Setpoint PI Controllers for Systems with Large Normalized Dead Time

Babar Z. Khan and Brad Lehman

Abstract—This paper derives proportional-integral (PI) control algorithms for first-order plants with time lag. First, the effects of proportional control on overshoot and rise time are analyzed. Then these results are applied to aid in the design of PI controllers that utilize a pole-zero cancellation. New PI tuning methods are proposed for design criteria typical of those found in metal rolling mill applications with large normalized dead time.

I. INTRODUCTION

RECENT work by [2], [8], [9], and [14] has begun to improve tuning algorithms for proportional-integral-derivative (PID) industrial controllers. Many industrial PID control algorithms were designed over 30 years ago [6], [7], [15], [16], prior to the advances in robust control and prior to the PC-computing/software revolution. Therefore, commonly used PID tuning methods were derived based on "observations" without extensive numerical simulations to verify issues such as disturbance rejection, sensitivity, robustness, setpoint settling time, etc.

This paper continues the work of [2], [8], [9], and [14] by deriving simple PI controllers for applications in industrial/process control. Results were obtained by performing extensive numerical simulations. Performance is evaluated using both time domain and frequency domain criteria. The choice of studying PI controllers and not PID controllers in this work evolves from the fact that approximately 90% of all industrial PID controllers have the derivative action turned off [3].

The research presented is motivated by rolling mill control applications in the (aluminum) metal industry. In rolling mills, it is often desired to track step reference inputs so that the steady-state thicknesses are within 1–2% of a desired value [4]. Disturbances tend to be low frequency and are often due to initial conditions. By keeping settling time small, less raw material is wasted. For instance, in the flattening of aluminum for the beverage can industry, up to 2% of the aluminum sheets are routinely discarded in typical cold rolling mill processes due to transient start-up times [4]. Another important design criteria is to have small overshoot. High overshoot tends to add wear on the equipment by causing heavy rollers to "oscillate" [1], [4], [11].

The above design criteria are typical of setpoint industrial problems and are not unique to the metal industry. A main

B. Z. Khan is with the Department of Electrical and Computer Engineering, Mississippi State University, Mississippi State, MS 39762 USA.

B. Lehman is with the Department of Electrical and Computer Engineering, Northeastern University, Boston, MA 02115 USA.
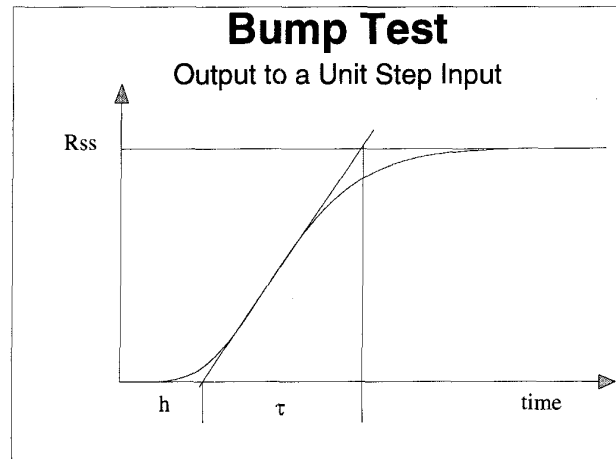
Fig. 1. Unit step open-loop response of a plant.

difficulty in tuning PI controllers for the metal industry, however, is that the so-called "normalized dead time" in the three term, first-order plus time delay (FOPTD), model is larger than many other industrial processes (up to 10!) [3].

The three-term model is often used to estimate models of higher order industrial processes and is given by

$$\frac{R_{ss}e^{-hs}}{\tau s + 1}. \tag{1}$$

The constants R_{ss}, τ, and h are normally experimentally determined from the so-called "bump" test. That is, the input is set to a unit step and the open-loop response is plotted, as in Fig. 1. The steady-state gain, R_{ss}, is equal to the final value of the output. A tangent line is drawn to the point of maximum slope and intersects the time axes at $t = h$ and reaches a height R_{ss} at time $t = h + \tau$. (The work of [2] suggests that τ may be selected in a less conservative manner.) It should be noted that rolling mill models, motivating this paper, are sometimes modeled in a form similar to (1). (More advanced techniques for modeling/control of rolling mills are also well documented [11].) The normalized dead time is given as h/τ, and processes with large h/τ, perhaps greater than 0.5, are considered difficult to control [2].

Contributions of Paper: The main results of the paper are as follows:

1) The effect of proportional control for FOPTD plants is studied. Presently, there are no techniques in the literature for accurately predicting overshoot and rise time for such systems. This research provides the first simple estimates for these important transient response characteristics. The results are presented in Section II.

2) In Section III-B, the results of Section II are used to improve techniques for selecting PI controller gains that utilize pole-zero cancellations.

3) In Section III-C, a new algorithm for selecting PI controller gains for FOPTD plants is presented. This algorithm has excellent settling time for step reference inputs and step load disturbances, especially for systems with large normalized dead time.

II. PROPORTIONAL CONTROL

A. Introduction

In this section, the effects of proportional control on the percent overshoot and the rise time for FOPTD plants are discussed. The inspiration of analyzing proportional controllers is derived from the fact that several prevalent PI control algorithms [5], [7], [9], [12] use the zero of the PI controller to cancel the pole of the plant. This reduces the system to a proportional controller acting on an astatic plant (a plant whose pole is at the origin i.e., $p = 0$) with time delay.

In the literature, there exist well-known techniques for predicting the percent overshoot and the rise time of first- and second-order systems when there is no delay, $h = 0$. For example, in second-order systems, the overshoot and rise time are calculated using the damping factor and the location of the dominant system poles. There are no techniques for predicting overshoot and rise time, however, when $h > 0$. It should be mentioned that unlike the case of no delay ($h = 0$), first-order plants with delay and proportional control often exhibit overshoot.

Consider a first-order plant with time delay, h. Suppose it is desired to control the system using unity feedback and proportional control as in Fig. 2. Using the final value theorem, the steady-state value of the system for a unit step reference input is given by $y_{ss} = K/(K + p)$. Clearly, the steady-state value of the system does not reach unity. This shows that the output of the system in Fig. 2 does not track a step reference input, except for astatic plants. A typical input step response to this delay differential equation is given in Fig. 3. The solution is obtained using a package called Simnon [13]. The parameters used are $p = 1$, $h = 1$, and $K = 1$. M_P is the maximum value and t_r is the rise time. Percent overshoot is defined as $100\ (M_P - y_{ss})/y_{ss}$, and rise time is defined as the time it takes for the output to rise from 10–90% of y_{ss}, where y_{ss} is the final value of the output. The results of this section were obtained by performing extensive numerical simulations and curve fitting techniques on the FOPTD model to obtain estimates on M_P and t_r.

B. Procedure

The results of this section consider systems as in Fig. 2. To determine a relationship between overshoot and rise time to system parameters (p, K, and h), extensive numerical simulations were performed. First, time was rescaled so that $h = 1$. Then, p was fixed and the output was simulated for different values of K. Next, p was incremented and the procedure was repeated over and over again. As a final step,
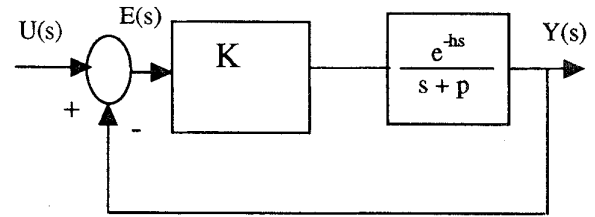

Fig. 2. First-order plant with time delay and proportional control.

time was rescaled so that percent overshoot and rise time could be read from saved files. All simulations were run for step inputs. For each value of ph, overshoot versus Kh, and rise time versus Kh curves are plotted as shown in Figs. 4 and 5, respectively. From Fig. 4, it can be seen that the relationship between overshoot and Kh, for large values of Kh, is almost linear, and for small values of Kh is close to exponential. Also, the slope and the break point of the curve depends on the location of ph. When ph is small, the slope of the curves increases and the break point occurs for larger Kh. On the other hand, for rise time, the relationship is estimated by monotonically decreasing curves. As expected, large ph or large Kh has fast rise time. Fig. 5 plots rise time for $ph = 0.5$, 1.0, 1.5, 2.5, 5.0, 7.0, and 10.0.

The breakpoints of the overshoot curves can be estimated theoretically. Breakpoints are the points for which overshoot begins to appear and usually corresponds to when the dominant pole is about to leave the real axis. For K larger than the breakpoint gain, $K > K_b$, the dominant pole enters the complex plane (causing overshoot). The transfer function of the closed-loop system is given as $T(s) = (Ke^{-hs})/(s + p + Ke^{-hs})$. At the breakpoint, or the saddle point, it is known that $dT(s)/ds = 0$ [5]. Therefore, at the breakpoint $dT(s)/ds = Ke^{-hs}(1 - hKe^{-hs}) - (s + p + Ke^{-hs})(-hKe^{-hs}) = 0$, which implies that $1 + ph + sh = 0$. It is easy to obtain that $K_b = (he^{1+ph})^{-1}$. This K_b is the break point for a particular h and p. For $K > K_b$ overshoot should begin to appear.

C. Results

Data points for overshoot and rise time have been curve fitted using table curve [13]. To achieve accuracy, each curve for overshoot is divided into four parts and curve fitted separately. It is not necessary to divide the rise time curves into different sections.

1) Percent Overshoot: An estimate on the percent overshoot for the system in Fig. 2 is given by

$$\text{Percent overshoot} = \text{Max}[0, M(p, K, h)] \qquad (2)$$

where $M(p, K, h)$ is chosen from one of four cases below. (If $M(p, K, h)$ is complex then percent overshoot = 0.) Note that the below formulas are valid only when the system is bounded input/bounded output (BIBO) stable. For example, for stability independent of delay, it is necessary and sufficient that $0 < |K| < p$. For $0 < p < K$, the upper bound on h is obtained by simultaneously solving: $\cos(wh) = -p/K$ and $\sin(wh) = w/K$, where w takes on any positive value [10].
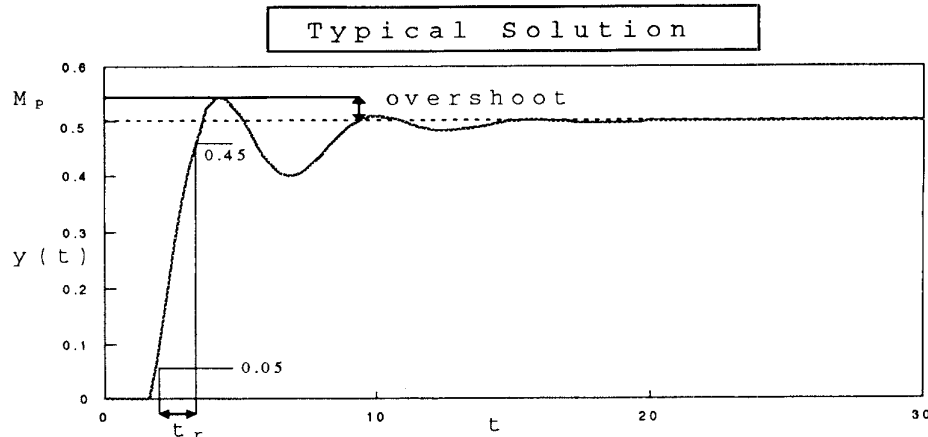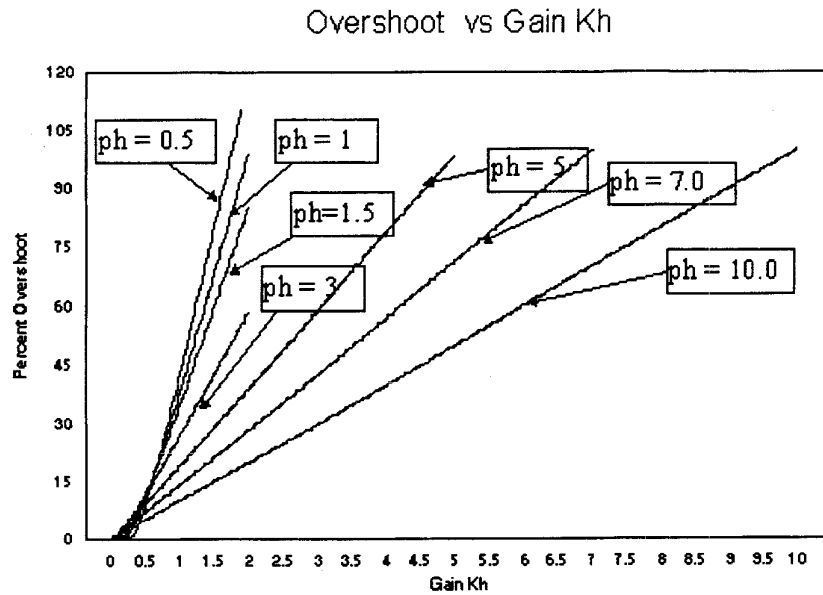
Fig. 3. A typical step response for a FOPTD plant.

Fig. 4. Overshoot versus gain, Kh, for different values of ph.

Case I: If $0 \le ph \le 0.5$, and $0 \le Kh \le 0.7$, then

$$M(p, K, h) = -0.5 + (38.943 + 78.182e^{-ph/0.60834})$$
$$\cdot \, [Kh - 0.009\,38 - 0.3706e^{-ph/1.109\,74}]^{1.5}.$$

Case II: If $0 \le ph \le 0.5$, and $0.7 \le Kh \le 1.5$, then

$$M(p, K, h) = (-49.49 + 55.46ph - 33.507p^2h^2)$$
$$+ \, Kh(99.572 - 71.6378ph + 38.6303p^2h^2).$$

Case III: If $0.5 \le ph \le 3$ then

$$M(p, K, h) = A(p, h) + B(p, h)Kh + C(p, h)e^{D(p, h)Kh}$$

where $A(p, h)$, $B(p, h)$, $C(p, h)$, and $D(p, h)$ are given as follows:

$$A(p, h) = -0.5967 + 0.07306ph - 59.467e^{-ph/1.0871}$$
$$B(p, h) = 22.9 - 1.346ph + 73.31e^{-ph/1.67}$$
$$C(p, h) = -0.430\,35 + 0.046\,11ph + 58.842e^{-ph/1.159}$$
$$D(p, h) = -2.221 + 0.058\,09ph - 0.8723e^{ph/1.6229}.$$

Case IV: If $3 \le ph \le 10$ then

$$M(p, K, h) = A_1(p, h) + [B_1(p, h)][Kh]$$

where

$$A_1(p, h) = 0.0578 - 0.005653ph - 58.493e^{-ph/1.1135}$$

and

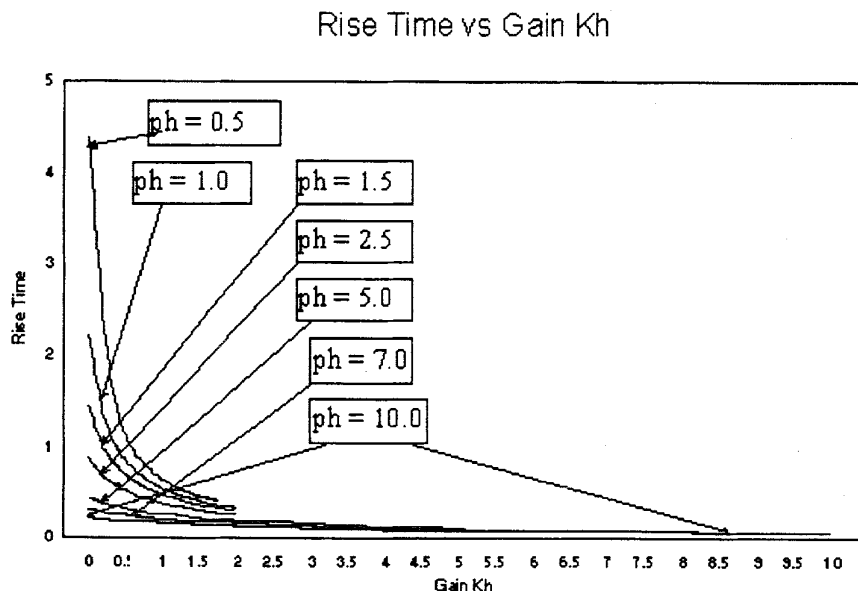$$B_1(p, h) = 15.424 - 0.637ph + 61.7669e^{-ph/2.3895}.$$

Rise Time vs Gain Kh

Fig. 5.   Rise time versus gain, Kh, for different values of ph.

2) Rise Time: Let t_r denote the rise time of the system in Fig. 4 for a step input. Then for $0.5 \leq ph \leq 10$

$$t_r = [A_2(p, h)e^{-Kh/B_2(p, h)} + C_2(p, h)e^{-Kh/D_2(p, h)}]h \quad (3)$$

where $A_2(p, h)$, $B_2(p, h)$, $C_2(p, h)$, and $D_2(p, h)$ are given as follows:

$$A_2(p, h) = 0.39955 - 0.035512ph + 9.4069e^{-2.2294ph}$$

$$B_2(p, h) = -0.14406 + 0.16277ph + 0.34998e^{-0.45502ph}$$

$$C_2(p, h) = 0.17204 - 0.005396ph + 1.0779e^{-0.45261ph}$$

$$D_2(p, h) = -1.3604 + 1.2889ph + 3.08798e^{-0.37192ph}.$$

All the above equations are accurate (r^2 coefficient of determination $= 0.99$) in their respective ranges and are only valid when the system is BIBO stable.

Example 2.1: Suppose $p = 1.6667$, $K = 1$, and $h = 1.82$. Since $K < p$, the system is BIBO stable independent of delay, and the above techniques may be used. Since $ph = 3.03394$, Case IV is used to obtain: $A_1(p, h) = -3.7945$, $B_1(p, h) = 30.843$. Therefore, the percent overshoot is estimated to be 52.34%. In fact, the actual percent overshoot via numerical simulation is given as 52.589%. This yields an error of 0.25%.

Using (3), it is obtained that $A_2(p, h) = 0.303174$, $B_2(p, h) = 0.43771$, $C_2(p, h) = 0.42879$, $D_2(p, h) = 3.548623$. This gives an estimate on the rise time as $t_r = 0.473$ s. The actual rise time, obtained from simulation, is given as 0.454 s.

Example 2.2: Suppose $p = 1$, $K = 3$, and $h = 1$. In this case, the system is not BIBO stable. For stability, we must have $K < 2.26182634$. Hence, the above formulas do not apply.

Example 2.3: Suppose K in Example 2.2 is free to be chosen. Suppose further, that the system design specifications are such that the rise time be approximately 1 s and that the percent overshoot be as low as possible. For fixed ph, the percent overshoot monotonically increases as Kh increases, as

shown in Fig. 4. Therefore, the smaller Kh, the less overshoot. By Fig. 5, it is seen that the rise time is approximately 1 s for $Kh \approx 0.375$. Therefore (since $h = 1$), choosing $K \approx 0.375$ will give the adequate performance subject to the design constraints. The rise time will approximately be 1 s and the percent overshoot can be read off Fig. 4 to be about 5%. Alternatively, more precise estimates can be obtained by using the above formulas, if desired. The next section will further demonstrate how the results in Section II can be used to aid in the design of PI controllers.

III. PROPORTIONAL-INTEGRAL (PI) CONTROL

A. Introduction

This section deals with the PI control of a FOPTD plant. As the introduction discussed, this paper is motivated by applications in the metal industry. These plants often have large normalized dead time, $ph \geq 0.5$, and sometimes have the following design criteria: 1) The step reference input response should have quick (optimal) settling time and small overshoot (sometimes 0%). There should be zero steady-state error. 2) Step load disturbance rejection should be fast to maintain tolerances after nominal setpoint response has been achieved. 3) The closed-loop system should not be sensitive to variations in process dynamics.

Fig. 6 illustrates one type of FOPTD plant with unit feedback and PI control. The variable $\tilde{W}(s)$ represents the load disturbance. It should be mentioned that Fig. 6 can be redrawn as in Fig. 7. In this case, the relationship between the constants L_P, L_I, and τ and the constants K_P, K_I, and p are given by

$$\tau = \frac{1}{p}$$

$$L_P = K_P \tau$$

$$L_I = \frac{K_I}{K_P}$$

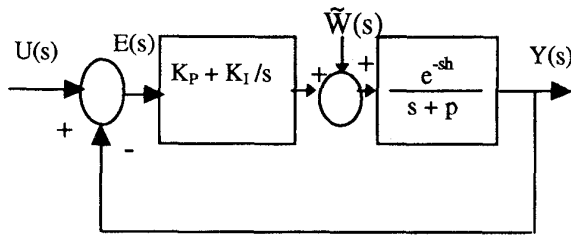$$w(t) = \tau \tilde{w}(t). \quad (4)$$

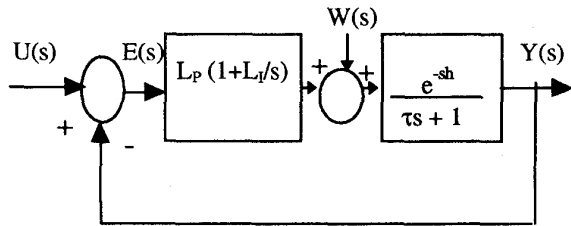Fig. 6. First-order plant with time delay and PI control.

Fig. 7. First-order plant with time delay and PI control.

Using the final value theorem, it is straightforward to verify that the steady-state errors for step inputs and step loads are zero when PI controllers are used. Therefore, the issue is not whether the output will track a reference step input, but rather how fast the tracking will occur (settling time, rise time, etc.). To measure performance of a PI controller, we often use the following procedure.

1) At $t = 0$ a unit step input is placed at the reference. The disturbance $w(t)$ is initially set equal to zero. The 2% settling time, denoted by t_s, and the percent overshoot are measured.

2) At $t = t_1$, sufficiently large, the load disturbance $w(t)$ is set equal to one. The 2% settling time for the step load is measured and denoted as t_d.

Finally, Section III-C presents a new PI control algorithm and discusses its sensitivity to process characteristics using frequency domain criteria. PI tuning algorithms are usually based on time-domain performance. Recent work of [2] and [9], however, suggests that it is also important to examine robustness properties of the closed-loop system, especially since the FOPTD model is usually an approximation of an unknown plant. For example, gain margins for robust systems should be between 2 and 5 [2]. Additionally, this paper uses the quantity M_s, the inverse of the shortest distance from the Nyquist curve to the critical point -1, to help quantify robustness. This sensitivity characteristic (see [2] for further details) is given by

$$M_s = \max_{0 \le \omega < \infty} \left| \frac{1}{1 + G_P(i\omega)G_C(i\omega)} \right|$$
$$= \max_{0 \le \omega < \infty} |S(i\omega)| \qquad (5)$$

where, for Fig. 6, $G_C(s) = K_p + K_I/s$, $G_P(s) = e^{-sh}/(s + p)$, and $S(s)$ denotes the sensitivity of the system.

It is usually desirable to require M_s to be in the range of 1.3 and 2. Smaller values of M_s improve robustness by guaranteeing stability, even if the gain is increased by a factor of $M_s/(M_s - 1)$ or decreased by a factor $M_s/(M_s + 1)$.

M_s also indicates a measure of robust stability in the face of nonlinear actuator characteristics, $f(x)$, inserted into the loop, satisfying $x M_s/(M_s + 1) < f(x) < x M_s/(M_s + 1)$. Additionally, the gain margin is greater than $M_s/(M_s - 1)$, and therefore, there is a link between M_s and gain margin. Likewise, [2] and [7] suggest that phase margins for robust systems should be greater than 60°. Large phase margins help ensure stability in the face of unknown/perturbed dead time, h. This can be explained by observing that when a FOPTD plant with PI control has phase margin ϕ_m and crossover frequency ω_c for a fixed delay h_0, then the closed-loop system will remain stable for delay $h = h_0 + \Delta h$ provided that $\Delta h < \phi_m/\omega_c$.

The above gain and phase margin rules are generalizations and therefore, are not applicable to all plants estimated by FOPTD models. However, some experimental and analytical verification of these rules of thumb have been reported [2] and they presently appear to be among the most accepted robustness measures in the literature.

B. PI Controllers with Pole-Zero Cancellation

For simplicity, this section discusses how to select PI control parameters for systems as modeled in Fig. 7. Conversion to parameters K_P and K_I in Fig. 6 is obtained using the simple algebraic relationships given in (4).

Several well-known PI tuning methods use the gain L_I to cancel the stable pole of the plant at $p = 1/\tau$. In particular, this section describes the Bryant [5], [12] and Haalman [7] method for selecting PI gains. The Bryant method has found widespread application in rolling mill control in the metal industries [4], [5], [7] (critically damped case). The Haalman method is similar to the Bryant method, but allows for more percent overshoot. These techniques have found prevalence in controlling systems with large dead time [2], which are usually difficult to control. It should be noted that the IMC PI also utilizes a pole-zero cancellation (this technique is not discussed here) [9].

Although the technique of using L_I to cancel the stable plant pole has found widespread application, its shortcomings have been well noted in the literature [9]. The setpoint response will be sluggish for processes with small ph. Additionally, when poles and zeros are canceled, there will be uncontrollable modes in the closed-loop response, leading to poor performance if the modes are excited [2].

Mathematically the rules for gain selection are given as follows:

Bryant [5]: Step 1: Select $L_I = L_{IB} = 1/\tau$.

Step 2: Select $L_P = L_{PB}$ using the following criteria:

1) $L_{PB}L_{IB}h = 0.368$ (for critically damped, $\zeta = 1.0$).
2) $L_{PB}L_{IB}h = 0.403$ (for slightly damped, $\zeta = 0.6$).
3) $L_{PB}L_{IB}h = 1.571$ (for oscillatory, $\zeta = 0.0$).

In a similar manner, Haalman has given the following algorithm for selecting PI controller gains:

Haalman [7]: Step 1: Select $L_I = L_{IH} = 1/\tau$.

Step 2: Select $L_P = L_{PH}$ to satisfy $L_{PH}L_{IH}h = 2/3$. The value of 2/3 was found by Haalman to minimize the mean square error for a step change in the setpoint.

In both the Bryant and Haalman algorithm, L_I is selected as $L_I = 1/\tau$, and the loop gain becomes

$$\frac{L_P\left(1 + \dfrac{\frac{1}{\tau}}{s}\right)e^{-hs}}{\tau s + 1} = \frac{\left(\dfrac{L_P}{\tau}\right)e^{-hs}}{s}. \tag{6}$$

Therefore, the pole-zero cancellation reduces Fig. 7 to the case of Fig. 2 with $K = L_P/\tau$ and $p = 0$. That is, the closed-loop system is equivalent to a first-order, delayed, astatic plant with unity feedback. The power of the results of Section II is that accurate estimates are obtainable for percent overshoot for PI controllers with $L_I = 1/\tau$. As outlined above, the Bryant algorithm presents a simple procedure for controlling percent overshoot in a qualitative manner only, while Haalman's technique provides no estimates on percent overshoot. Furthermore, the issue of rise time is not addressed in either method. This is further explained in the example below.

Example 3.1: Suppose that the plant is as in Fig. 7 with $\tau = 0.6$ and $h = 1.82$. Assume no load disturbance. Then, using the Bryant algorithm (critically damped, $\zeta = 1$, case), $L_{IB} = 1.66667$, and $L_{PB} = 0.1213$. Simulations show that these gains give no overshoot, a rise time of 5.98 s, and a settling time of 11.84 s. The Haalman method has controller gains $L_{IH} = 1.66667$, and $L_{PH} = 0.220$. This gives an overshoot of 17.5%, a rise time of 2.283 s, and a settling time of 13.47 s.

Suppose, now, it is desired to have less than 5% overshoot but rise time as quick as possible for a unit step reference input. Neither the Bryant nor Haalman method specifies how to select controller gains for design criteria like this. Such design criteria, however, are typical in many physical systems e.g., rolling mill control [4] where overshoot may be detrimental to the heavy equipment in motion.

By canceling out the plant pole, Section II techniques can be used to design a controller. It has been previously noted that for fixed p and h, the rise time decreases as K increases. Therefore, for these design criteria, it will be desirable to select the largest possible K that does not cause the overshoot to be too large. For $K = 0.28$ the percent overshoot is $\approx 5\%$ for the first-order plant with delay $h = 1.82$ and $p = 0$ (Case I). This implies that by choosing in Fig. 7 $L_P = 0.167$ and L_I the same as before, the new design criteria are met. Clearly, the new formulas proposed in Section II give additional power to the techniques of designing PI controllers with pole-zero cancellation. It is now possible to accurately estimate percent overshoot and rise time for different values of L_P (L_I still chosen to equal $1/\tau$).

C. Proposed New PI Controller

This section recommends PI tuning formulas that satisfy the design criteria 1)–3) in Section III-A. The results were obtained by using extensive numerical simulations [12] and curve fitting techniques [13] on the FOPTD model. The suggested new PI controller is given as $K_P + K_I/s$ where

TABLE I
OVERSHOOT, 2% SETTLING TIME, 2% SETTLING TIME FOR UNIT STEP LOAD, AND M_s FOR DIFFERENT ALGORITHMS WHEN $p = 1$, $h = 10$ ($ph = 10$)

	M_p (%)	t_s (sec)	t_d (sec)	M_s
Eq. (3.4)	0.0	27.2	38.4	1.68
Haalman [7]	17.5	74.1	76.3	1.91
Bryant [5,11]	0.0	65.2	86.025	1.39
Ziegler-Nichols [15,16]	0.0	1358.0	1368.7	1.10
ISE-Setpoint	18.5	64.8	75.7	2.41

Case I: If $0.01 \le ph \le 0.2$, then

$$K_P = K_P(p, h)$$
$$= -0.404p^2h + 0.723p + \frac{0.3852}{h}$$
$$K_I = K_I(p, h)$$
$$= -0.525p^2 + \frac{0.4104p}{h} - \frac{0.000\,24}{h^2}. \tag{7a}$$

Case II: If $0.2 \le ph \le 20$, then

$$KP = K_P(p, h)$$
$$= \frac{0.404}{h} + 0.256p - 0.1275\sqrt{\frac{p}{h}}$$
$$K_I = K_I(p, h)$$
$$= \frac{0.0808}{h^2} + \frac{0.719p}{h} - 0.324\sqrt{\frac{p}{h^3}}. \tag{7b}$$

PI controller (7a) and (7b) gives outstanding response to both setpoint changes and step disturbances, especially when ph is large. Overshoot for step inputs is almost always less than 1% and the settling time is exceptional. Fig. 8 gives plots of the output of the FOPTD plant with controller parameters given by (7a) and (7b). In Fig. 8(a), $ph = 10$, which is considered to be an extremely large normalized dead time. As previously mentioned, plants with large ph are considered difficult to control, and PI gains are often tuned using pole-zero cancellations. As seen in Fig. 8(a), however, the proposed controller (7a) and (7b) has superior settling time for both step reference inputs and step load disturbances when compared to other commonly used PI tuning laws. For example, controller (7a) and (7b) gives no overshoot, $t_s = 27.2$ and $t_d = 37.4$ (the load disturbance becomes nonzero at $t = 100$). As Table I shows, (7a) and (7b) gives outstanding settling times compared to other known PI tuning laws used for systems with large ph. Derivation of the ISE-setpoint PI tuning algorithm is given in the Appendix.

Fig. 8(b) and (c) plots the output of the system using various PI control laws and different values of ph. When $ph = 1$, PI controller (7a) and (7b) still gives outstanding performance. This is not surprising since (7a) and (7b) are designed to have setpoint following for systems with large normalized dead time ($ph \ge 0.5$). For $ph = 0.1$, Fig. 8(c) shows that the output still meets the design criteria; however, load disturbance rejection is slower. The results for $ph = 1$ and $ph = 0.1$ are summarized in Table II.

As Fig. 8(d) shows, for $0.01 \le ph \le 20$ the gain margin, G_m, is always greater than 2.5, and the phase margin, ϕ_m,
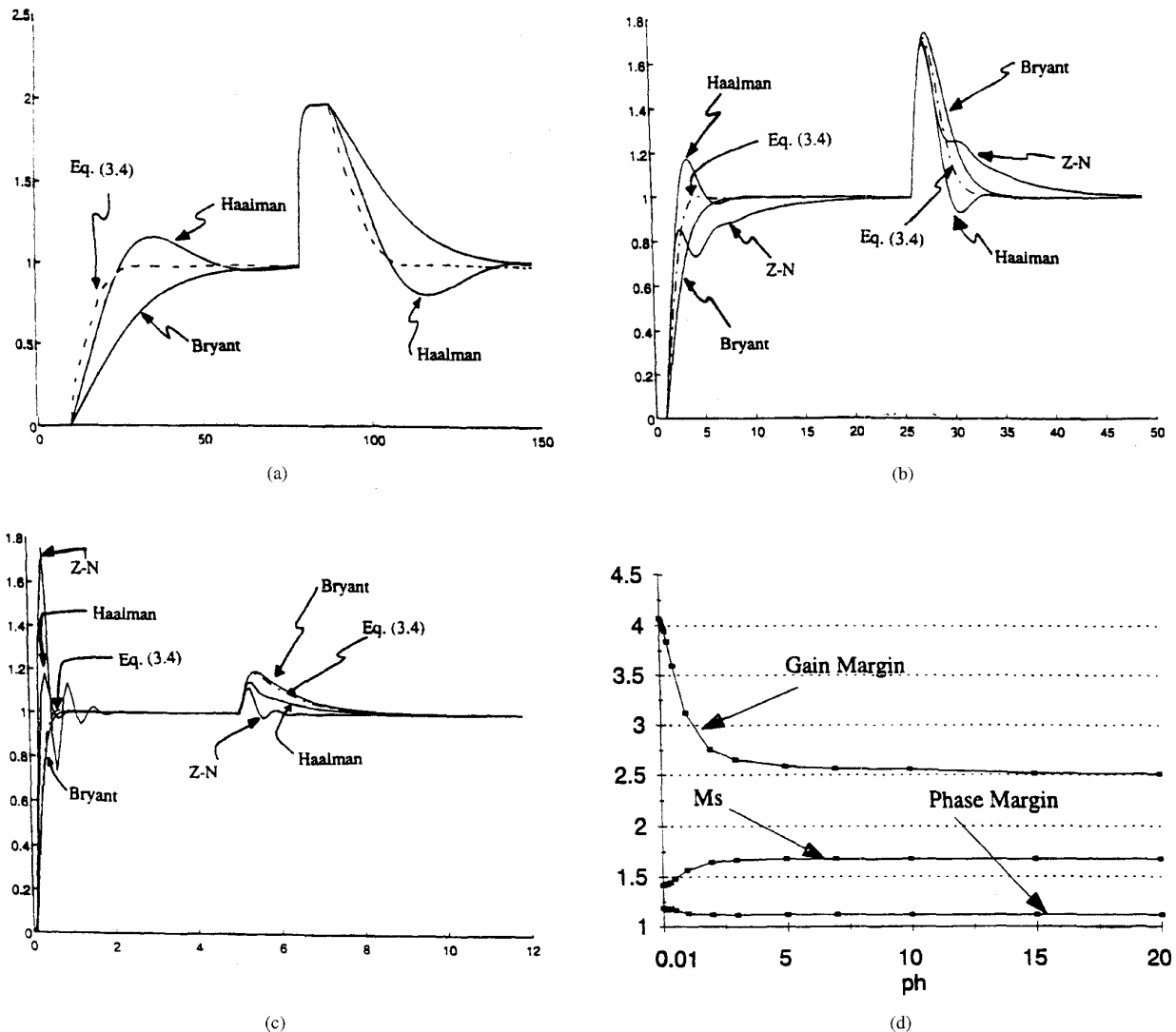
Fig. 8. Characteristics of proposed controller (7a) and (7b). (a)–(c) plot output versus time for $ph = 10$, $ph = 1$, and $ph = 0.1$, respectively. (d) plots gain margin, phase margin, and M_s versus ph.

TABLE II
OVERSHOOT, 2% SETTLING TIME, 2% SETTLING TIME FOR UNIT STEP LOAD, AND M_s FOR
DIFFERENT ALGORITHMS WHEN $p = 1$, $h = 1$ ($ph = 1$), AND $p = 1$, $h = 0.1$ ($ph = 0.1$)

| | ph = 1.0 | | | | ph = 0.1 | | | |
	M_P (%)	t_s (sec)	t_d(sec)	M_s	M_P (%)	t_s (sec)	t_d (sec)	M_s
Eq. (3.4)	0.42	3.84	7.23	1.57	0.44	0.552	2.73	1.43
Haalman [7]	17.5	7.4	7.16	1.92	17.53	0.741	2.16	1.92
Bryant [5,11]	0.0	6.525	9.18	1.39	0.0	0.652	2.88	1.39
Ziegler-Nichols [15,16]	0.0	17.04	19.12	1.31	76.5	1.532	0.90	3.63
ISE-Setpoint	13.83	9.42	11.0	2.29	22.6	1.259	2.90	2.14
Cohen-Coon [6]	37.25	10.36	10.03	2.64	82.07	1.786	0.92	3.96
ITAE-Setpoint [9]	1.95	3.63	6.81	1.63	3.25	0.614	2.51	1.56

is above 60° (for simplicity, Fig. 8(d) plots phase margin in radians). Additionally, M_s will always be between 1.3 and 2.0, which is as desired. In fact, none of the 11 benchmark PI tuning algorithms described in [9] guarantee both $G_m \geq$ 2.5 and $\phi_m \geq 60°$ for such a large range of normalized dead time i.e., $0.01 \leq ph \leq 20$.

It is interesting to note that PI controller (7a) and (7b) gives controller gains that are sometimes similar to the ITAE-setpoint PI controller [9] in the ranges $0.3 \leq ph \leq 1$. In particular, the integral gains correspond very closely. The proportional gains for (7a) and (7b) are usually lower than those for ITAE-setpoint, especially for smaller ph. The smaller

proportional gain has the effect of decreasing the overshoot and increasing the gain and phase margin at the expense of increasing the rise time.

IV. CONCLUSIONS

This paper introduces new tuning techniques for PI controllers based on FOPTD models. Proportional control is discussed, and formulas that relate plant parameters to percent overshoot and rise time are presented. These formulas are then used to aid in the design of PI controllers with pole-zero cancellation. In Section III-C, a PI controller is given in (7a) and (7b) that has outstanding settling time for step reference inputs and step load disturbances, especially for systems with large normalized dead time. Frequency domain criteria are used to show that (7a) and (7b) yields a robust closed-loop system. For the design criteria presented, typical of those found in rolling mills, (7a) and (7b) outperforms conventional PI controllers. All results were obtained by performing numerical simulations and curve fitting techniques.

APPENDIX

The work of [14] gives estimates for optimal ISE-setpoint PI controllers, derived by using MATLAB integration routines and quasi-Newton algorithms, for the ranges $0.1 \leq ph \leq 1$. This Appendix briefly discusses an alternate (unusual) technique for deriving the optimal ISE-setpoint PI controller $(K_P + K_I/s)$, based on complex variable theory, and then discusses the ISE-setpoint performance for $0.01 \leq ph \leq 20$. The tuning algorithms combine the results and techniques of [10] and [14] to obtain the simple tuning rules.

The cost function to be minimized is given as

$$J = \int_0^\infty e^2(t)\,dt$$

where $e(t) = \mathcal{L}^{-1}[E(s)]$, and $\mathcal{L}^{-1}$ denotes the inverse Laplace operator. If the system is stable, then Parseval's theorem [10] gives

$$J = \frac{1}{2\pi i} \int_{-i\infty}^{i\infty} E(s)E(-s)\,ds.$$

Using the method of residues and letting $h = 1$, [10] explicitly derives

$$J = \frac{1}{2\Delta}\left[(p^2 - \rho^2)\frac{K_I - K_P p + (p^2 - \rho^2)\cosh(\rho)}{K_P \rho^2 - pK_I - \rho(p^2 - \rho^2)\sinh(\rho)}\right.$$
$$\left. + (p^2 + \sigma^2)\frac{K_I = K_P p + (p^2 + \sigma^2)\cos(\sigma)}{K_P \sigma^2 + pK_I - \sigma(p^2 + \sigma^2)\sin(\sigma)}\right]$$

where $\Delta = \sqrt{(K_P^2 - p^2)^2 + 4K_I^2}$, $\rho = \sqrt{(\Delta + p^2 - K_P^2)/2}$, $\sigma = \sqrt{(\Delta - p^2 + K_P^2)/2}$. For $h \neq 1$, the cost is obtained by rescaling to obtain: $J(p, K_P, K_I, h) = hJ(ph, K_P h, K_I h^2, 1)$. An interesting aspect of this

approach is that it gives an explicit formula for J. This can be used as a measure of performance for the PI control algorithm (even in the case when optimizing the ISE-Setpoint is not the desired objective).

Using numerical optimization and curve fitting techniques [13], the rules for choosing K_{Popt} and K_{Iopt} which optimize J are given as

Case I: If $0.01 \leq ph \leq 0.2$, then

$$K_{Popt} = K_{Popt}(p, h)$$
$$= \frac{0.7388}{h} + 0.3185p$$
$$K_{Iopt} = K_{Iopt}(p, h)$$
$$= \frac{-0.000\,308\,2}{h^2} + \frac{0.5291p}{h}.$$

Case II: If $0.2 \leq ph \leq 20$, then

$$K_{Popt} = K_{Popt}(p, h)$$
$$= \frac{0.808}{h} + 0.511p - 0.255\sqrt{\frac{p}{h}}$$
$$K_{Iopt} = K_{Iopt}(p, h)$$
$$= \frac{0.095}{h^2} + \frac{0.846p}{h} - 0.381\sqrt{\frac{p}{h^3}}.$$

Choosing K_{Popt} and K_{Iopt} as above, an estimate on J is given by $J_{opt}(p, h) = 1.577h + 0.025ph^2 - 0.209\sqrt{ph^3}$.

REFERENCES

[1] B. Amstead, P. Ostwald, and M. Begeman, *Manufacturing Processes.* New York: Wiley, 1987.
[2] K. Astrom and T. Hagglund, *PID Controllers: Theory, Design, and Tuning.* Research Triangle Park, NC: Instrument Soc. Amer., 1994.
[3] C. Bissell, *Control Engineering.* New York: Chapman and Hall, 1994.
[4] S. Bouchillan and K. M. Finn, personal communications, ALCOA, Pittsburgh, PA, Apr. 1993/Mar. 1995.
[5] G. Bryant, *Automation of Tandem Mills.* London: Iron Steel Inst., 1973.
[6] G. Cohen and G. Coon, "Theoretical consideration of retarded control," *Trans. Amer. Soc. Mech. Eng.,* vol. 75, pp. 827–834, 1953.
[7] A. Haalman, "Adjusting controllers for a dead time process," *Contr. Eng.,* pp. 71–73, July 1965.
[8] C. Hang, W. Ho, and L. Cao, "A comparison of two design methods for PID controllers," *ISA Trans.,* vol. 33, pp. 147–151, 1994.
[9] W. Ho, C. Hang, and J. Zhou, "Performance and gain and phase margins of well-known PI tuning formulas," *IEEE Trans. Contr. Syst. Technol.,* vol. 3, pp. 245–248, June 1995.
[10] J. Marshall, H. Gorecki, K. Walton, and A. Korytowski, *Time-Delay Systems: Stability and Performance Criteria with Applications.* New York: Ellis Horwood, 1992.
[11] D. Schneider, "Control of processes with time delays," *IEEE Trans. Ind. Applicat.,* vol. IA-24, pp. 186–219, Mar./Apr. 1988.
[12] Simnon, *Simulation Language for Nonlinear Systems.* Dep. Automat. Contr., Lund Institute of Technology, Lund, Sweden, version 2.20, 1989.
[13] TableCurve, *Jandel Scientific TableCurve.* CA: ASIN Software, version 3.01, 1991.
[14] M. Zhuang and D. Atherton, "Automatic tuning of optimum PID controllers," in *IEEE Proc.-D,* vol. 140, no. 3, pp. 216–224, May 1995.
[15] J. Ziegler and N. Nichols, "Process lags in automatic-control circuits," *Trans. Amer. Soc. Mech. Eng.,* pp. 433–444, July 1943.
[16] _____, "Optimum setting for automatic controllers," *Trans. Amer. Soc. Mech. Eng.,* pp. 759–768, Nov. 1943.