IMPROVING THE PRECISION OF PLANT RESPONSE BY MODELING THE STEADY STATE ERROR

MSc. Dushko Stavrov, Assist. Prof Goran Stojanovski, Prof. DSc Stojche Deskovski,

Faculty of Electrical Engineering and Information Technology – University of Ss Cyril and Methodius in Skopje, Republic of Macedonia

dushko.stavrov@feit.ukim.edu.com.mk

Abstract: *Nowadays, one of the most common problems in control system theory that should be tackled is how to improve the precision of a plant in steady state, under a change in the target value of the plant. Well known fact is that the models we use for designing controllers are not ideal. Thus, when the controller is applied to the real plant there is difference in between the expected and obtained results. Likewise, the controllers should be designed to be at the same time robust to uncertainties and also fast enough to drive the system to the desired value. The purpose of this paper is to describe and finally implement the approach in which the idea is to improve the precision of the system in steady state by adding an additive term to the control value calculated by the predesigned PID controller. The PID controller is designed in advance, and has poorly tuned integral term. Afterwards, when the desired target value is changed the PID controller is not aware of that change, so its performance starts to drop and as a result the steady state error starts to increase. Therefore, to preserve the exactness of the plant's output an additive term to the control signal is calculated out of a polynomial second order model derived from the error values obtained in the previous measurements of the plant. The results from MATLAB simulations have shown that the PID controller could not keep up good performance when the target value of the system is changed. Hence, by adding an additive term to the control signal we gave to PID the needed 'awareness' and as a result of that we could improve the steady state error by small margin.*

KEYWORDS: PID CONTROLLER, ADDITIVE TERM, POLYNOMIAL SECOND ORDER, EXACTNESS,

1. Introduction

In control literature, one can easily find a variety of different examples for industrial control, where contemporary control algorithms are implemented. Surprisingly, there are not much known examples where the state-of-art control algorithms have been implemented in real-time control systems, like for example: missiles, jets, drones, robots, etc. Instead as control techniques for such systems researchers usually implement algorithms that are proven to be reliable, fast and easy to implement. In the light of this discussion we can add that nowadays control algorithms are not a single or stand-alone solution, like the basic PID controller is. Instead contemporary controllers are supported by a bundle of additional procedures. For example, the MPC algorithm ([6], [7]) which is considered as main candidate to replace the much simpler, wholly grail of industrial control - the PID controller, uses a lot of background computations to generate the control signal.

The PID controller is in the heart of control engineering practice for more than seven decades. The PID controller is one of the most commonly used controllers in industry. Some of the reports show that PID controllers are being used in 90-95% of the control loops in industry ([1], [2]). Its simple structure has made PID controller one of the most widespread controllers in all technical systems. Over the long history of its use and development, the simple notation of PID control mechanism has been augmented with new features that aim to improve its efficiency. However, the key question that many scientists try to solve is - which is the best procedure to tune the PID parameters, in order to achieve the desired control object performance. One should mention that one of the most broadly used method of computing the PID coefficients, in industry, is Ziegler Nichols method ([3]). However, we should make a notice of a reference of the tuning of PI and PID controllers ([4], [5]) whose second edition published in 2006, shows that there are more than 400 versatile methods of PID synthesis. Even though there are a lot of synthesis methods of PID controller, some reports say that around 80% of the PID controllers are poorly tuned, where 30% of the the PID controllers operate in manual mode [9].

Although there are a lot of advantages of PID control, it cannot be successfully used when dealing with system with drifting parameters. It is well known that the industrial plants, are subject to change in time and the possibility of parameter drift in the plant drastically increases as the plant is being operated. We interpret this as parameter uncertainties in the control object which usually leads to worse performance of the control system. Hence, if the PID controller was initially designed to work for a particular operating point, after the parameters drift, the controller should be adapted to the new operating conditions. If there is not some kind of supervisory system that automatically takes care of the adaptation, we should track the parameter drift and occasionally tune the PID parameters.

The above mentioned problem of parameters drift can be solved with adaptive control algorithms, which are making continuous or periodic corrections in the PID coefficients [12]. The problem that arises in this situation is the speed of adaptation of the coefficients. Surely, we want to reduce the time of adaptation to the possible minimum.

Nevertheless, there are other possibilities for correction of the effects derived from the parameter drift. In this paper, we assume that the parameters are already obtained using trial and error. Therefore, we propose an improvement to PID controller in form of an additive term to the PID control value, as $u_{PID} + \Delta u$, aiming to improve the precision of the control object in the steady state. This is of great importance in control systems in chemical industry and manufacturing plants, where the precision in steady state is of great concern for safety and as well as for cost effectiveness.

Moreover, the additive term is calculated as a root of the quadratic polynomial model which is modelled out of the set of previously stored values of the steady state error and additive term values. In mathematical terms the error model is given by:

$$E_M = f(\Delta u). \quad (1)$$

The rest of the paper is structured as follows. Firstly, the mathematical background of the simple and enhanced PID is presented; then the PID enhancement is discussed in more details. Secondly, the case study of a CSTR control system is modelled. Afterwards, both PIDs are applied to the CSTR system and the obtained results are analyzed. Finally, conclusions and outlook for future work are given.

2. Mathematical formulation of simple PID and enhanced PID

Simple PID formulation

Despite the simplicity of the basic notion of a PID controller, we can distinguish several different forms of implementation of a PID control law. Likewise, in industry various forms of PID controllers are used, more than ten in whole. For more information about different forms of PID realizations see references ([4], [5]).

In this paper we have focused on the simplest PID realization and that is the parallel form. The control signal with this PID form is generated by the following equation:

$$u_{PID}(t) = u_0 + K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{de(t)}{dt}. \quad (2)$$

Where, u_0 is a bias in the control signal and $e(t) = SP - PV$ is the current error, which is calculated as difference between set point (SP) also known as reference value and the process value (PV). The coefficients of the PID are, K_p- the proportional term, K_i - the integral term and finally K_d is the derivative term. By any means the generation of the control signal is done very fast and the control value only depends on the current as well as the past values on the error. On Figure 1, the control loop consisted of PID controller and control object is shown.

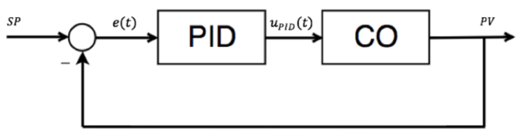

Figure 1 *PID control loop.*

Enhanced PID formulation

As we mentioned before, in the introduction, the simple PID controller doesn't have awareness of how good its parameters had been tuned. Accordingly, in this paper we have tried to give the needed awareness to the PID with the objective to deal with error in steady state as well as to improve the time needed to get in steady state. The principle schematic representation of the approach is given on the Figure 2.

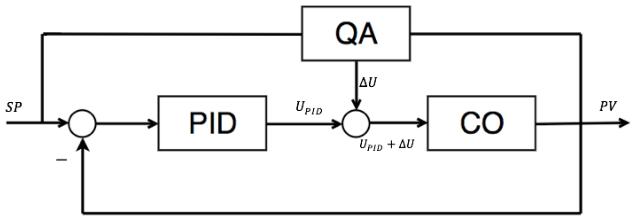

Figure 2 *PID control loop.*

On Figure 2 the change we have done on the simple PID structure is shown. As opposed to what is on the Figure 1 we can see that on Figure 2 there is an additional component named QA. It is an abbreviation of the word Quadratic Approximation. The main idea is, as the control system operates, in the background to have some kind of supervisory mechanism which principle purpose would be to model the steady state error in the system. Afterwards the same model presented with equation (1) will be exploited as apparatus out of which the additive term Δu will be calculated.

The whole process of plant control, calculation of the PID control value in addition to additive term is given with the flow chart on Figure 3. Where N_s is the number of simulation steps of the plant (control object). As long as the counter $i \leq N_s$ the plant is controlled in as presented on Figure 2. When the condition given with equation, (3) is true, the steady state error defined with the equation (4) is calculated and stored. The name, mod stands for function which gives information whether in division between i and DV there is residuum or not. If the residuum is zero that means that i is divisible with the number DV. The index j in the brackets, in (4), indicates how many times the condition given with (3), was fulfilled and in the same time, it gives the number of collected steady state error points which after that will be used for designing a quadratic model, equation (1). The variable PV_{ss} stands for steady state value of the process value. Further, DV stands for the Dynamical Variable, which defines on how many simulation steps an error point should be collected.

$$mod(i, DV) = 0, \quad (3)$$

$$E_{ss}(j) = SP - PV_{ss} \quad (4)$$

The next step is to check whether $j \geq 6$, if it is not, then a simple metric is used for producing the points needed for quadratic model fitting.

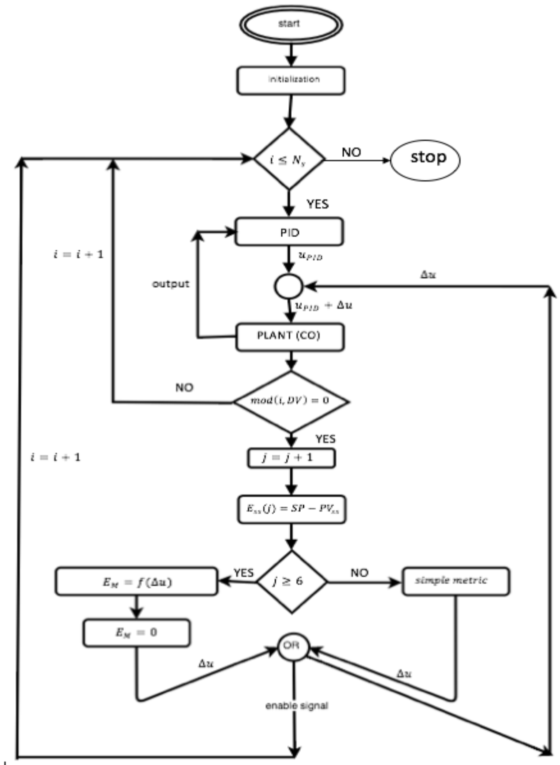

Figure 3 *Flow chart diagram of proposed algorithm.*

The metric that has been used is,

$$\Delta u = k_{ss} \sum_{m=1}^{j} E_{ss}(m). \quad (5)$$

where the parameter $\boldsymbol{k_{ss}}$ is obtained using the trial and error method. Afterwards, when 6 points are gathered, it is easy to fit a quadratic model like the one given with (6). To fit the model (6) we only need 3 points ([10], [11]). In other words, the sufficient number of points is equal to the number of unknown parameters. In this paper the model was dynamically generated out of the last 6 steady state error points. The equation of the model is given by***:***

$$E_M(\Delta u) = A(\Delta u)^2 + B(\Delta u) + C \quad (6)$$

The parameters A, B and C of the model are calculated by solving the next equation:

$$p = (M^T M)^{-1} M^{-1} \cdot J \quad (7)$$

Where $p \in \mathbb{R}^3$ is a vector of parameters,

$$p = [A, B, C]^T, \quad (8)$$

$M \in \mathbb{R}^{6\times3}$ is a matrix of 6 points, which are considered to determine the parameters p.

$$M = \begin{bmatrix} (\Delta u_1)^2 & \Delta u_1 & 1 \\ \vdots & \vdots & \vdots \\ (\Delta u_6)^2 & \Delta u_6 & 1 \end{bmatrix} \quad (9)$$

At last, $J \in \mathbb{R}^6$ denotes the vector of error values. Equation (7) can be solved if the inverse $(M^T M)^{-1}$ exists, which means that the matrix M should have rank equal to the number of parameters, in our case that number is 3. In other words, the points used for regression have to be distributed in a way that the rank of matrix M is not smaller than 3.

At first sight, 3 points seem to be enough to solve the equation (7). However, there are cases in which the chosen 3 points are not suitable. First of all, it is clear that two points should not be placed in the same location which leads to a reduced rank of M. Furthermore, it has to be ensured that the points are not distributed on a line. Anyway, if that is the case then the information provided by the points is not adequate to describe a quadratic function exactly.

3. Case Study: Nonlinear Non-isothermal Continuous Stirred Tank Reactor

Consider a simple liquid-phase, irreversible chemical reaction where chemical reactant A is converted to product B. The reaction that happens in the reactor can be written as follow $A \rightarrow B$. Also, we assume that the rate of reaction is first-order with respect to reactant A:

$$r = kC_A \tag{10}$$

where r is the rate of reaction of A per unit volume, k is the reaction rate constant and C_A is the molar concentration of reactant A. For a single-phase reaction as we are assuming here, the rate constant is typically a strong function of reaction temperature. The rate constant is given with the equation:

$$k = k_0 e^{-\frac{E}{RT}} \tag{11}$$

where E is the activation energy, R is the gas constant and k_0 is the frequency factor.

The graphical representation of the CSTR is given on Figure 4. The input in the system is the inlet flow which is consisted of reactant A with concentration C_A. Often, the reactions happening in CSTR system have significant heat effects. Thus it is important to be able to add or remove heat from them. Adding or removing heat from reactor depends on the temperature difference between the cooling jacket fluid and the reactor fluid.

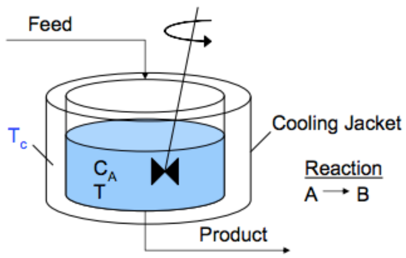

Figure 4 *CSTR system.*

The model of this control system is given by the following two equations [8]:

$$\frac{dT}{dt} = \frac{q}{V}\left(T_f - T\right) + \frac{\Delta H}{\rho C_p} k_0 e^{-\frac{E}{RT}} + \frac{UA}{V\rho C_p}\left(T_c - T\right) \tag{12}$$

$$\frac{dC_A}{dt} = \frac{q}{V}\left(C_{Af} - C_A\right) - k_0 C_A e^{-\frac{E}{RT}} \tag{13}$$

where T_c, the temperature of the cooling jacket fluid, is the manipulated variable and T, reactor temperature, is the controlled variable. Production of the desired component concentration depends on coolant flow rate, reactor temperature and reaction rate. It is assumed here that the cooling jacket flow is fixed. Other parameters values contained in equations (12) and (13) are given in Table 2.

For the purpose of simulating and also solving the equations (12) and (13) MATLAB has been used. More precisely for numerically solving of system equations the ode23t function was employed with integration step of T =0.01 min. The initial conditions of the system are given in the Table 1.

Table 1: *Initial conditions for temperature and concentration*

$\boldsymbol{T(0)}$	$\mathbf{296.6}\ \boldsymbol{K}$
$\boldsymbol{C_A(0)}$	$0.98\ mol/m^3$

Table 2: *Model parameters.*

Parameter	Value	Parameter	Value
Volumetric flowrate $[m^3/sec]$	$q = 100$	Overall heat transfer coefficient $[W/(m^2K)]$	$UA = 5 \cdot 10^4$
Volume of CSTR $[m^3]$	$V = 100$	Feed concentration $[mol/m^3]$	$C_{Af} = 1$
Density of A-B mixture $[kg/m^3]$	$\rho = 1000$	Feed temperature $[K]$	$T_f = 350$
Heat capacity of mixture $[J/(kgK)]$	$C_p = 0.293$	Activation energy $[J/mol]$	$E/R = 8750$
Heat of reaction $[J/mol]$	$\Delta H = 5 \cdot 10^4$	Pre-exponential factor $[1/s]$	$k_0 = 7.2 \cdot 10^{10}$

4. Implementation of the discussed algorithms, PID and enhanced PID and results

In this part we will apply the two PIDs, discussed before, on the CSTR system. First of all, the parameters of both PIDs will be defined. Further, the two algorithms will be simulated in MATLAB, in the fashion given on Figure 1 and Figure 2, where the CO (Control Object) is the highly nonlinear system CSTR. To prove that, the CO is highly nonlinear, we have carried out an open loop simulation. The simulation of the algorithms was carried out over a period of 135 minutes or speaking in simulation steps $N_s = 1350$. The next, Figure 5 shows that at temperature of the 305 of T_c the CSTR system exhibits limit cycle behavior.

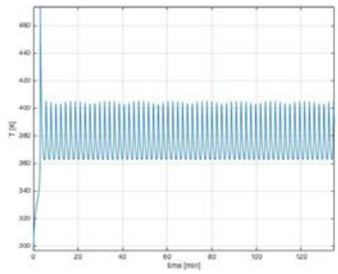

Figure 5 *Open loop simulation of the CSTR system at coolant temperature of 305 K.*

Furthermore, in this paper we assumed that all of the parameter in the system are constant and do not survive drift. Both controllers, simple PID and enhanced PID will be compared in two different scenarios. The first one (Scenario 1) is when the reference value (SP) changes from 300 to 305 K. The second one (Scenario 2) is when the reference value (SP) changes from 300 to 295 K. In both scenarios the responses of the PIDs will be compared. The IAE (Integral Absolute Error) metric given with the equation:

$$IAE = \frac{1}{N_s}\sum_{r=1}^{N_s} |e(r)|, \tag{14}$$

is used to estimate how well one of the controllers performs over the other. We should also mention that the parameters of the two controllers in both scenarios are the same and are given in Table 3,

Table 3: *both PID parameters*

$\boldsymbol{K_p}$	$\mathbf{4.5}$
$\boldsymbol{K_i}$	0.5

K_d	0.04

Let's first consider the Scenario 1, when the SP changes from 300 to 305 K, at 60 minute. On the next figure (Figure 6) are given the responses of both PIDs, the response of the simple PID is given in blue whereas the response of the enhanced PID is given in red.

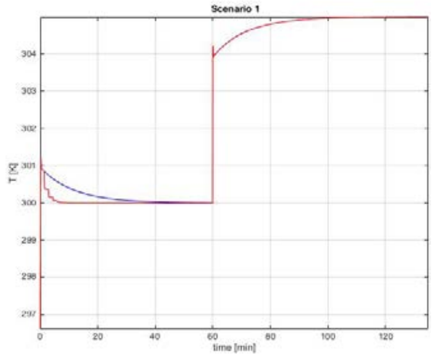

Figure 6 *Response of the CSTR system in cases when it is controlled by simple PID (blue line) and enhanced PID (red line) in Scenario 1.*

From the Figure 6 we can conclude that the overall response of the system has improved. Indication for that is the metric IAE, its values show that the enhanced PID is performing better than the simple PID. The IAE values, for Scenario 1, are given in the Table 4:

Table 4: *IAE values for the Scenario 1*

Simple PID	0.1801
Enhanced PID	0.1156

Let's now consider the Scenario 2, when the SP changes from 300 to 295 K, at 60 minute. On the next figure (Figure 7) are given the responses of both PIDs, the response of the simple PID is given in blue whereas the response of the enhanced PID is given in red.

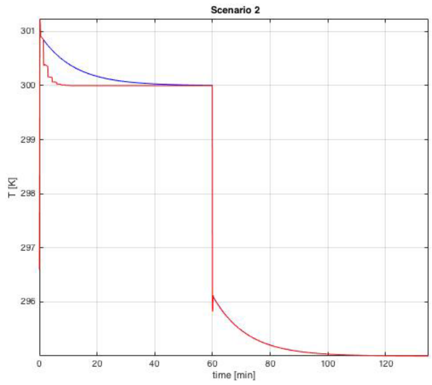

Figure 7 *Response of the CSTR system in cases when it is controlled by simple PID (blue line) and enhanced PID (red line) in Scenario 2.*

Figure 7 indicates that the enhanced PID controller in first phase of the simulation, until 60 minute, is able to converge very fast to the SP value, but after the 60 minute the enhanced PID and simple PID are performing equally bad. That was also the case in the first scenario. It is a problem that indicates that the quadratic model has not ability to adapt very well. A solution for this problem is already being considered; it is thought that the problem might be solved by adding additional information into the model, such as the temperature T or the SP value. However, the values of the IAE metric as in the Scenario 1 led us to the same conclusion in the Scenario 2. The enhanced PID performs better by small margin. The IAE values in Scenario 2 are given in the table that follows:

Table 5: *IAE values for the Scenario 2*

Simple PID	5.5467
Enhanced PID	5.4821

5. Conclusion and outlook for future work

In this paper we present an enhanced PID controller used to compensate for the steady state error. The presented controller is compared with a standard PID controller most commonly used in industry, with poorly tuned integral term. The proposed algorithm uses historical values for the steady state error and the additive control term to create a simple quadratic model of the plant's steady state error. The simulations have shown that the enhanced PID using the additive control term beats the performance of the simple PID by a small margin.

Future work will consist of implementing and afterwards comparing the same controllers presented here, in a case where they are used to control a system, possibly the same one CSTR system, which exhibits drift in some of the parameters. In such conditions, it is expected that the proposed control approach will have superior performances over the standard PID.

ACKNOWLEDGEMENTS

The work was funded by the Faculty of Electrical Engineering and Information Technologies in Skopje, Republic of Macedonia, through the ERESCOP Project.

REFERENCES

[1] Astrom, K. J. and Hagglund, T., Advanced PID control, North Carolina: ISA, 2006.

[2] Rotach, V.Ya, Teoriya avtomaticheskogo upravleniya: uchebnik dlya VUZov (Automated Control Theory: Univeristy Textbook), Moscow: MEI Publishing House, 2008.

[3] Ziegler, J.G. and Nichols, N.B., Optimum Settings for automatic controllers, Trans. ASME, 1942 vol.64, pp 759-768.

[4] O'Dwayer, A., Handbook of PI and PID controller Tuning Rules, London: Imperial College Press, 2009 3th ed.

[5] O'Dwayer, A., Handbook of PI and PID controller Tuning Rules, London: Imperial College Press, 2006 2th ed.

[6] Badwe, A. S., R. D. Gudi, R. S. Patwardhan, S. L. Shah, and S. C. Patwardhan, Detection of Model-Plant Mismatch in MPC applications, J. Process Control, 19, 1305 (2009)

[7] Camacho, E. F., and C. Bordons, Model Predictive Control 2nd ed., Springer-Verlag, New York, 2003

[8] D.E. Seborg, D. A. Mellichamp, T. F. Edgar, and F. J. Doyle III, Process dynamics and control: John Wiley & Sons, 2010.

[9] P. Van Overschee and B. De Moor, in Preprints, Proc. PID '00: IFAC Workshop, Terrassa, Spain, pp. 687-692, 2000

[10] G. Stojanovski, L. Maxeiner, S. Kramer, S. Engell, "Real-time Shared Resource Allocation by Price Coordination in an Integrated Petrochemical Site", 2015 European Control Conference (ECC), pp. 1492-1497, 2015.

[11] S. Wenzel, R. Paulen, S. Kramer, B. Beisheim, S. Engell, "Shared Resource Allocation in an Integrated Petrochemical Site by Price-based Coordination Using Quadratic Approximation", 2016 European Control Conference (ECC), June 29 – July 1, 2016 Aalborg, Denmark.

[12] Rania A. Fahmy, R. I. Badr, Farouk A. Rahman, "Adaptive PID Controller Using RLS for SISO Stable and Unstable systems", Advances in Power Electronics, Volume 2014, Article ID 507142.