L) is 0.4 seconds and system time constant is 1.0 seconds. As seen in the figure, system waits 0.4 seconds and then starts to pursuit step response (point 3

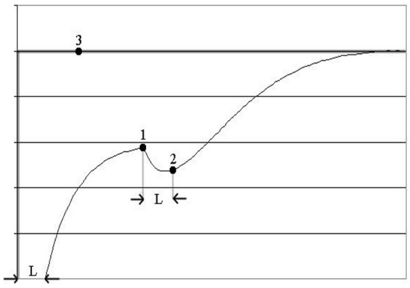

Figure 9: Self-Tuning

states step response). Then PLC program determines system characteristics, and find PID parameters using Ziegler-Nichols PRM in point 1. After finding the parameters, the PLC loads these parameters to PID controller. Because 0.4 second dead time system waits until point 2, and then PID controller starts to control system in point 2. Thus, system makes no steady state error and then pursuits step response like in Figure 10. In this figure, point 1 states square step input and point 2 states system's response (system has self-tuning PID parameters – found in Figure 9).

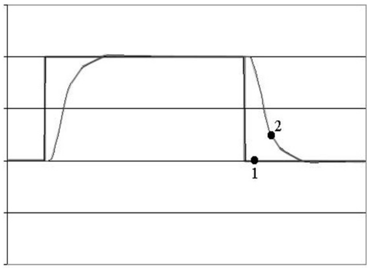

Figure 10: System Response with Self-Tuning PID parameters (Response to Square Step Input)

5 Self-Tuning PID Controlled Industrial Thermal System

After simulating Self-Tuning PID, the controller is applied to an industrial thermal system. The steps explained in detail in 5.1, 5.2 and results are given in 5.3.

5.1 Thermal System

This system can be represented as in Figure 11. In this figure, u(t), y(t), Rt, Ct, Ti and To represents control signal, system output, thermal resistor, thermal capacity, thermal output and thermal disturbances respectively [13]. This system's differential equation can be given in (9).

$$q(t) = C_t \frac{d\left(T_i(t) - T_0(t)\right)}{dT} + \frac{T_i(t) - T_0(t)}{R_t} \quad (9)$$

After Laplace transform, the s-domain equation can be given in (10),

$$Q(s) = T_i(s) + \left(\frac{1}{R_t} + C_t s\right) T_0(s) \quad (10)$$

and the transfer function of this thermal system is,

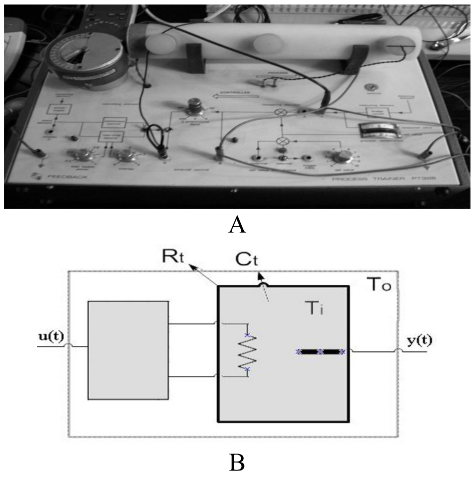

Figure 11: A – Thermal System
B - Representation of Thermal System

$$T(s) = \frac{Rt}{Rt.Ct.s + 1} \quad (11)$$

Thus, in this paper self-tuning PID controller is applied to a first order dead time industrial thermal system. There is no need to calculate Rt and Ct, because self-tuning algorithm will find its parameters for PID without Rt and Ct.

5.2 Connecting System to PLC

PLC needs two input, one of them is system output (or feedback), and reference signal, due to Self-Tuning PID controller program. Then program produces control signal for thermal system. Reference and system output is connecting to PLC's analog inputs and control signal is sent from analog output to system. The connections are given in Figure 12.

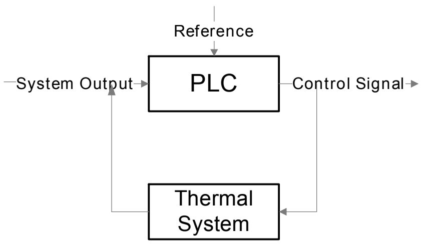

Figure 12: Connections of Thermal System to PLC

5.3 Self-Tuning PID Controller Results

The self-tuning PID simulation is given in Figure 9, and thermal system self-tuning PID response is obtained in Figure 13. As seen in the figure, system waits until dead time finishes and then starts to pursuit step response. PLC program determines system characteristics, and find PID parameters using Ziegler-Nichols PRM. Then, PID controller is started working and system is controlling with PID controller. The simulation result and the real system result is the same. Simulation of system response to square step input is given in Figure 10, and thermal system response to square step input is given in Figure 14. L/τ ratio for this system is 0.29 (Dead time is 0.2 seconds and system time constant is 0.7 seconds). Thus, $0< L/\tau <1.07$. As seen in Figure 14, Self-Tuning PID controller has a good performance because of L/τ's interval. It can be seen from Figure 14. Results are summarized in Table 1.

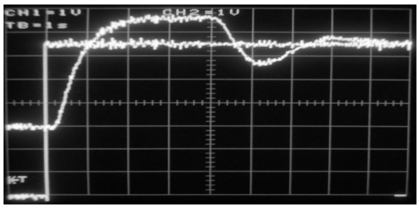

Figure 13: Self-Tuning Analysis in Thermal System

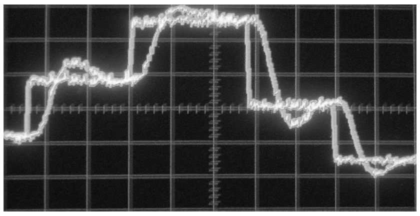

Figure 14: Thermal System Response with Self-Tuning PID parameters (Response to Square Step Input)

System Type	:	FODS
System Dead Time	:	0.2 sec
System Time Constant	:	0.7 sec
Expected Settling Time	:	2.1~3.5sec
PID Gain (K)	:	0.714
PID Integral Time (Ti)	:	0.400
PID Derivative Time (Td)	:	0.100
Settling Time	:	2.2 sec
Overshoot	:	15%
L/τ ratio	:	$0 < 0.29 < 1.07$

Table 1: Self Tuning PID Controller Results for a Thermal System

6 Conclusions

In this article the Ziegler-Nichols process reaction method (PRM) based self-tuning PID controller is presented and its application on a programmable logic controller is given. For this purpose first of all at the implementation part industrial PID algorithm is used where PID's derivative input is taken from system output and filtered, so high-frequency signals' effect is minimized. Then, integral term is confirmed to obtain a more robust PID structure and finally the output of PID is limited due to PLC's maximum and minimum range. Secondly, Ziegler-Nichols method is given and together within robustness definition is defined. It can be seen that most industrial systems are in the group of this robustness limit, and due to fact that most industrial systems are usually modeled first order or second order dead time system at practice. Ziegler-Nichols PRM is simulated in section 4 for these types of systems and get in touch with robustness definition.

For implementing the developed algorithm a Siemens S7-400 CPU 412-2 DP PLC is selected as a controller due to its good performance and its developed structure. Afterwards the developed PLC algorithm is tested on an industrial thermal system. The results showed that self-tuning PID controller has a good performance on these kind of industrial systems. The simulation results and real system results are compatible.

As a result in this work, PID application and system simulation blocks are obtained for a general use in other industrial systems.

References

[1] Astrom K. J., Hagglund T., *PID Controllers: Theory, Design and Tuning,* Instrument Society of America, Research Triangle Park, NC, 1995.

[2] Atherton D. P., Majhi S., *Limitations of PID Controllers,* Proceedings of the American Control Conference, pp. 3843-3847, June 1999.

[3] Friedland B., *Advanced Control System Design,* Prentice-Hall International Editions, 2001.

[4] Hagglund T., *A Dead Time Compansating Three Term Controller,* 9th IFAC/IFORS Symp. On Identification and System Parameter Estimation, Budapest, Hungary, 1991.

[5] Kurtulan S., *Programmable Logic Controllers and Applications,* Bilesim, 1996.

[6] Marshall, J. E., Gorecki, H., Korytowski, A., Walton, K., *Time-Delay Systems: Stability and Performance Criteria with Applications,* Ellis Horwood, NY, 1992.

[7] Middleton R. H., Goodwin G. C., *Digital Control and Estimation: A Unified Approach,* Prentice-Hall, Englewood Cliffs, 1990.

[8] Ogata, K., *Discrete-Time Control Systems,* Prentice-Hall, Englewood Cliffs, NJ, 1995.

[9] Olsson G., Pianni G., *Computer Systems for Automation and Control,* Prentice Hall, 1992.

[10] Siemens, *Simatic S7-300 / 400 Programmable Controller System Manuel,* 2000.

[11] Silva G., Datta A., Bhattacharyya S. P., *PID Controllers for Time-Delay Systems*, Birkhauser Boston, 2005.

[12] Silva G., Datta A., Bhattacharyya S. P., *On the Stability and Controller Robustness of Some Popular PID Tuning Rules*, IEEE Transactions on Automatic Control, Vol.48, pp. 1638-1641, 2003.

[13] Yücelen T., Kaymakçı Ö., Kurtulan S., *Digital PID Controller Design with PLC for an Industrial Thermal System,* 3rd Automation Symposium, Denizli, Turkey, 2005.