Validity of several tuning methods for different PID algorithms

Farah Saleena Taip[1], Ming T. Tham[2]
[1]Department of Process and Food Engineering, , Universiti Putra Malaysia 43400 Serdang, Selangor
[2] School of Chemical Engineering and Advanced Materials, Newcastle University, , NE1 7RU, United Kingdom
Email: saleena@eng.upm.edu.my

ABSTRACT

This article provides a brief review of a number of tuning methods, some old and others more recent. The characteristics of PID controllers tuned using these approaches are evaluated by application to simulated FOPTD processes with different time-delay to time-constant ratios. Different measures were used to assess their performance and robustness properties, and the applicability of the tuning relationships to more typical (non-ideal) PID controllers is also considered. As expected, the more recently proposed tuning methods provide better performance and robustness. However, not all tuning methods are suitable for different structures of PID algorithms, as some of them do not provide acceptable responses.

***Keywords**: PID controllers, controller tuning, tuning comparison*

INTRODUCTION

The PID controller remains the most popular control algorithm used in industry despite the continuous advances in control theory. It has a simple and easily understood structure but at the same time, can provide excellent control performance over a wide range of dynamic characteristics. A PID controller consists of proportional (P), integral (I), and derivative (D) terms, each having an associated parameter that will influence closed-loop system response. Most loops deploy PI control however, as derivative action is not used very often. This makes the tuning of a PI controller less complex, as only two parameters need to be selected [1,2].
Controllers are tuned to minimize or eliminate offset; to minimize the effect of disturbances; to ensure and maintain stability; and to provide smooth and rapid response. In practice, however, it is difficult to achieve simultaneously all the above objectives. Tradeoffs often have to be made so that load and setpoint performances are acceptable whilst robustness is maintained [1-4]. Nevertheless, much work has been directed at developing the tuning relationships that will best alleviate this problem. Which of these tuning methods will give the "best" response? Are they applicable to non-ideal PID structures? This paper seeks to provide some answers to these questions.There is only one form of PI controller. PID controllers, however, can have different structures.

Ideal PID

The PID algorithm considered in most publications is the "ideal PID" which has the following transfer function:

$$\frac{U(s)}{E(s)} = G_C(s) = K_C\left(1 + \frac{1}{T_I s} + sT_D\right) \tag{1}$$

PID controllers used in industry may not have the same structure though [5-8].

Series PID

There is a slightly different version of the PID controller, known as the "series" or "interacting" controller.

$$G_C'(s) = K_C'\left(1 + \frac{1}{sT_I'}\right)(1 + sT_D') \tag{2}$$

It is called interacting as the derivative and integral terms interact with each other [5-8]. The ideal and interacting PID controllers are related via the following expressions:

"Commercial" PID

The derivative term in Eq. 1 causes realization problems, and a more practical form is:

$$G_C(s) = K_C\left(1 + \frac{1}{T_I s} + \frac{T_D s}{1 + sT_D / N}\right) \tag{3}$$

The derivative term in Eq. 1 is cascaded with a low-pass filter with a time-constant, T_D / N . N is usually chosen to be between 5 to 20. The sensitivity of the algorithm to noise is increased with higher values of *N*. [5-8].

Setpoint Weighted or Output Filtered PID

Normally, a PID controller is driven by the error between the setpoint and the controlled output. However there is a more flexible structure given by:

$$U(s) = K_C[(bX(s) - Y(s)) + \frac{1}{sT_I}(X(s) - Y(s)) + \frac{T_D s}{1 + sT_D / N}(cX(s) - Y(s))] \tag{4}$$

Here, the responses to setpoint changes depend very much on the values of *b* and *c*, which are either “0” or “1”. By setting them equal to zero, “kicks” in the controller output are avoided when there is large step-change in setpoint [5-7].

Tuning Methods

This paper considers 6 tuning methods and they are as below. Only the salient features will be highlighted as details of the techniques can be found in the associated references. The methods are Ziegler Nichols (ZN), Cohen-Coon (CC), Direct Synthesis (DS), Abbas method (AA), Simplified IMC (SIMC) and Gani Phase Margin method (GPM).

Arguably the best known of PID tuning techniques is the ZN method. The original proposition is a closed-loop tuning procedure: place the system under proportional control and determine the gain that will yield a marginally stable closed-loop. This critical gain and corresponding frequency are then used in formulae to obtain P, PI or PID settings. However, it is very undesirable to deliberately cause cycling in a production plant, and fortunately, there is a set of open-loop ZN tuning relationships [3,4,10,11]. The CC method was developed empirically to overcome this [3,12,13]. Both the ZN and CC tuning methods can be considered as “recipe” based approaches. They depend solely on the parameters of the process model with no provision to counter the effects of process-model mismatch in the implementation of the final controller. Most modern approaches to controller tuning, however, provide this flexibility. The DS method is one of them and has a tuning parameter that corresponds directly to the closed-loop time-constant. [3,4,10,14].

The Abbas [15] method is very similar to DS with Pade approximation for the time-delay. However, Abbas took the development further by relating the loop gain, *K*, to the overshoot of closed-loop, *V*, and to the process time-delay to time-constant ratio, *R*. Therefore, instead of λ, the tuning parameter that needs to be specified is *V*. This method only holds for overshoot between 5 to 20%. Internal model control is another parametric model-based design method and an extension of DS. Theoretically, it can yield perfect load disturbance rejection as well as setpoint tracking, if the process model is perfect and the inverse of the model is physically realizable [3,4,10,11,12]. Skogestad [9] proposed a simpler set of design relationships using the controller. His aim was to obtain effective tuning relations that are easy to remember. After rigorous simulation studies, he concluded that the best responses were obtained by setting τ_f equal to θ and it is called Simplified IMC (SIMC). Designing controllers in the frequency domain is another popular approach because it is applicable to processes of any order and dynamics [16-20]. A key feature of the Gain Phase Margin method (GPM) is that the method allows robustness properties to be specified explicitly: the controller is usually tuned to satisfy phase margin, P_m, and gain margin, A_m, specifications.

Simulation Studies

Only setpoint tracking was considered and all the tuning methods were tested on three different simulated processes, each with different time-delay to time-constant ratios, *R*:

Process 1 (P1), $G_{p1}(s) = \frac{e^{-2s}}{4s+1}$, *R* = 0.5; Process 2 (P2), $G_{p2}(s) = \frac{e^{-4s}}{4s+1}$, *R* = 1 and Process 3 (P3), $G_{p3}(s) = \frac{e^{-8s}}{4s+1}$ *R* = 2

RESULT AND DISCUCCIONS

Comparing results based on a single criterion can be misleading. Therefore, in this study, the Integral of Absolute Error (IAE); percentage overshoot, settling time, A_m, P_m and the delay margin, D_m, were used. The results are summarised in Figures 1 to 6. Each of these figures contains 3 main groups, corresponding to results obtained from the control of processes P1 to P3. The sub-groups depict the results obtained from DS, AA, ZN, CC, SIMC and GPM tuning methods, in that order. The bars in each sub-group correspond to the PI, ideal-PID, series-PID, commercial PID and output filtered PID controllers respectively.

Performance

The measures relating to performance are the IAE, percentage overshoot and settling time. These are shown in Figures 1 to 3 respectively. Figure 1 shows that performances degrade with increasing values of *R*, which is expected. Within each process category, the highest IAE values can generally be attributed to CC and ZN tuning methods. ZN tuned PI controllers performed particularly poorly for *R*=1 and *R*=2. Despite their different structures, PID controllers gave similar results in each process groups. This implies that the different PID formulations could be tuned using the various tuning relations without severe performance degradation, at least in terms of the IAE measure.

A system may generate low IAE values, but its response may exhibit intolerably high overshoots and vice versa (compare Figure 1 with Figure 2). Low IAE values were recorded because of the shorter settling times (Figure 3). This is illustrated especially by results obtained from applying the ZN tuned PI controller to process P3. Although there was no overshoot, i.e. over-damped response, it had the longest settling time, resulting in the largest IAE value recorded. Of all the tuning techniques considered, the AA method is the only one that depends on an overshoot specification. At *R*=0.5, with the exception of the series PID algorithm, the other controllers yield responses with overshoots quite close to the specified value of 5%. At other values of *R* however, only the AA tuned PI controller gave responses with overshoots that matched the specification. The results indicate that the AA tuning method is generally applicable to PI controllers only. If used to tune PID controllers, the results will be unpredictable especially for processes with high time-delay to time-constant ratios. However, the pattern of results in Figure 3 does not facilitate generalisations. ZN and CC tuned controllers tend to give high overshoots for systems with *R*=0.5 and *R*=1.0, particularly for the commercial and series PID algorithms. For process P3 where *R*=2, the DS tuned controller gave rise to the highest overshoots. The tuning method that gave the most consistent performance across the different controller structures was the GPM method. In Figures 1 to 3, all GPM tuned controllers gave very similar results for each process type, regardless of the controller structures.

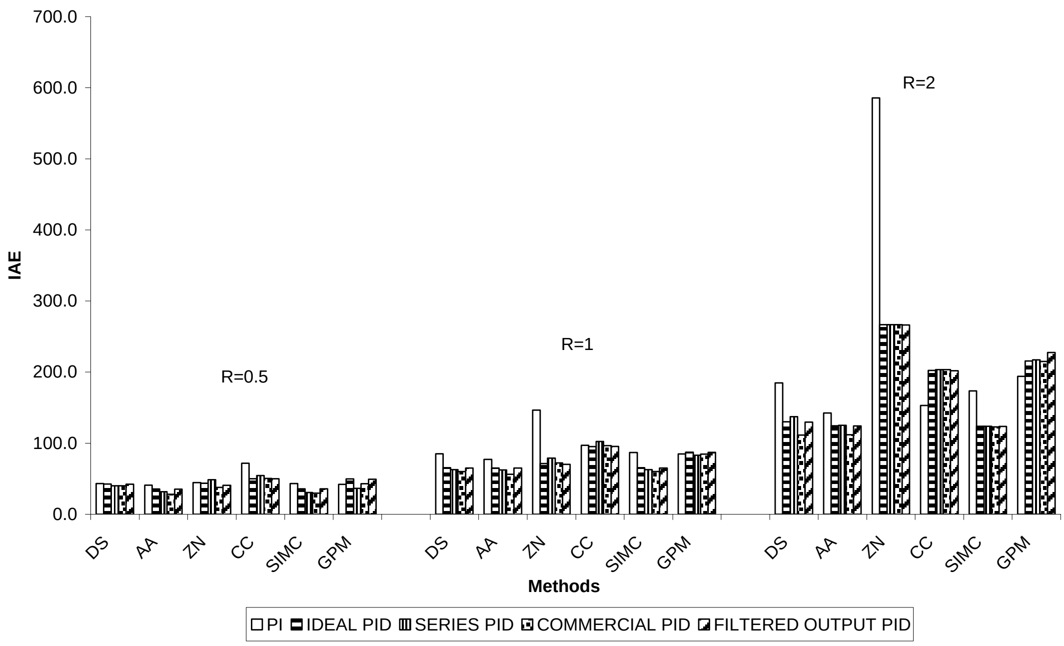

Figure 1 - IAE Values

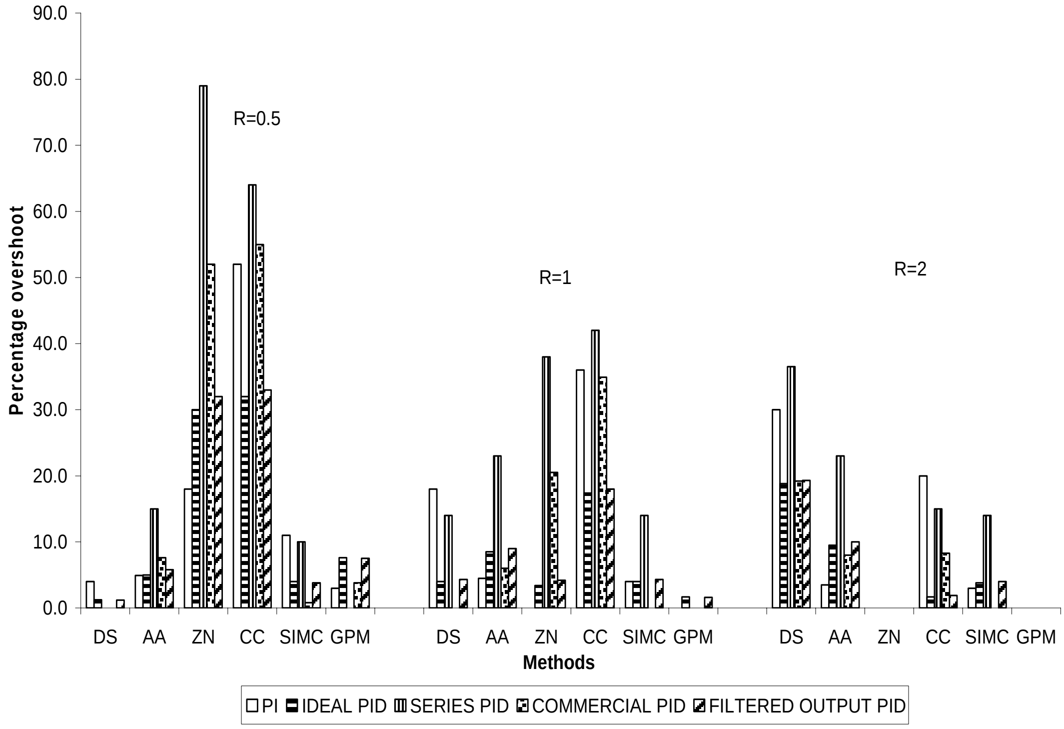

Figure 2 - Percentage Overshoots

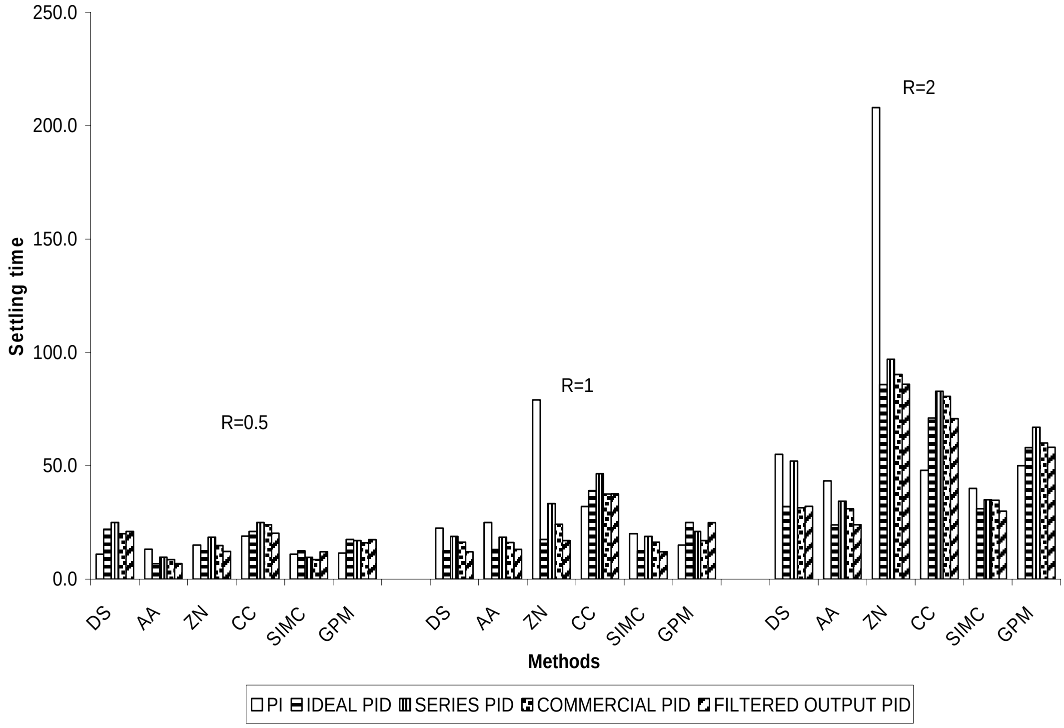

Figure 3 - Settling Times

Robustness

The robustness/relative stability metrics obtained using the different tuning techniques and controller structures are plotted in Figures 4 to 6, showing the gain margin, A_m, phase margin, P_m, and the delay margin, D_m, respectively.

Discounting the GPM method for the moment, where A_m is the design specification, Figure 4 shows that across the three processes considered, SIMC tuning provided consistently the most robust designs in the sense that similar results were obtained using the same settings with different controller structures. Indeed, SIMC tuned PI controllers gave identical A_m and P_m (Figure 5) values regardless of the controlled process, a result that can be proved analytically. Like the SIMC, the GPM method also gave consistent results across the board. However, note that the gain margin values for PID controllers were not equal to the design specification of A_m =3. Because of the phase advance imparted by the derivative term, the phase and delay margins were higher than those obtained for the corresponding PI cases. With a specification of A_m =3, GPM tuning gave systems with higher gain margins than those tuned using SIMC, but similar phase margins obtained for processes P1 (R=0.5) and P2 (R=1.0).

Figure 7 and Figure 8 show the time responses of SIMC and GPM tuned controllers when applied to P3. Figure 7 shows the setpoint tracking responses of the different controller types, tuned using SIMC rules. A range of response characteristics can be observed. Note especially the erratic trajectory of the system when controlled by the series PID algorithm. This is due to the nature of the zeros of the close-loop transfer function, brought about by the structure of this particular controller. To the uninitiated, such response characteristics could be quite disconcerting, and may lead to a decrease in confidence, in either the tuning method or the controller type. The same set of controllers, tuned using the GPM method gave rise to the responses shown in Figure 8. The series PID algorithm again gave the least smooth response, although it not as pronounced as in the SIMC case. Nevertheless, using the GPM tuning procedure, despite the different controller forms, the responses obtained are very similar. Therefore, it is suggested that practitioners would be more comfortable using GPM tuning rather than the SIMC rules.

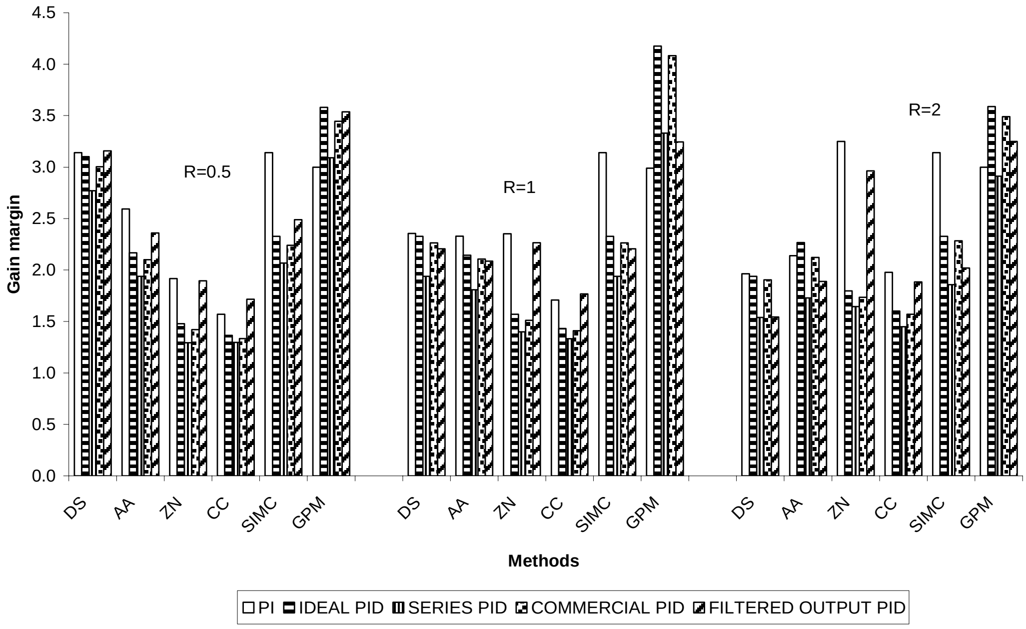

Figure 4 - Gain Margins

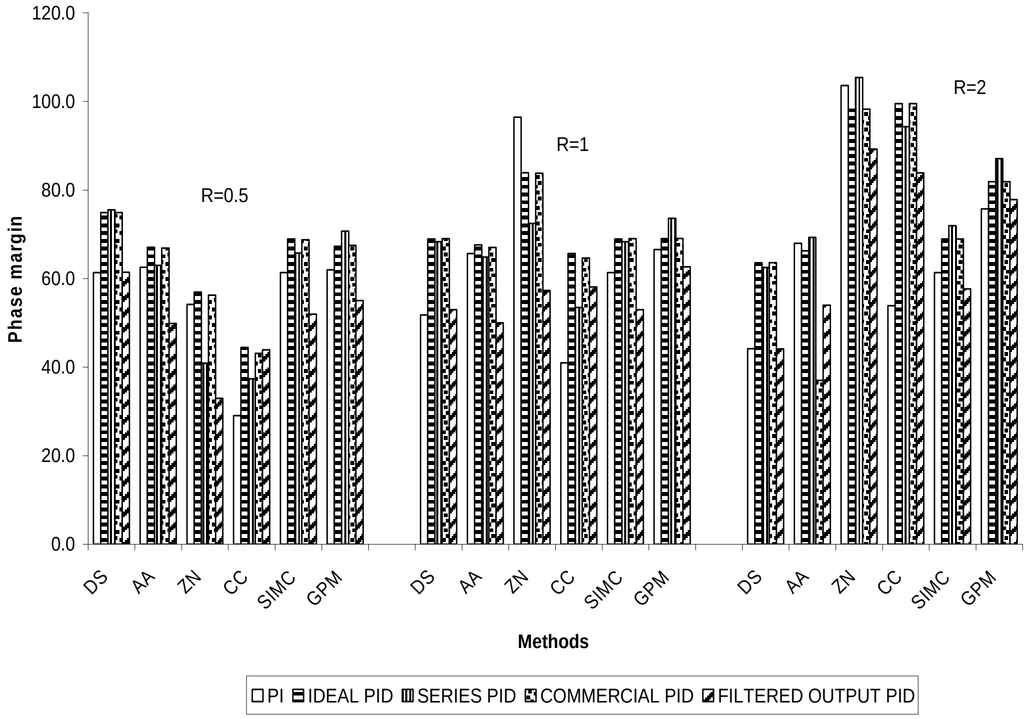

Figure 5 - Phase Margins

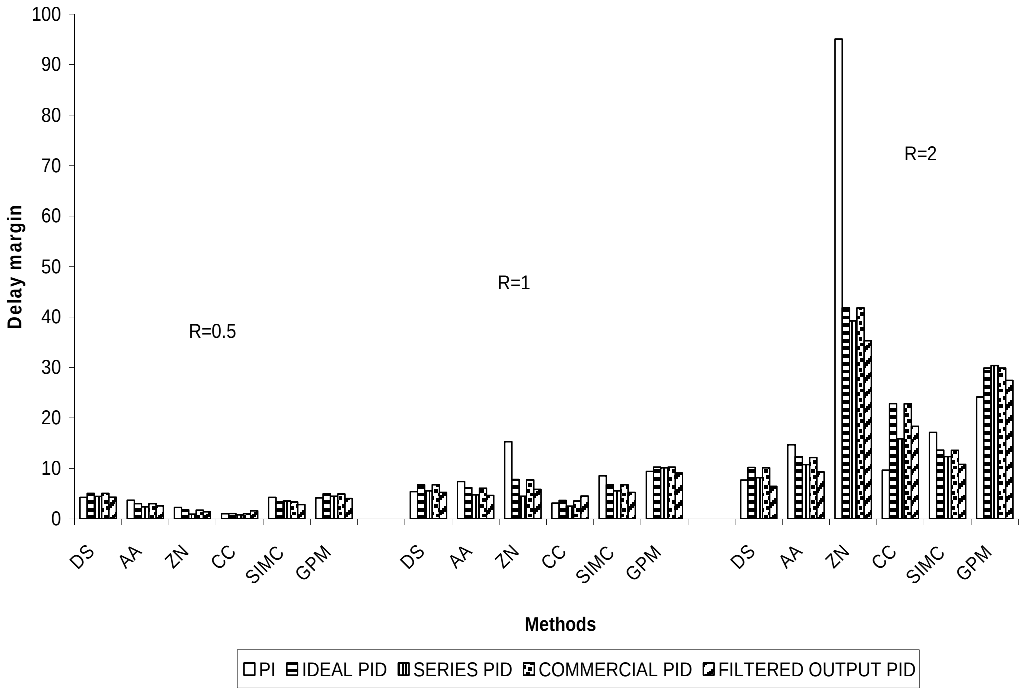

Figure 6 - Delay Margins

.

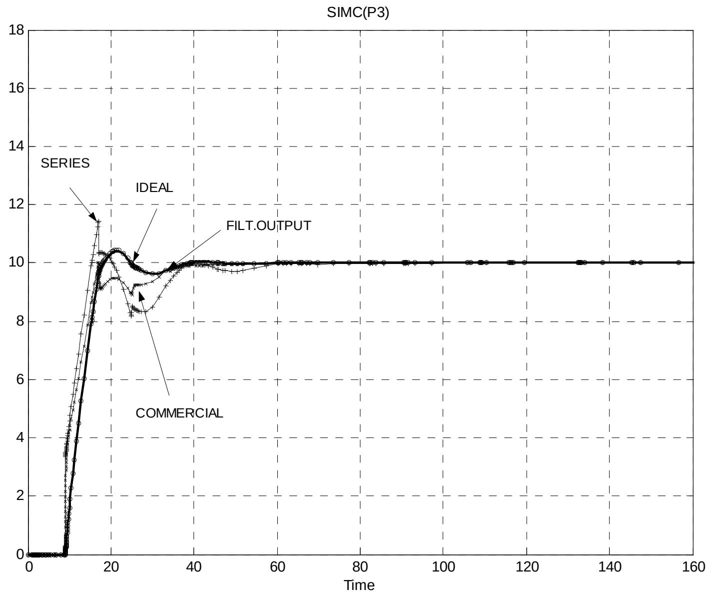

Figure 7 - SIMC tuned Controllers applied to P3

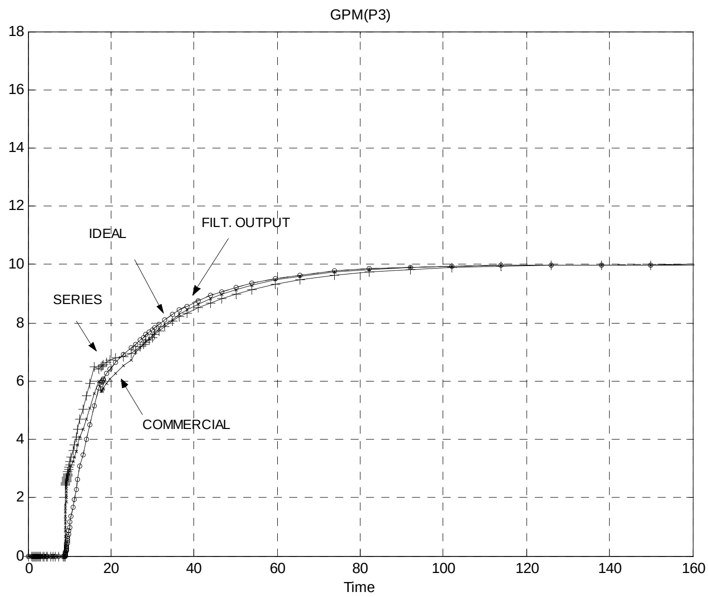

Figure 8 - GPM tuned Controllers applied to P3

Explicit robustness tests were also conducted, where controllers designed using a nominal model were applied to processes with parameters $\pm 25\%$ of the nominal case and performances are indicated by IAE values. A change in process gain caused a general deterioration in control, as would be expected, indicated by higher overshoots and longer settling times, leading to IAE values which are higher than the nominal cases. Systems tuned using GPM method is the least affected and most suitable for all controller algorithms. Controllers tuned by the ZN and CC methods are badly affected by process-model mismatch in the process gain, especially when the nominal gain is lower. However, the degradation due to using a higher nominal gain for tuning is not as pronounced. Generally, all resulted in smaller or no overshoot at all, slower response with longer settling times, causing an increase in IAE. Figure 9 shows the percentage change in IAE when the process gain is smaller than the nominal case. A positive value means an increase in IAE (degradation in performance) while negative value means that there is reduction in IAE (improved performance). IAE changes the most when R=2, (larger delay), with the PI controller being the least affected. The increase in IAE is as low as 2% for PI and up to 35% for the Commercial PID.

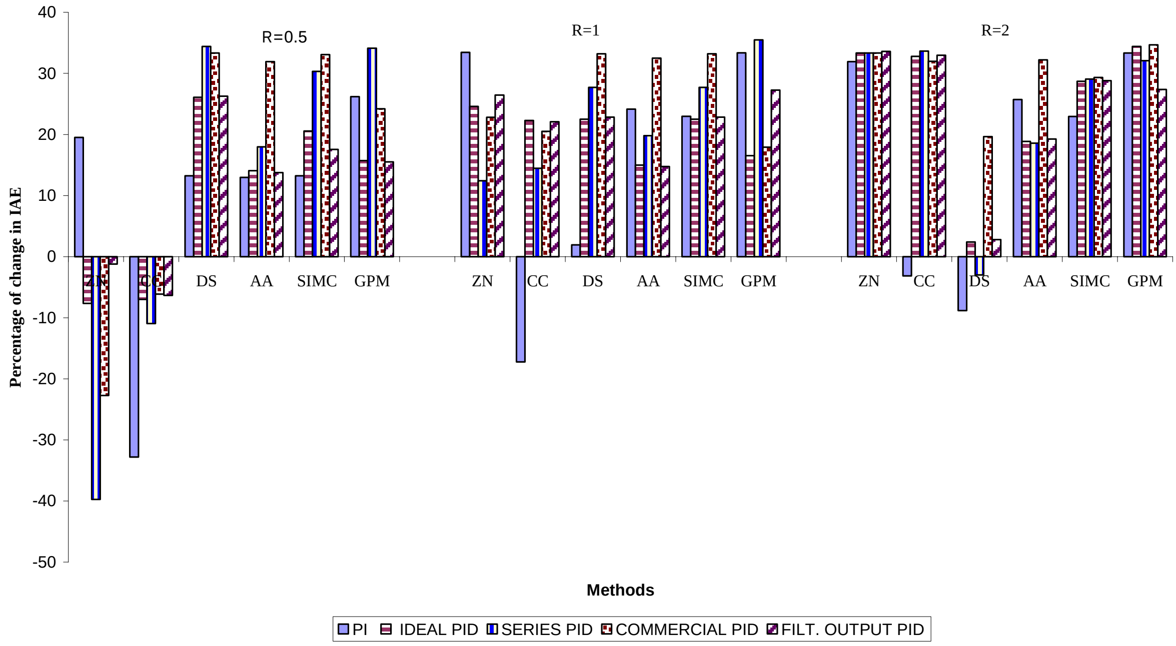

Figure 9 - Percentage of change in IAE with mismatch in process gain (-25%)

Concluding Remarks

All the PID controller structures considered collapse to the same PI form when derivative action is not required. Apart from the series PID, the other PID algorithms share very similar structures, therefore giving similar performances. The series PID was found to be the least amenable to tuning with methods that are developed for the ideal PID, with the highest propensity to give erratic initial responses when tracking a setpoint change. The more recently proposed techniques provided better performances and as expected, processes with larger time-delay to time-constant ratios were more difficult to control. The DS, AA, SIMC and GPM methods facilitate performance shaping via a single parameter. The DS and SIMC methods accommodate this through closed-loop time constant specifications, while fine-tuning with the AA method involves setting a value for the desired overshoot. Of these, the GPM tuning method gave the most consistent results across the different processes and controller structures. However, the tuning procedure is by far the most involved, requiring the solution of a set of equations. The SIMC method gave the next best set of results. Although stability margins are less than those obtained using GPM, the advantage of the SIMC tuning method lies in its simplicity.

REFERENCES

[1] Astrom, K., Hagglund, T. 2001. The Future of PID Control. Control Engineering Practice, **9**: 1163-1175

[2] Astrom, K., Hang, C., Persson, P., Ho, W.K. 1992. Towards Intelligent PID Control. Automatica **28**: 1-9

[3] Seborg, D., Edgar, T., Mellichamp, D. 1989. *Process Dynamics and Control*. New York: John Wiley.

[4] Coughanowr, D. 1981. *Process Systems Analysis and Control.* New York: McGraw Hill.

[5] Astrom, K., Hagglund, T. 1995. *PID Controllers : Theory, Design and Tuning, 2nd Edition*. New York: Research Triangle Park.

[6] Goodwin, G., Graebe, S.F., Salgado, M.E. 2001. *Control System Design*. New Jersey: Prentice Hall.

[7] Astrom, K. 1996. PID Control. *The Control Handbook*. Florida: CRC Press.

[8] St. Clair, D.W. 2000. The PID Algorithm, URL: http://members.aol.com/pidcontrol/pid_algorithm.htm

[9] Skogestad, S. 2002. Simple Analytic Rules For Model Reduction and PID Controller Tuning, Journal of Process Control **13**: 291-309.

[10] Luyben, W. 1990. *Process Modelling, Simulation and Control for Chemical Engineers, 2nd Edition*. New York: McGraw Hill

[11] Shinskey, F.G. 1988. *Process Control Systems, Application, Design and Tuning.* New York: McGraw Hill

[12] Chau, P. 2002. *Process Control.* UK: Cambridge University Press.

[13] Harriott, P. 1964. *Process Control.* New York: McGraw Hill

[14] Marlin, T. 1995. *Process Control: Designing Processes and Control Systems for Dynamic Performance.* New York: McGraw Hill.

[15] Abbas, A. 1997. A New Set of Controller Tuning Relations. ISA Transactions **36**: 183-187

[16] Ho, W.K., Lim, K.W., Hang, C.C., Ni, L.Y. 1999. Getting More Phase Margin and Performance Out of PID Controllers. Automatica **35**: 1579-1585.

[17] Ogata, K., 1997. *Modern Control Engineering, 3rd Edition*. New Jersey: Prentice Hall.

[18] Franklin, G.F., Powell, J.D., Abbas, E. N. 1986. *Feedback Control of Dynamic Systems. 4th Edition.* Reading, Mass.: Addison–Wesley.

[19] Ho, W.K., Lim, K.W., Xu. W. 1998. Optimal Gain and Phase Margin Tuning For PID Controllers. Automatica **8**: 1009-1014

[20] Caldwell, W. 1959. *Frequency Response for Process Control*. New York: McGraw Hill.

NOMENCLATURE

K_c The proportional gain
T_I integral time and derivative time
T_D the tuning constants
$U(s)$ is the output of the controller
$E(s) = X(s) - Y(s)$ the error between setpoint, $X(s)$
$Y(s)$ controlled output
K process gain
τ_f time controller
P_m phase margin
A_m gain margin
R time delay to time constant ratio
IAE Integral Absolute Error