Proc. ISA 2002 Annual Meeting, **424**, P057, Chicago, IL (2002)
R. Rice and D. J. Cooper

Design and Tuning of PID Controllers for Integrating (Non-Self Regulating) Processes

Bob Rice
Chemical Engineering Dept
University of Connecticut, U-3222
Storrs, CT 06269-3222

Doug Cooper
Chemical Engineering Dept
University of Connecticut, U-3222
Storrs, CT 06269-3222

KEYWORDS

PID, IMC, derivative filter, integrating process

ABSTRACT

Presented is an easy to use and broadly applicable method for tuning PID controllers for integrating processes. Details are presented on the requirements for collecting closed loop dynamic process test data near the design level of operation, the fitting of an integrating dynamic model form to this test data and correlations for computing controller tuning values based on the parameters from the resulting model fit.

The method presented is applicable to PID control algorithms in both the interacting and non-interacting derivative forms. The work builds on the work of Chien and Fruehauf [8] and their use of the internal model control (IMC) structure to derive tuning correlations for integrating processes. One novel contribution of this work is the extension of the tuning correlations to include the PID with derivative filter forms.

The design and tuning method is demonstrated on process simulations for both set point tracking and disturbance rejection. Results show that the methods described here compare favorably with other more computationally intensive approaches.

INTRODUCTION

Self-regulating processes seek a natural steady state operating level in open loop if the manipulated and disturbance variables are held constant for a sufficient period of time. It is not uncommon for some temperature, level, pressure and other measured process variables to move in an unbounded manner when perturbed in open loop by a manipulated or disturbance variable. Such behavior is characteristic of integrating (non-self regulating) processes.

Integrating processes are surprisingly challenging to control. They can move to extreme and even dangerous levels if left unregulated, so it is often necessary to tune controllers on such processes in closed loop. An additional challenge is that the tuning correlations proven for self regulating processes can yield poor performance when applied to integrating processes.

The design and tuning of controllers for both self regulating and integrating processes follows a simple three step procedure: collect closed loop dynamic process data, fit the data with a simple linear model, and use the model parameters in correlations to obtain PID tuning parameter values. This procedure is detailed and demonstrated in the remainder of this paper.

FORMS OF THE PID CONTROL ALGORITHM

Vendors offer different forms of the PID algorithm. The two most popular forms are explored here. Both forms are identical in capability but require slightly different correlations for tuning. One form is referred to under a variety of names including the ideal, non-interacting and ISA algorithm [1-3]. In this work it will be referred to as the *ideal* form. The ideal PID transfer function is expressed:

$$G_C(s) = K_C\left(1 + \frac{1}{\tau_I s} + \tau_D s\right) \tag{1}$$

The other PID form is interchangeably referred to as the interacting, series and industrial form [4]. In this work it will be referred to as the *interacting* form. The interacting PID algorithm is expressed:

$$G_C(s) = K_C\left(1 + \frac{1}{\tau_I s}\right)\left(\tau_D s + 1\right) \tag{2}$$

The PID controller has well-known difficulties when there is noise in the measured process variable $y(t)$ [5]. Specifically, derivative action causes measurement noise to be amplified and reflected in the controller output signal $u(t)$. This is because a noisy signal produces conflicting derivatives as the measurement slope appears to dramatically alternate direction at every sample. The measurement derivative can alternate between a large increasing slope followed by a large decreasing slope, sample after sample. The result is a series of alternating and compensating controller actions that can cause performance to degrade and in some cases lead to controller instabilities.

To improve performance when there is noise or random error in the measured process variable, the PID algorithm is modified with a derivative filter. This filter limits large control moves by subtracting some fraction of the derivative of the control action from the calculated controller output. A large control move has a large derivative, and it is this undesirable action that is dampened by the filter. The extent of filtering is regulated by the filter constant, α.

The ideal PID with derivative filter [6] has the transfer function:

$$G_C(s) = K_C\left(1 + \frac{1}{\tau_I s} + \tau_D s\right)\frac{1}{\alpha \tau_D s + 1} \tag{3}$$

The interacting PID with derivative filter form [2,3,7,8] is expressed:

$$G_C(s) = K_C\left(1+\frac{1}{\tau_I s}\right)\left(\frac{\tau_D s+1}{\alpha\tau_D s+1}\right) \tag{4}$$

CONTROLLER DESIGN AND IMC TUNING

Designing any controller from the family of PID algorithms entails:

- collecting dynamic process data as near as practical to the design level of operation,
- fitting a simple linear model to the process data, and
- using the resulting model parameters in correlations to obtain initial controller tuning values.

For integrating processes, dynamic test data is often collected in closed loop to maintain stability. In the examples presented here, the process is under P-Only control and controller tuning has been tweaked enough to maintain a reasonably steady operation. The set point is then stepped far enough such that the resulting controller output change forces a clear response in the measured process variable. The process data is recorded during this transient event.

The next step is to fit this dynamic test data with a simple linear model. For non-self regulating processes, the appropriate model is the first order plus dead time integrator (FOPDT Integrator) form, expressed:

$$G_M = \frac{K_p^* e^{-\theta_p s}}{s} \tag{5}$$

Where K_p^* is the integral process gain and has units of $[y(t)/(u(t)\cdot t)]$
θ_p is dead time and has units of time, t.

The parameters from this model fit are then used in tuning correlations to obtain controller tuning values. The correlations used here are derived from the IMC block diagram [6,8,9], shown in Fig. 1:

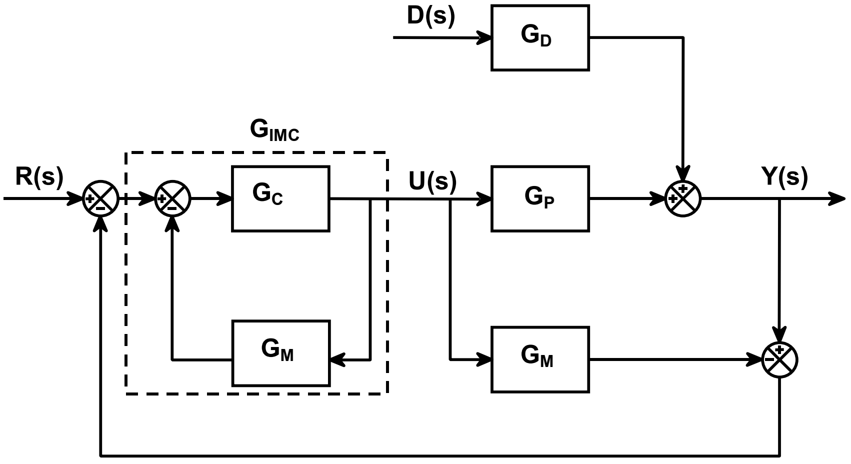


FIG. 1 - IMC BLOCK DIAGRAM

The controller transfer function obtained from this block diagram is expressed:

$$G_C = \frac{G_{IMC}}{1 - G_{IMC} G_M} \tag{6}$$

Using the procedure outlined in Chien and Fruehuaf [8], tuning rules are derived for the entire family of PID controllers is listed in Table 2. The PID w/ filter correlations are a novel contribution of this work.

	K_C	τ_I	τ_D	α
PI	$\frac{1}{K_p^*}\frac{2\tau_C + \theta_P}{(\theta_P + \tau_C)^2}$	$2\tau_C + \theta_P$		
PID Ideal	$\frac{1}{K_p^*}\left(\frac{2\tau_C + \theta_P}{(\tau_C + 0.5\theta_P)^2}\right)$	$2\tau_C + \theta_P$	$\frac{0.25\theta_P{}^2 + \theta_P\tau_C}{2\tau_C + \theta_P}$	
PID Interacting	$\frac{1}{K_p^*}\left(\frac{2\tau_C + 0.5\theta_P}{(\tau_C + 0.5\theta_P)^2}\right)$	$2\tau_C + 0.5\theta_P$	$0.5\theta_p$	
PID Ideal w/ filter	$\frac{1}{K_p^*}\frac{(2\tau_C + 1.5\theta_p)}{(\tau_C^2 + 2\tau_C\theta_p + 0.5\theta_p{}^2)}$	$2\tau_C + 1.5\theta_p$	$\frac{0.5\theta_p{}^2 + \tau_C\theta_p}{(2\tau_C + 1.5\theta_p)}$	$\frac{(0.5\tau_C{}^2)(2\tau_C + 1.5\theta_p)}{(\tau_C^2 + 2\tau_C\theta_p + 0.5\theta_p{}^2)(0.5\theta_p + \tau_C)}$
PID Interacting w/ filter	$\frac{1}{K_p^*}\frac{(2\tau_C + \theta_p)}{(\tau_C^2 + 2\theta_p\tau_C + 0.5\theta_p{}^2)}$	$(2\tau_C + \theta_p)$	$0.5\theta_p$	$\frac{\tau_C^2}{\tau_C^2 + 2\theta_p\tau_C + 0.5\theta_p^2}$

TABLE 2 – IMC TUNING CORRELATIONS FOR FOPDT INTEGRATOR PROCESSES

The IMC method includes a closed loop time constant, τ_C, that is used to adjust controller performance. A larger τ_C provides for a slower, conservative response with a long settling time, while a smaller τ_C provides for a more rapid and aggressive response with a shorter settling time. Tyreus [10] suggests that τ_C can be computed from the process dead time as:

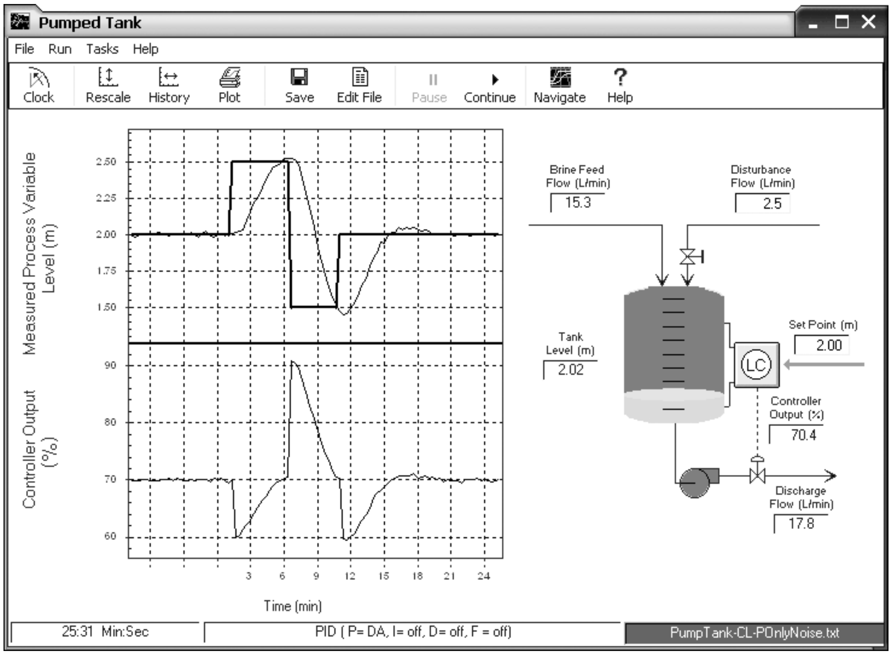
$$\tau_C = \theta_p \sqrt{10} \tag{7}$$

IMPLEMENTATION EXAMPLES

Controller performance can be evaluated in terms of set point tracking, disturbance rejection or both. The IMC tuning correlations offer reasonable tuning for any of these criteria. Good performance can be defined many ways. This work presents examples where a compromise is sought between settling time and overshoot while minimizing the “chatter” in the controller output. All examples in this work use the Control Station® software package (www.controlstation.com) for simulation, analysis and plotting.

EXAMPLE 1

Figure 4 shows the pumped tank case study from Control Station. The measured process variable is liquid level. To maintain level, the controller manipulates liquid flow rate out of the bottom of the tank by adjusting a throttling valve at the discharge of a constant pressure pump. The disturbance variable is the flow rate of a secondary feed to the tank. The height of liquid in the tank does not impact the discharge flow rate. As a result, when the total flow rate into the tank is greater than the discharge flow rate, tank level will continue to rise until the tank is full, and when the total flow rate into the tank is less than the discharge flow rate, the tank level will fall until empty. Therefore, both the disturbance and manipulated variable have integrating effects on the process variable.



Brine Feed Flow (L/min) 15.3

Disturbance Flow (L/min) 2.5

Tank Level (m) 2.02

Set Point (m) 2.00

Controller Output (%) 70.4

Discharge Flow (L/min) 17.8

25:31 Min:Sec

PID (P= DA, I= off, D= off, F = off)

PumpTank-CL-POnlyNoise.txt

FIG. 4 – THE PUMPED TANK INTEGRATING PROCESS

Using the controller design procedure discussed in previously, tank level is stabilized with a P-Only controller. Dynamic process data is generated by stepping the set point around the design level of operation enough to see a clear response in the measured process variable. A dynamic FOPDT Integrator model is fit to the closed-loop response data.

Inspection of the model fit in Fig 5 shows that the model fit reasonably describes the dynamic behavior of the process with an integral gain of -0.022 m/(%·min) and dead time of 1.1 min, or:

$$G_p(s) = \frac{-0.022}{s} e^{-1.1s} \tag{8}$$

Using Eq. 7, the predicted closed loop time constant is:

$$\tau_C = \theta_p \sqrt{10} = 3.5\,\text{min} \tag{9}$$

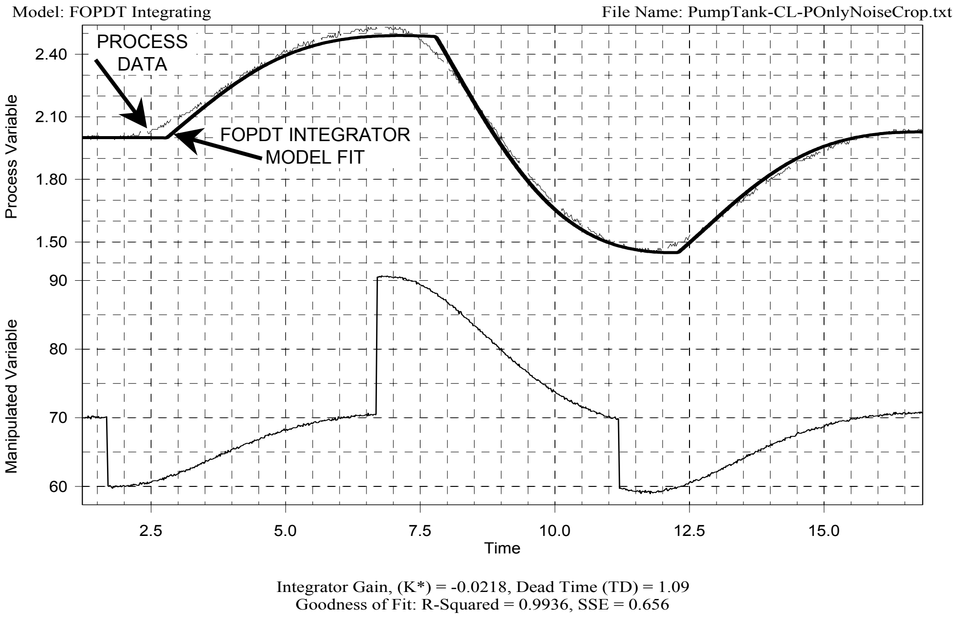


FIG. 5 – FOPDT INTEGRATOR MODEL FIT FOR THE PUMPED TANK PROCESS

Substituting this closed loop time constant and the above FOPDT Integrator model parameters into the correlations for the PID Ideal controller in table 2 yields the tuning values:

PID Ideal	K_C = -22.9 %/m	τ_I = 8.0 min	τ_D = 0.51 min	
PID Ideal w. Filter	K_C = -19.5 %/m	τ_I = 8.5 min	τ_D = 0.51 min	$\alpha = 0.64$
PID Inter. w. Filter	K_C = -18.28 %/m	τ_I = 8.0 min	τ_D = 0.55 min	$\alpha = 0.59$

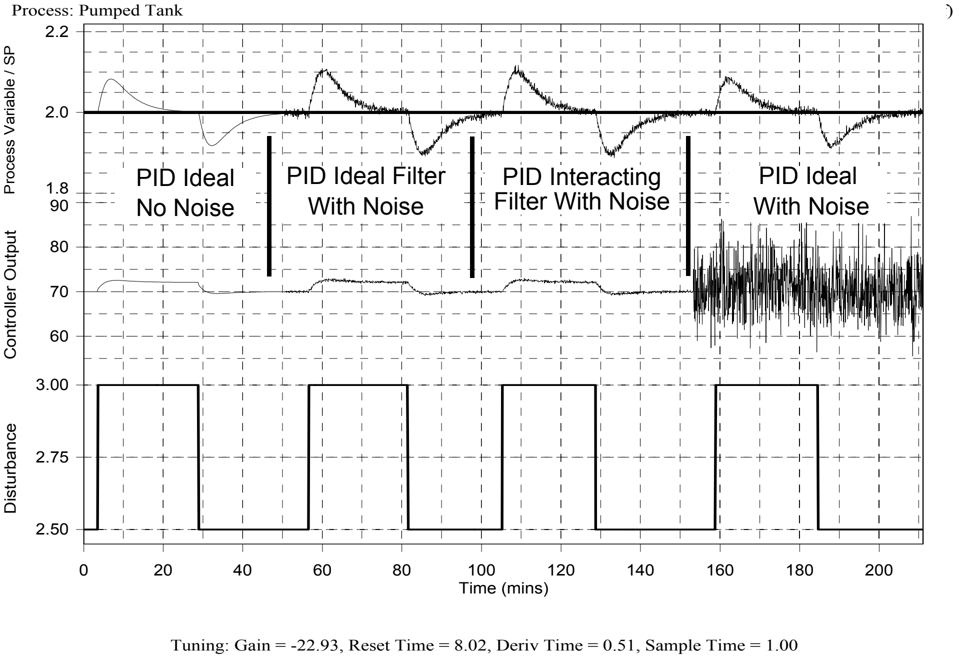
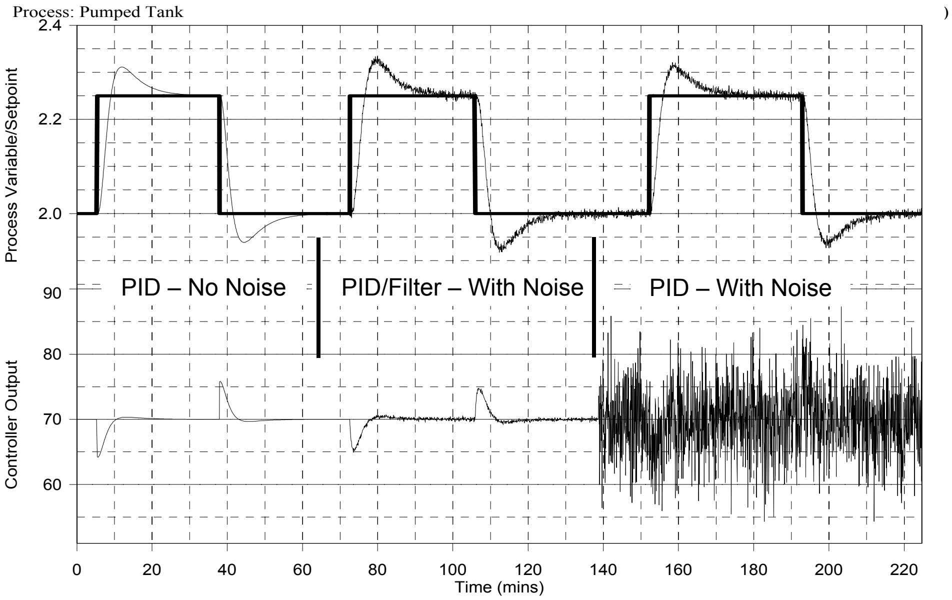


FIG. 6 – EFFECT OF NOISE ON THE DISTURBANCE REJECTION PERFORMANCE

Figure 6 shows the disturbance rejection performance for the Pumped Tank under PID control. The first disturbance rejection segment shows the performance of the ideal PID algorithm when process noise is set to zero. The second and third segment shows that the ideal and interacting PID w/ filter algorithms behave the same in the presence of noise. Note that controller tuning is different for the ideal and interacting forms as detailed in Table 2. Also note that the filtered algorithm in the presence of noise (segments 2 and 3) performs similar to the unfiltered algorithm with no noise (segment 1). The last segment shows that with no filter, the PID algorithm causes dramatic chatter in the controller output.



Tuning: Gain = -22.93, Reset Time = 8.00, Deriv Time = 0.51, Sample Time = 1.00

FIG. 7 – EFFECT OF NOISE ON SET POINT TRACKING PERFORMANCE

Figure 7 shows set point tracking performance for the Pumped Tank. The first set point tracking segment shows the performance of the ideal PID algorithm with the process noise set to zero. The second segment shows that the ideal PID w/ filter in the presence of noise performs similar to the unfiltered algorithm with no noise (segment 1). The last segment again shows that with no filter, the PID algorithm causes dramatic chatter in the controller output.

EXAMPLE 2

A widely published example of an integrating process used for PID tuning comparisons [10-15] is:

$$G_p(s) = \frac{0.0506}{s} e^{-6s} \qquad G_D(s) = \frac{1}{(s+10)(s+20)} \tag{10}$$

The process transfer function $G_P(s)$ of Eq. 10 is implemented in Control Station with normally distributed random error to approximate noise in the measured process variable. The disturbance transfer function $G_D(s)$ of Eq. 10 is the disturbance model used for the disturbance rejection scenarios.

Rather than simply plugging the known process model parameters into the tuning correlations, the design procedure described earlier is followed here. Hence, the transfer function process is stabilized with a P-Only controller and dynamic process data is generated by stepping the set point at the design level of operation. A FOPDT Integrator model is fit to the data as shown in Fig. 8.

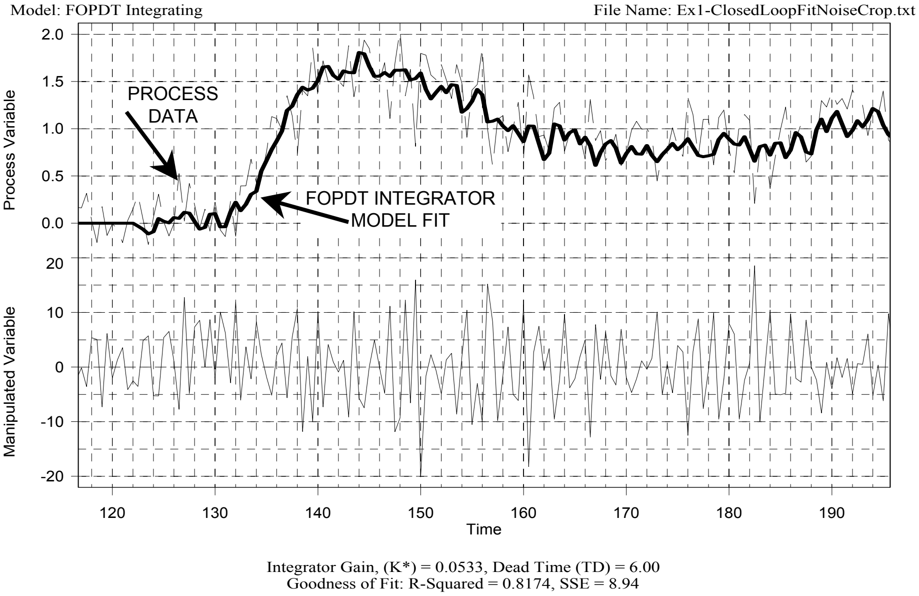


FIG. 8 - FOPDT INTEGRATOR FIT OF CLOSED LOOP DATA

Visual inspection of the FOPDT Integrator model fit in Fig. 8 shows a reasonable description of the dynamic process behavior of $K_p^* = 0.053$ and $\theta_p = 6.0$. Using these model parameters in the IMC tuning rules of Table 2, PID w/ Filter tuning parameters are computed and listed in Table 9, along with tuning values published using different methods proposed by other researchers.

	IMC	Luyben	Wang/Cluett	Z-N
K_c	1.5	2.4	2.0	3.0
τ_I	47	52.5	31.2	12
τ_D	2.8	3.8	1.6	3
α	0.64	0.1	.16	0.1

TABLE 9 - PID WITH FILTER TUNINGS COMPUTED BY DIFFERENT METHODS

Luyben [12] in Table 9 uses the classical frequency-response method where the favored maximum closed-loop log modulus is specified. Wang and Cluett [13] use the desired control signal trajectory as a performance specification to solve for the PID controller parameters in the frequency domain. The Ziegler Nichols [12] brings the process to the brink of stability to determine the ultimate gain and ultimate period, and calculate their parameters based off of these values.

Figure 10 shows the set point tracking capabilities of the PID w/ filter controller using the four sets of tuning parameters in Table 9. As shown, the IMC method provides equal performance in set point tracking and superior filtering of excessive controller action when compared to the other methods. It is important to recognize that simply increasing the filter constant for the other methods will not provide improved filtering while maintaining identical performance. All the tuning parameters work together and if the filter constant is increased, the other values must be adjusted to prevent the controller from becoming too sluggish.

The larger gain present in the Ziegler Nichols tunings produces a larger overshoot in the set point tracking control objective, but is faster to recover from a disturbance introduced to the system. As evident by this result, there is a definite tradeoff between the set point tracking and disturbance rejection scenarios. The larger gain helps the disturbance rejection, but increases the amount of overshot in the set point tracking.

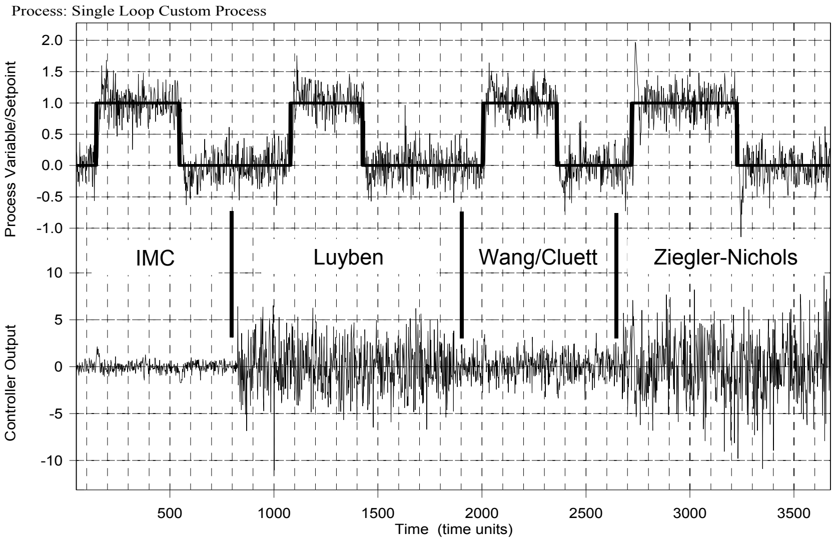


FIG. 10 - SET POINT TRACKING PERFORMANCE FOR THE EQ. 10 PROCESS

Figure 11 shows the disturbance rejection performance for the Eq. 10 process using the tuning values listed in Table 9. Again, the IMC tuning parameters provide superior filtering capability.

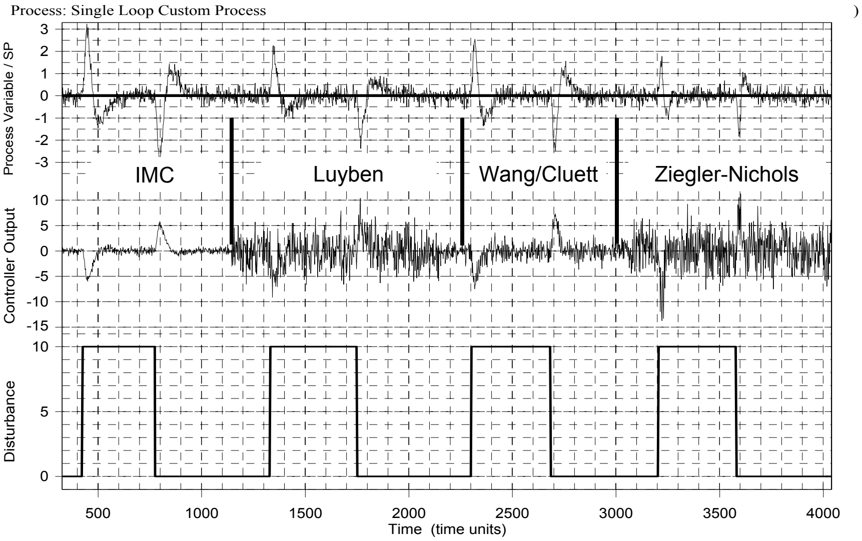


FIG. 11 - DISTURBANCE REJECTION PERFORMANCE FOR THE EQ. 10 PROCESS

Figure 11 shows that the IMC disturbance rejection performance has a larger maximum deviation compared with the other methods, though the settling time is comparable with that of Luyben and Wang and Cluett. Interestingly, Ziegler Nichols provides the best performance while exhibiting the largest chatter in the controller signal. Beyond that, the concept of bringing process to the brink of stability as required by the Ziegler Nichols method is not the preferred method by operations personnel.

CONCLUSION

An easy to use and broadly applicable method for tuning PID controllers for integrating processes has been detailed and demonstrated. The method, based on IMC, entails collecting closed loop dynamic process test data near the design level of operation, fitting an integrating dynamic model to this test data and using the resulting model parameters in correlations to compute controller tuning values.

Tuning correlations for PID with interacting and non-interacting derivative forms and PID with derivative filter are presented. Demonstrations of set point tracking and disturbance rejection show that the method compares favorably with other more computationally intensive approaches.

NOMENCLATURE

K_p	=	process gain [y/u]
K_p^*	=	integral gain [y/(u·time)]
θ_p	=	dead time [time]
K_c	=	controller gain [u /(y·time)]
τ_D	=	derivative time [time]
α	=	derivative filter constant
τ_C	=	closed loop time constant [time]

REFERENCES

1. Stephanopoulos, G., "*Chemical Process Control*," 4th Ed., Prentice Hall, New Jersey, 1984.

2. Seborg, D., Edgar, E.F., and Mellichamp, D.A., "*Process Dynamics and Control*," 9th Ed., John Wiley & Sons Inc., 1989.

3. Ogunnaike B.A., and Ray W.H, "*Process Dynamics, Modeling, and Control*," 2nd Ed., Oxford University Press, 1994.

4. Harris, T.J., Tyreus, B.D.. "Internal Model Control. 4. PID Controller Design." Comments. *Ind. Eng. Chem. Res*. 26, 1987, pp. 2161-2162.

5. Luyben, W.L., "Effect of Derivative Algorithm and Tuning Selection on the PID Control of Dead-time Processes," *Ind. Eng. Chem. Res.,* 40, 2001, pp. 3605-3611.

6. Rivera, D. E., Morari, M., and Skogestad, S., "Internal model control: PID controller design" *Ind. Eng. Chem. Process Des. Dev.,* 25, 1986, pp. 252-265.

7. Smith, C., and Corripio A.B., "*Principles and Practice of Automatic Process Control,*" 2nd Ed., Wiley, New York, 1997.

8. Chien, I.-L., and Fruehauf, P.S., "Consider IMC Tuning to Improve Controller Performance', *Chem. Eng. Prog.,* 86, October, 1990, pp. 33-41.

9. Fruehauf, P.S., Chien, I.-L., Lauritsen, M.D., "Simplified IMC-PID Tuning Rules," *ISA Transactions*, 33, 1994, pp. 43-59.

10. Tyreus, B.D., and Luyben, W.L., "Tuning PI Controllers for Integrator/Dead Time Processes," *Ind. Eng. Chem. Res.,* 31, 1992, pp. 2625-2628.

11. Visioli, A., "Optimal Tuning of PID Controllers for Integral and Unstable Processes," *IEE Proc.-Control Theory Appl*., 148, No. 2, March, 2001, pp. 180-184.

12. Luyben, W.L., "Tuning Proportional-Integral-Derivative Controllers for Integrator/Deadtime Processes," *Ind. Eng. Chem. Res.,* 35, 1996, pp. 3480-3483.

13. Wang, L., and Cluett, W.R., "Tuning PID Controllers for Integrating Processes," *IEE Proc. Control Theory Appl*., 144, No. 5, September, 1997, pp. 385-392.

14. Kaya, I., and Atherton, D.P., "A PI-PD Controller Design for Integrating Processes," *Proc. of the American Control Conf.*, June, 1999, pp. 258-262.

15. Poulin, É., Pomerleau A., "PID Tuning for Integrating and Unstable Processes," *IEE Proc.-Control Theory Appl*., 143, No. 5, September, 1996, pp. 429-435.