PROCESS DESIGN AND CONTROL

Identification and Tuning of Integrating Processes with Deadtime and Inverse Response

William L. Luyben†

Process Modeling and Control Center, Department of Chemical Engineering, Lehigh University, Bethlehem, Pennsylvania 18015

The classical example of an integrating process with inverse response is level control of a boiler steam drum. The "boiler swell" problem can lead to a transfer function between the drum level and boiler feedwater flow rate that contains a pure integrator and a positive zero, in addition to some deadtime and lags. This paper presents a procedure for identifying the transfer-function parameters for this type of system from step response data. A proportional−integral controller tuning procedure is also presented. Because the process contains an integrator and the proportional−integral controller also contains an integrator, controller tuning is somewhat complex. The proposed method determines the smallest possible value for integral time. Then, using this value, the controller gain that gives a +2 dB maximum closed-loop log modulus is calculated. Simple Matlab programs are given for performing these calculations. The use of proportional−integral−derivative control is also briefly studied, but the typically noisy level signals preclude the use of derivative action.

1. Introduction

Integrating processes are common in chemical engineering. The most frequently encountered is the liquid level in a vessel. The transfer function contains a "1/*s*" term. Integrating processes that also include other dynamic elements are also encountered. The combination of integration and inverse response gives a particularly interesting and difficult-to-control process.

The classical example of an integrating process that has an inverse response is a boiler steam drum. The level is controlled by manipulating the boiler feedwater (BFW) to the drum. The drum is located near the top of the boiler and is connected to it by a large number of tubes. Liquid and vapor water circulate between the drum and the boiler as a result of the density difference between the liquid in the "downcomer" pipes leading from the bottom of the drum to the base of the boiler and the vapor/liquid mixture in the "riser" pipes going up through the boiler and back into the steam drum. Saturated steam comes off the top of the drum and is typically superheated by flowing back into the boiler before going to a steam turbine driving an electric generator.

If the BFW is significantly colder than the temperature of the saturated steam in the steam drum, increasing the flow of BFW can cause a temporary decrease in the liquid level because the cold feed reduces the amount of vapor in the boiler tubes. The reduction in the gas volume in the boiler tubes means an increase in the liquid volume in these tubes, so the liquid level in the drum drops. Of course, this is only a temporary phenomenon. Eventually, the increase in the BFW will drive the level up. The long-term response is essentially a ramp for a step change in the BFW because the drum is an integrator.

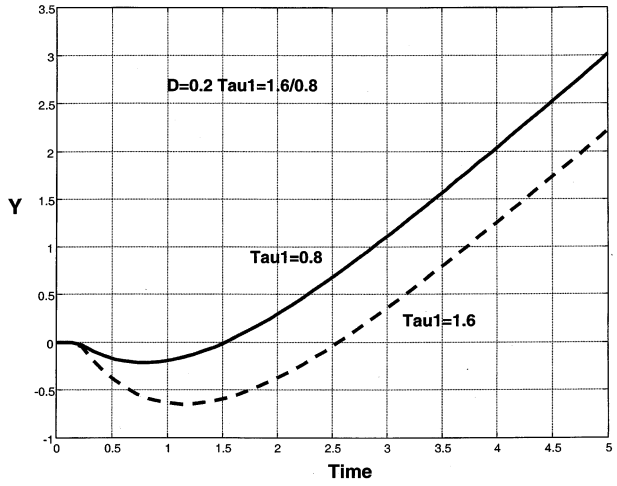

Figure 1. Open-loop step response of the integrator process with inverse response.

Figure 1 illustrates this type of response. The open-loop process transfer function for the process is

$$G_{M(s)} = \frac{K_P(-\tau_1 s + 1)e^{-Ds}}{(\tau_2 s + 1)s} \tag{1}$$

The τ time constants and the deadtime D have units of time. The positive zero is located at $+1/\tau_1$. Note the pure integrator ("*s*" in the denominator), which requires the parameter K_P to have units of reciprocal time. Figure 1 shows responses for two values of τ_1. The other param-

† Tel.: 610-758-4256. E-mail: WLL0@Lehigh.edu.

10.1021/ie020935j CCC: $25.00
Published on Web 05/15/2003

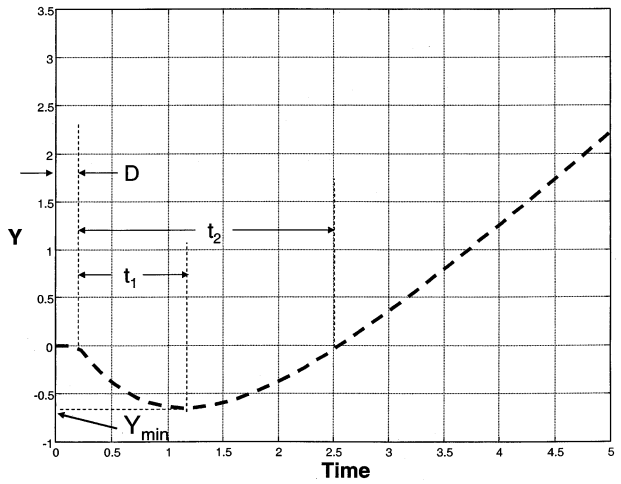

Figure 2. Identification parameters from the step test.

eter values of the transfer function are $D = 0.2$, $K_P = 1$, and $\tau_2 = 1$.

Many papers have discussed tuning methods for processes with inverse response and/or deadtime, but most of these do not consider integrating processes. The literature in this area is summarized in several papers.[1−4]

2. Identification

2.1. Proposed Procedure. There are four parameters in the transfer function to be estimated. If the process is subjected to a step input Δu, the deadtime can be easily determined, as shown in Figure 2. This leaves three remaining parameters: K_P, τ_1, and τ_2.

The time response of the process, assuming for the moment that $D = 0$, can be easily found (for example, using the Matlab function "ilaplace").

$$y_{(t)} = \Delta u K_P[t - \tau_1 - \tau_2 + (\tau_1 + \tau_2)e^{-t/\tau_2}] \qquad (2)$$

The derivative of this function is

$$\frac{dy}{dt} = \Delta u K_P\left[1 - \left(\frac{\tau_1 + \tau_2}{\tau_2}\right)e^{-t/\tau_2}\right] \qquad (3)$$

It is clear from Figure 2 that the derivative is zero at time $= t_1$, where the value of the output is Y_{min}. In addition, at time $= t_2$, the value of Y is zero.

Because we can read from the output curve t_1, t_2, and Y_{min}, these values are known. Of course, we know the magnitude of the step input Δu. Then we apply eq 2 twice and eq 3 once.

At $t = t_1$:

$$y_{min} = \Delta u K_P[t_1 - \tau_1 - \tau_2 + (\tau_1 + \tau_2)e^{-t_1/\tau_2}] \qquad (4)$$

$$\frac{dy}{dt} = 0 = \Delta u K_P\left[1 - \left(\frac{\tau_1 + \tau_2}{\tau_2}\right)e^{-t_1/\tau_2}\right] \qquad (5)$$

At $t = t_2$:

$$0 = \Delta u K_P[t_2 - \tau_1 - \tau_2 + (\tau_1 + \tau_2)e^{-t_2/\tau_2}] \qquad (6)$$

Equations 4−6 provide three equations in the three unknowns.

These three equations are highly nonlinear and cannot be solved analytically. However, the Matlab function "fsolve" can easily be used to numerically find the solution. The method converges quite robustly if the initial guesses of the three unknowns are not too far away from the true values.

```
% Program "inversecalc.m"
%    use fsolve to solve for parameters
%
%   Data from step test
ymin=-0.36;t1=0.35;t2=0.85;
parameters=[ymin t1 t2];
%  Initial Guesses
xo=[1 1 1]';
options=optimset('MaxFunEvals',1000);
%  Use fsolve to solve three nonlinear algegraic equations
[x,fval]=fsolve(@inverse,xo,options,parameters);
kp=x(1,1)
tau1=x(2,1)
tau2=x(3,1)

% function "inverse.m"
%    Used by "fsolve" to find solution of 3 nonlinear algebraic equations
%
function f=inverse(x,parameters)
kp=x(1,1);
tau1=x(2,1);
tau2=x(3,1);
ymin=parameters(1);
t1=parameters(2);
t2=parameters(3);
%  Three nonlinear algebraic equations
  f(1,1)=ymin-kp*(t1-tau1-tau2+(tau1+tau2)*exp(-t1/tau2));
  f(2,1)=1-(tau1+tau2)*(exp(-t1/tau2))/tau2;
  f(3,1)=t2-tau1-tau2+(tau1+tau2)*(exp(-t2/tau2))/tau2;
```

Figure 3. Matlab programs to solve for transfer-function parameters.

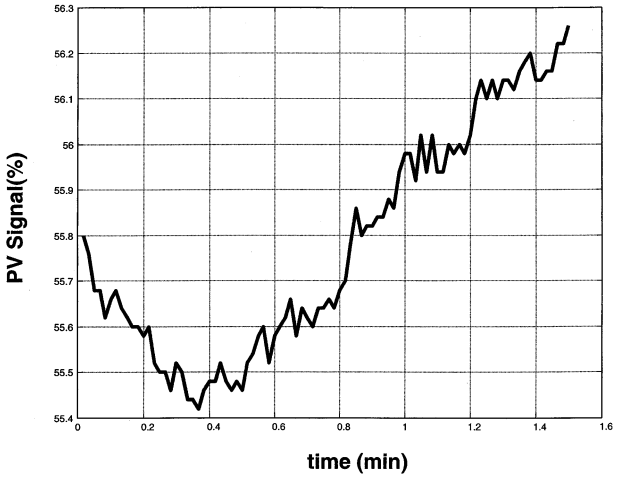

Figure 4. Example open-loop step response.

Figure 3 lists two Matlab m-files. The first, inversecalc.m, contains the experimental data points (t_1, t_2, and Y_{min}) and sets up initial guesses of the unknowns (K_P, τ_1, and τ_2).

2.2. Example. Typical data are plotted in Figure 4. The transmitter output signal (PV) is plotted in percent of scale. The step input is a 10% change in the signal sent to the valve (OP). The values read from the data are $D = 0.1$ min, $t_1 = 0.35$ min, $t_2 = 0.85$ min, and ΔY_{min} at $t_1 = -0.36\%$. Note that Y is expressed as the change in Y from the steady-state value.

Initial guesses of unity are made for all variables. The programs given in Figure 3 give the following values calculated from the data: $\tau_1 = 0.418$ min, $\tau_2 = 1.06$ min, and $K_P = 0.547$ min^{-1}.

3. Controller Tuning

Once the parameters of the transfer function are known, the next step is to find reasonable tuning constants. The use of proportional-only control is recommended for most level control loops. However, in the

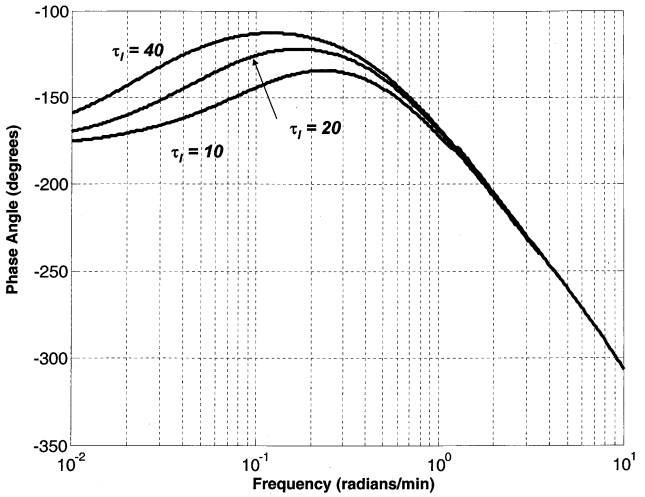

Figure 5. Open-loop phase angle plot.

```
% Program "tauimincalc.m"
%  Uses "fsolve" to find minimum taui that gives +2 dB Lc
%

d=0.1;tau1=0.418;tau2=1.06;kp=0.547;

parameters=[d tau1 tau2 kp];
options=optimset('MaxFunEvals',1000);
%  Initial guesses of tauimin and wmin
xo=[40 0.15]';
[x,fval]=fsolve(@tauimin,xo,options,parameters);
tauimin=x(1,1)
wmin=x(2,1)


% function "tauimin.m"
%    Calls "fsolve" to find solutions of two nonlinear algebraic equations
function f=tauimin(x,parameters)
taui=x(1,1);
w=x(2,1);
d=parameters(1);
tau1=parameters(2);
tau2=parameters(3);
kp=parameters(4);
f(1,1)=(120*pi/180) -w*d - atan(w*tau1)-atan(w*tau2)+atan(w*taui)-pi;
f(2,1)=-d - tau1/(1+(w*tau1)^2) - tau2/(1+(w*tau2)^2) + taui/(1+(w*taui)^2)
```

Figure 6. Matlab programs to solve for minimum integral time.

boiler steam−drum application, the use of proportional−integral (PI) control is normally required for several reasons. First, liquid holdup in the steam drum is typically quite small compared to the total holdup of water in the entire boiler. Second, there are hard constraints on how low and how high the level can fall or rise. The upper limit is set to prevent water from being carried over into the turbine and to prevent damage of the demisting equipment. The lower limit is set by the need to maintain a high circulation rate, which is driven by density differences and the height of the liquid leg.

The use of proportional−integral−derivative (PID) control is rare because the level signal is typically quite noisy. The use of feedforward control is standard. This is called "three-element" level control. The steam flow rate signal is used to anticipate the need for changes in the water flow. We do not consider this structure in this paper, but the use of PID control is briefly considered to illustrate its advantages.

3.1. Setting Integral Time. The transfer function of a PI controller contains an "*s*" in the denominator.

$$G_{C(s)} = \frac{K_C(\tau_I s + 1)}{\tau_I s} \tag{7}$$

The process transfer function also contains an integrator, as shown in eq 1. These two integrators in series make tuning difficult. This difficulty can best be explained from a frequency-domain perspective. Each integrator contributes 90° of the phase angle lag. This means that the phase angle of the total process transfer function $G_C G_M$ starts at low frequencies at a phase angle of −180°.

Figure 5 gives phase angle plots for several values of integral time τ_I. These plots start at −180°. For large values of τ_I, the phase angle increases as the lead in the controller transfer function adds positive phase angle. However, if τ_I is too small, the positive phase angle from the lead occurs after the negative phase angles from the positive zero, the lag, or the deadtime in the process transfer function have already dropped the phase angle below −180°. Thus, the phase angle never rises far enough above −180° to provide a reasonable phase margin.

It is clear that there is a minimum value of the integral time that gives a peak in the phase angle curve that is high enough for reasonable closed-loop robustness. If we select a design criterion that the peak phase angle must be about −120° (−2.09 rad), this minimum value of τ_I can be found by using the equations for the phase angle.

$$\arg(G_M G_C) = -wD - \pi - \arctan(w\tau_1) - \arctan(w\tau_2) + \arctan(w\tau_I) \tag{8}$$

The second term on the right-hand-side of the equation ("$-\pi$") comes from the two integrators, one from the process and one from the PI controller. Note that the positive zero (τ_1) contributes a *negative* phase angle, as does the lag (τ_2). The lead (τ_I) in the controller contributes a positive phase angle.

The derivative of eq 8 with respect to frequency ω is

$$\frac{\mathrm{d}[\arg(G_M G_C)]}{\mathrm{d}\omega} = -D - \frac{\tau_1}{1+(w\tau_1)^2} - \frac{\tau_2}{1+(w\tau_2)^2} - \frac{\tau_I}{1+(w\tau_I)^2} \tag{9}$$

The derivative is equal to zero at the peak in the phase angle curve. Setting eq 8 equal to −2.09 rad and setting eq 9 equal to zero give two equations in two unknowns: the values of the minimum integral time $(\tau_I)_{min}$ and the frequency ω_{min} at which this peak in the phase angle curve occurs. See Figure 5.

These two nonlinear algebraic equations cannot be solved analytically but can be solved numerically using Matlab's "fsolve" function. The programs for achieving this are given in Figure 6.

Initial guesses of the two unknowns must be made. These can be estimated by looking at the open-loop phase angle curve given in Figure 5. It is clear that the phase angle never comes up to −120° if the integral time is less than about 40 min. With this value of τ_I, the peak in the curve occurs at about 0.15 rad/min. So the guessed values of the two variables are set at 40 and 0.15, respectively, in the program shown in Figure 6.

The result of the calculation is a minimum integral time of 23 min. Note that this integral time is unexpectedly large compared to the other time constants. However, it cannot be made any smaller if reasonable robustness is desired and if a PI controller is used. The use of derivative action permits a reduction in integral time, as will be illustrated later in this paper. However,

```
% "swellfindkc.m"
%
%  For positive zero (tau1), lag, deadtime and integrator process
% Given tau1, tau2, Kp and d,   Set taui =  minimum taui
% Find kc of PI controller that gives closedloop log modulus = +2 dB
d=0.1;tau1=0.418;tau2=1.06;kp=0.547;
% Set taui = taui minimum
taui=23.0;
w=logspace(-2,2,300);
nw=length(w);
%
% Calculate ultimate gain and frequency
for n=1:nw
  argg(n)=-w(n)*d-atan(w(n)*tau1)-atan(w(n)*tau2)-pi/2;
end
k=1;while argg(k)>-pi;k=k+1;end
wu=w(k)
pu=2*pi/wu
ku=wu*sqrt(1+(wu*tau2)^2)/sqrt(1+(wu*tau1)^2)/kp
%
% Find kc that give Lcmax = +2 dB
kc=ku/2;
dkc=ku/10;
flagm=-1;
flagp=-1;
error=10;
loop=1;
%  Iterative loop to calculate closedloop log
modulus
while error>0.01
  loop=loop+1;
  if loop>100
    error('loop>100')
    end
% Generate closedloop log modulus for PI
controller
s=i*w;
num=kp*[-tau1 1];
den=conv([tau2 1],[1 0]);
g=polyval(num,s) ./ polyval(den,s);
gdead=g .* exp(-d*s);
% PI controller
numgc=kc*[taui 1];
dengc=[taui 0];
gc=polyval(numgc,s) ./ polyval(dengc,s);
gtot= gdead .* gc;
gcl= gtot ./ (1+gtot);
cldb=20*log10(abs(gcl));
dbmax=max(cldb);
dberror=dbmax-2;
error=abs(dberror);
%  Use interval halving to find Kc
if dberror>0
  if flagm>0
    dkc=dkc/2;
  end
  flagp=1;
  kc=kc-dkc;
end
if dberror<0
  if flagp>0
    dkc=dkc/2;
    end
    flagm=1;
    kc=kc+dkc;
 end
end
kc
dbmax
clf
semilogx(w,cldb)
grid
xlabel('Frequency (rad/time)')
ylabel('Closedloop Log Modulus (dB)')
title(d=0.1;tau1=0.418;tau2=1.06;kp=0.547')
```

Figure 7. Matlab program to find the controller gain that gives a +2 dB maximum closed-loop log modulus.

derivative action can seldom be used in these applications because of the noisy level signal.

3.2. Finding Controller Gain. Once the value of τ_I is established, the final job is to determine a reasonable controller gain K_C. One of the most useful and practical frequency-domain methods for doing this is to find the controller gain that gives a peak in the closed-loop log modulus curve that is +2 dB. This criterion gives a reasonable closed-loop damping coefficient of about 0.4 and reasonable robustness. The closed-loop log modulus is the magnitude of the closed-loop servo transfer function expressed in decibels.

$$L_C = 20 \log \left| \frac{G_C G_M}{1 + G_C G_M} \right| \qquad (10)$$

Figure 7 lists a Matlab program that finds the value of K_C that gives a +2 dB peak by iteratively generating $L_{C(i\omega)}$ plots. Note that the ultimate gain is calculated, and the initial guess of controller gain is set at half the ultimate gain. The results of the calculation give a controller gain $K_C = 0.854$ for the example process considered earlier.

Figure 8 gives the closed-loop log modulus curve with $K_C = 0.854$ and $\tau_I = 23$ min. Figure 9A shows the response of the closed-loop system for a unit step change in setpoint using a PI controller with these tuning parameters.

The load response is shown in Figure 9B. The PI controller takes a long time to return the process to the setpoint because of the large integral time. In the next section, we show that considerable improvement could be achieved if PID control could be used.

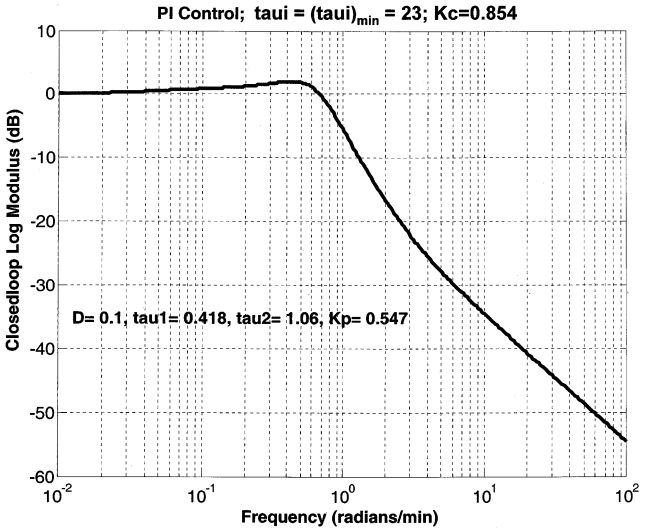

Figure 8. Closed-loop log modulus plot with PI control.

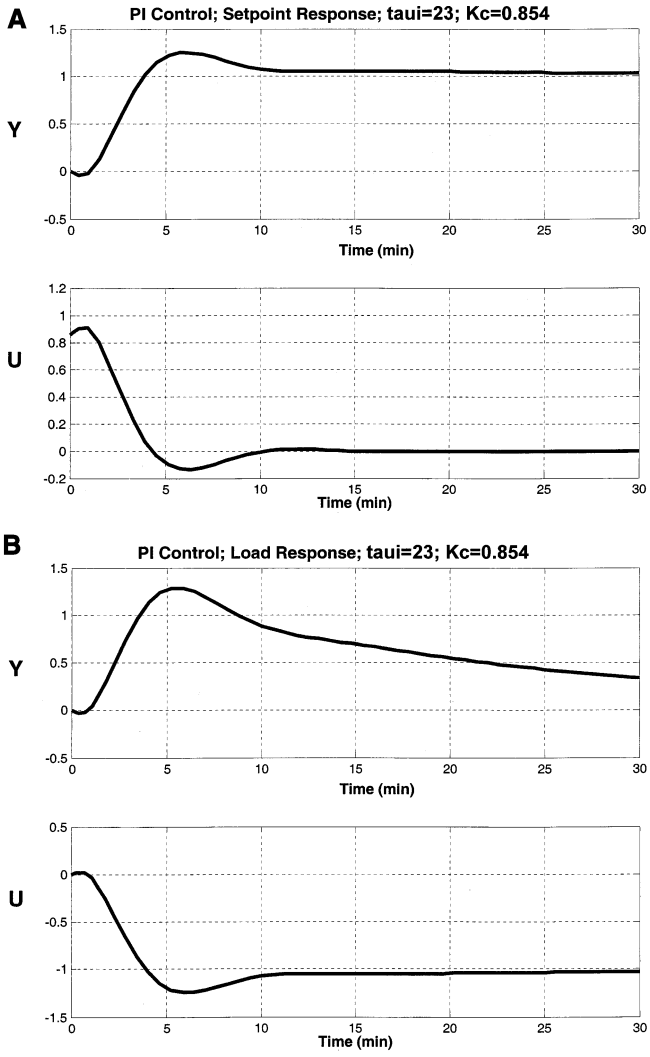

Figure 9. (A) Unit step setpoint response with PI control. (B) Load response with PI control.

3.3. Summary of the Proposed Procedure. The proposed identification and tuning procedures for an integrating process with inverse response and deadtime are summarized below:

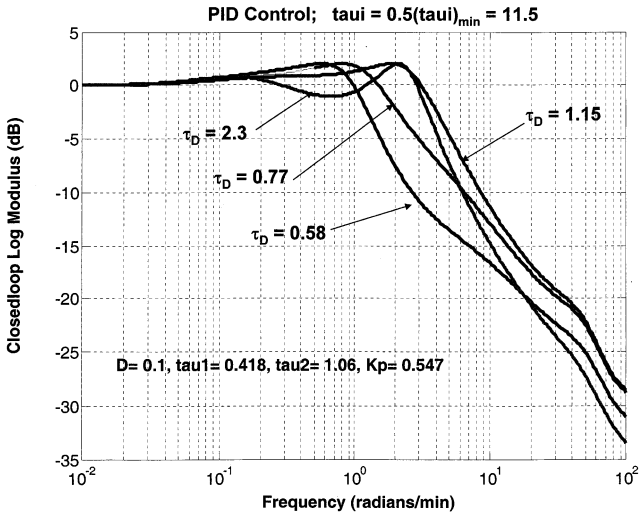

Figure 10. Closed-loop log modulus plot with PID control (50% minimum integral time).

1. Make a step change in the manipulated variable.
2. Read from the output response curve the deadtime, the time the output curve reaches its minimum (or maximum) value, the value of the output at this minimum, and the time the output curve reaches its initial steady-state value.
3. Solve the three simultaneous equations (eqs 4−6) for the three unknown parameters of the open-loop transfer function.
4. Solve eqs 8 and 9 for the integral time.
5. Find the value of the controller gain that gives a +2 dB maximum closed-loop log modulus.

3.4. PID Control. If the process signal is not too noisy, the use of derivative action can improve the closed-loop dynamics in this system because derivative action permits the use of smaller integral times. The positive phase angle produced by the derivative lead/lag raises the phase angle curve so the minimum integral time becomes smaller.

The transfer function of a PID controller is

$$G_{C(s)} = \frac{K_C(\tau_I s + 1)}{\tau_I s} \frac{(\tau_D s + 1)}{(\alpha \tau_D s + 1)} \qquad (11)$$

where τ_D is the derivative time and the parameter α is typically 0.1.

The tuning procedure proposed for the PID controller consists of the following steps:

1. Set the integral time τ_I. We wish to have a value smaller than the minimum PI integral time so that the load response is improved. Several values were explored, and the results are given for integral times that are 50% of the minimum (τ_I = 11.5 min) and 25% of the minimum (τ_I = 5.75 min).
2. Pick a value of the derivative time τ_D (to be varied).
3. Find the value of the controller gain that gives a +2 dB maximum closed-loop log modulus.
4. Vary the derivative time τ_D over a range of values and find the value that gives the *maximum* controller gain.

Note that there is a limited feasible range for τ_D values for which a controller gain can be found that achieves the +2 dB criterion. Figure 10 illustrates the effect of the derivative time on the closed-loop log modulus curves when the integral time is fixed at half

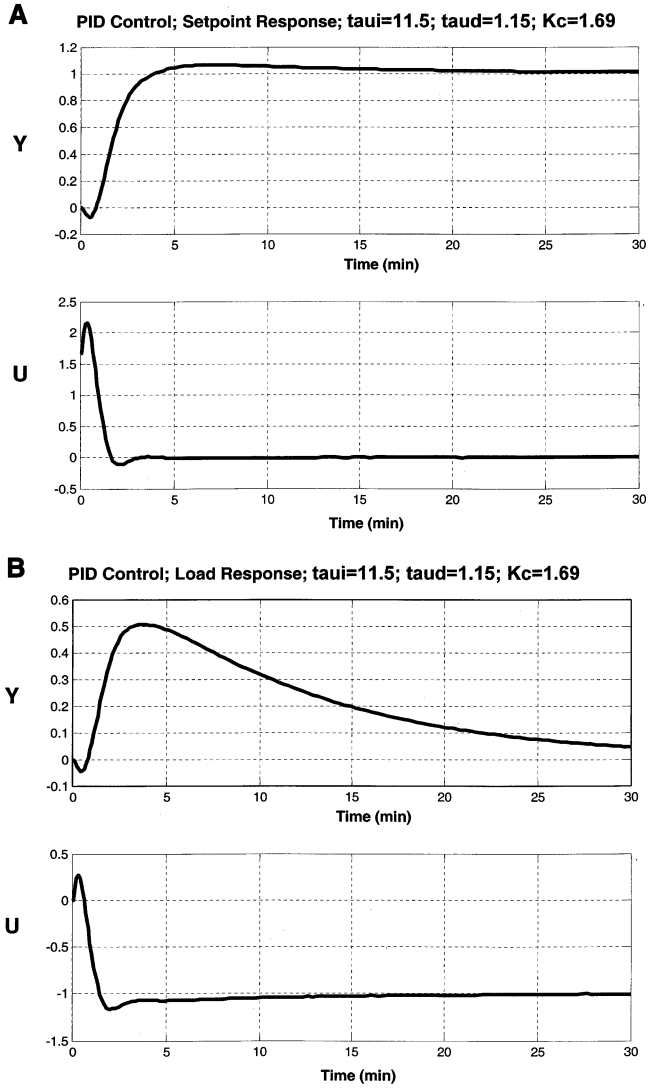

Figure 11. (A) Unit step setpoint response with PID control (50% minimum integral time). (B) Load response with PID control (50% minimum integral time).

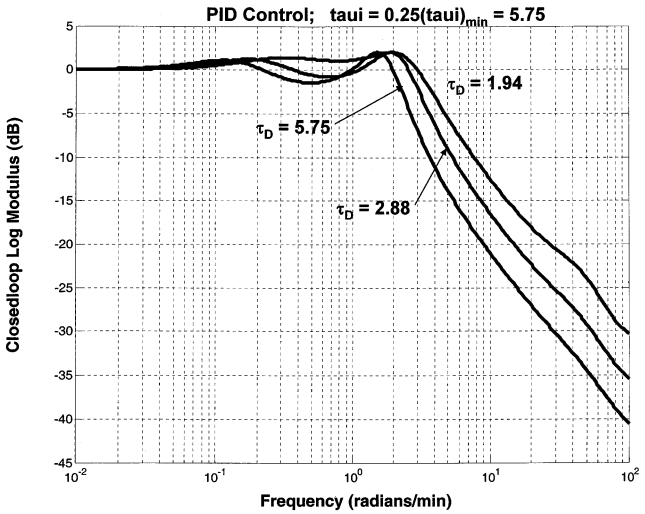

Figure 12. Closed-loop log modulus plot with PID control (25% minimum integral time).

the minimum (for a PI controller). Each of these curves has a different controller gain.

As τ_D increases, the peak moves to the right (higher frequencies indicating smaller closed-loop time con-

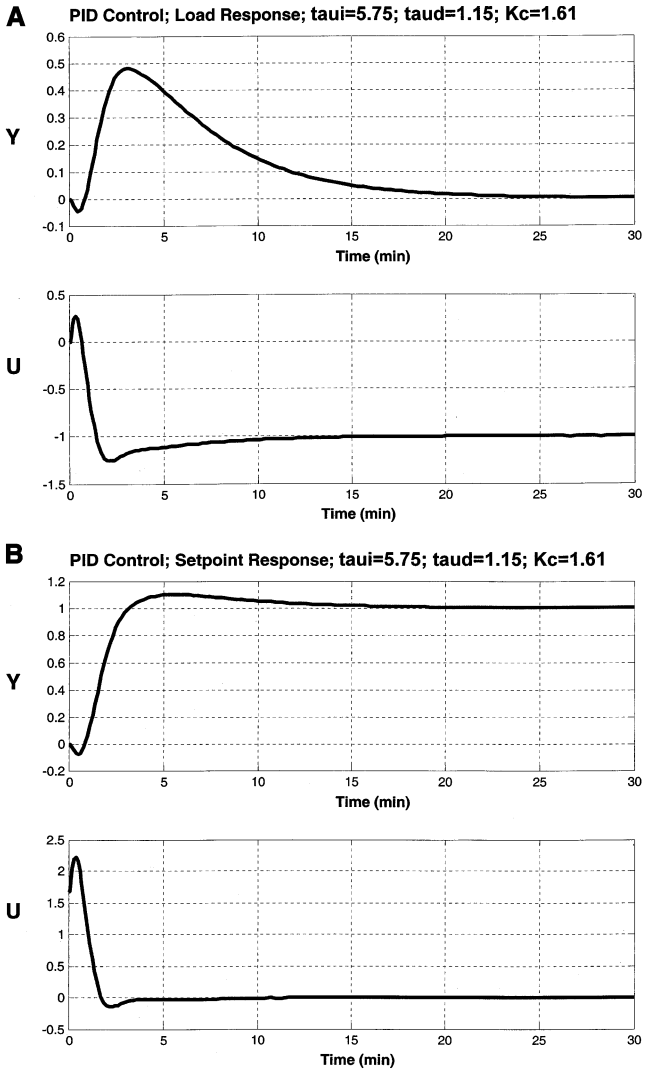

Figure 13. (A) Unit step setpoint response with PID control (25% minimum integral time). (B) Load response with PID control (25% minimum integral time).

stants). If τ_D is made too large, there is little improvement in the bandwidth but a droop begins to appear in the lower frequency part of the curves, indicating poorer low-frequency performance.

Results of the calculations are given in Table 1. A value of τ_D is selected that gives the maximum controller gain. When integral time is half the minimum, the optimum derivative time is 1.15 min, using a controller gain of 1.69. Figure 11 gives simulation results for the two cases: a setpoint change and a load disturbance.

Figure 12 illustrates the effect of derivative time on the closed-loop log modulus curves when the integral time is fixed at one-fourth the minimum for a PI controller. The optimum derivative time is 1.15 min, using a controller gain of 1.61. Lower values of τ_D are unfeasible (a controller gain that gives +2 dB cannot be found). Figure 13 gives setpoint and load responses for this case.

The responses of the PID controllers are much better than those of the PI, as expected. However, it should be remembered that noisy signals limit the use of derivative action, so the use of feedforward control with PI control may be required to improve the load response.

Table 1. PID Controller Tuning

τ_I (min)	τ_D (min)	K_C (min^{-1})	τ_I (min)	τ_D (min)	K_C (min^{-1})
11.5	0.576	1.29	5.75	1.15	1.61
	0.767	1.66		1.44	1.40
	1.15	1.69		2.88	0.774
	2.3	0.967		5.76	0.436

4. Conclusion

Identification and controller tuning procedures have been proposed for integrating processes with inverse response and deadtime. The method is easy to use and gives reliable controller tuning parameters.

It should be noted that noisy signals will make it more difficult to pick off the various points on the trajectory. Conventional filtering methods could be used for reasonable signal-to-noise ratios, but the method is probably not applicable when noise levels are high.

Nomenclature

D = deadtime (min)
G_C = controller transfer function
G_M = process open-loop transfer function
K_C = controller gain
K_P = process steady-state gain (min^{-1})
L_C = closed-loop log modulus (dB)
s = Laplace transform variable
t_1 = time when the response reaches a minimum (min)
t_1 = time when the response passes through the original steady-state value (min)
U = manipulated variable, process input (%)
Y = controlled variable, process output (%)
α = derivative parameter
τ_D = derivative time constant (min)
τ_1 = time constant of positive zero in a process open-loop transfer function (min)
τ_2 = time constant of lag in a process open-loop transfer function (min)
τ_I = integral time constant in the PI controller (min)
ω = frequency (rad/time)
ω_{min} = frequency where the peak in the open-loop phase angle curve occurs (rad/time)

Literature Cited

(1) Luyben, W. L. Tuning of proportional−integral controllers for processes with both inverse response and deadtime. *Ind. Eng. Chem. Res.* **2000**, *39*, 973−976.

(2) Shinskey, F. G. PID-deadtime control of distributed processes. *Control Eng. Pract.* **2001**, *9*, 1177−1183.

(3) Chien, I. L.; Peng, S. C.; Li, J. H. Simple method for integrating processes with long deadtime. *J. Process Control* **2001**, *12*, 391−404.

(4) Majhi, S.; Atherton, D. P. Autotuning and controller design for processes with small time delays. *IEE Proceedings: Control Theory and Applications* **1999**, *146* (5), 415−425.

Received for review November 21, 2002
Revised manuscript received February 13, 2003
Accepted April 15, 2003

IE020935J