A Two Step Closed Loop Parameter Estimation Technique for Self-Tuning Process Control

John D. Lane
Senior Applications Engineer
Bailey Controls Company

Tom J. Scheib
Chief Application Engineer
Bailey Controls Company

M. A. Keyes IV
Senior Group V P
McDermott International

Presented at
The AIChE Annual Meeting
Washington, D C
Nov 27 Dec 2, 1988
A Two Step Closed Loop Parameter Estimation Technique

for

Self-Tuning Process Control

John D. Lane
Senior Application Engineer
Bailey Controls Company

Tom J. Scheib
Chief Application Engineer
Bailey Controls Company

M.A. Keyes IV
Senior Group V.P.
McDermott International

Abstract

A general purpose closed loop parameter estimation technique has been developed for self-tuning process control applications. This technique is applicable to a wide class of industrial control problems including those with large unknown and time varying processes. The technique works by combining the continuous feed forward parameter estimates on and parameter tracking attributes of a recursive least squares algorithm with periodic dead time estimates.

The two step parameter estimation technique is coupled with a mode based control algorithm as a self-tuning auto-tuner. This configuration offers tight control for a wide variety of processes through inherent dead time compensation. Design and simulation testing have validated this methodology in conjunction with a PID auto-tuner.

This paper provides a description of the individual components of the self-tuning control system with a description of the special considerations required to integrate the two independent parameter estimators. Results of simulation studies are presented.

Introduction

Industrial processes with large unknown and/or time varying processes de-rank among the most difficult to control. Self-tuning methods present a way to handle this class of problems due to a reliance on recursive least squares parameter estimation and the PID control algorithm. Recursive least squares parameter estimation on the basis of a majority of self-tuning control tasks is feasible because of a requirement for explicit knowledge of the process dead time. Time varying control systems composed of a PID controller are unable to directly compensate for process dead time. To maintain stability, these control systems must be detuned degrading overall control performance.

One mechanism for extending self-tuning control in these cases is to couple a recursive least squares parameter estimator on a given basis with a dead time model and augment the control function with dead time compensation. Two particular methods one employing a bank of estimators with different assumed values or the dead time and a second employing a single estimator to identify a set of parameters extended over the time horizon of an assumed deadtime, have received much attention. Unfortunately these techniques, along with other parameter estimation applications, see [3] for a survey, have proven impractical for general process control in an industrial setting.

Recent work by Itakura [4] and by Nishikawa et al. [5] describes a technique for estimating process parameters including deadtimes for systems operating in closed loop. This technique uses time integrals of system variables to a step or impulse input. Proper selection of these integrals (control areas) results in a set of equations solvable for the process parameters. Employing control areas for parameter estimation allows direct identification of process dead time. Additonally, the smooth effects of integration attenuate the effects of process noise on the parameter estimates.

Here, a self-tuning/auto-tuning control technique is described that incorporates both a recursive least squares and a control area auto-tuner for parameter estimation. This novel formulation enjoys the continuous parameter tracking benefits of a recursive least squares algorithm and is able to maintain performance on processes with varying or unknown dead time.

The initial implementation of the two step closed loop parameter estimator is for a model based controller structure [6]. This combination was selected because of the deadtime compensation feature of the model based controller. A comparable formulation exists for a commonly used PID controller a deadtime and initial simulation testing has validated the feasibility of this approach.

The following discussion deals with the two step parameter estimator formulation for a model based controller. This discussion is organized as follows:

First, the data of the model based controller and the recursive least squares parameter estimator are reviewed. Both of these elements are described elsewhere e.g. Lane [7].

Second, the derivation of the control area parameter estimator for the model based controller is described. Simulation results are presented to support the selection of various parameters used by this algorithm and illustrate the strengths and limitations of this technique.
Basic Self-Tuning Controller Description

Initially the two stepceased oop parameter estimator was coupled with a model based controller as a self-tuner/auto-tuner. The mode based controller was selected over a more conventional PID controller for two reasons. The primary reason was that controller's ability to compensate for the effects of process dead time. This allowed the self-tuner/auto-tuner to be applied to a more general class of processes than the class of processes suitable for PID control. A second advantage of the model based controller was the use of a separate parameter for specifying closed loop performance. In contrast, PID parameter selection requires closed loop performance to be factored into the specification of a tuning parameters.

Figure 1 presents a block diagram of the model based controller and (1) defines its transfer function.

\[
\begin{align*}
U(s) & = \frac{\tau_M s + 1}{\tau_M s + 1} \\
e(s) & = \frac{\tau_M s + 1}{\tau_M s + 1} \\
\end{align*}
\]

This particular formulation assumes that the controlled process can be approximated by a first order lag with dead time. The controller is tuned in terms of the observed process gain and dead time. All other tuning parameters the user must specify. The desired time response of the process variable is a change in set point or a disturbance.

The properties of this model based controller are demonstrated in the literature [8] [9] and [10].

The basic self-tuning control system couples a variable forgetting factor version of the least squares parameter estimation with the model based controller. This algorithm addresses many of the practical difficulties associated with the application of self-tuning control. In addition, the parameter estimation algorithm uses a step size. The base developed procedure is based on the model base parameter estimation and the real case of an industrial setting.

The recursive least squares parameter estimator is, given a control input, estimates parameters based on an assumed plant form shown in (2). Further details of this algorithm and the mapping between estimated and controller parameters can be found in Lane [7].

\[
\begin{align*}
y(t) & = a_1 y(t-1) + b_0 u(t-k) + c \\
\end{align*}
\]

where \( y(t) \) current process variable \( y(t-1) \) previous process variable \( u(t-k) \) control output delayed by \( k \) \( a_1, b_0, c \) model parameters

\[
e(t) = y(t) - \hat{y}(t) \\
\]

\[
w(t) = \theta(t) P(t) e(t) \\
\]

\[
\theta(t) = \theta(t-1) + \frac{P(t) e(t)}{1 + w(t)} \\
\]

\[
\eta(t) = \eta(t-1) + e^2(t) \\
\]

where \( \theta(t) \) \( \theta(t-1) \) \( \eta(t) \)

\[
P(t) = \left[ \begin{array}{c} \theta(t) P(t-1) e(t) \end{array} \right] \\
\]

\[
P(t) = \left[ \begin{array}{c} \theta(t) P(t-1) e(t) \end{array} \right] \\
\]

\[
\eta(t) = \eta(t-1) + e^2(t) \\
\]

\[
\eta(t) = \eta(t-1) + e^2(t) \\
\]

\[
\eta(t) = \eta(t-1) + e^2(t) \\
\]

Time horizon for forgetting factor.

The basic self-tuning controller also incorporates an open loop test on the routine for automatic tuning parameters and tailoring the recursive least squares estimator to the actual process. See Lane [11] for further details.

Deadtime/Parameter Estimation Technique

The use of the previously described self-tuning control system is constrained to processes with a known and fixed process delay or to processes where the delay can be measured or inferred online. To extend the applicability of this system to processes with unknown or time varying dead time, a feature based least squares parameter estimation technique was developed. This technique has the advantage of estimating all process parameters including dead time in closed loop. The technique is described in the work of Itakura [4] and Nashimura [5]. The basic idea is that certain parameter measures of the system response to a step setpoint change can
be directly related to the process parameters. Using various control areas involving the process output, one can develop and then solve a set of non-linear algebraic equations in the process parameters.

The following steps describe the procedure used for estimating the deadtime and process parameters of a first order near system using the controller (1):

1. The maximum deviation of the process variable from setpoint is determined over a fixed time horizon along with the average value for the controller output. This information is used to determine when the process has reached steady state and to provide an initial value of the controller output.

2. The setpoint in the controller is increased (decreased) by a prescribed amount to start the estimation procedure.

3. The integrals in (8) and (9) are calculated numerically as the process responds to the setpoint change. Note that the values of both of these integrals converge as the process output approaches the new setpoint. Therefore, these integrals can be calculated over the finite time horizon from 0 to $t_{est}$. Figure 2 illustrates the calculation of these integrals.

4. The process is considered to have achieved steady state conditions when the error between the process variable and the new setpoint is less than twice the maximum deviation measured in step 1 for a fixed amount of time.

5. The final value of the control output is calculated by averaging over a fixed time horizon. This minimizes the effects of process noise on the test.

6. After observing the process response to a step setpoint change (10) through (12) are used to estimate the process parameters. Clearly, $\tau_{est}$ and $\tau_{est}$ cannot be computed directly from these equations. Excellent results, nevertheless, have been obtained by solving the exponential form of (11) from an initial $\tau_{est}$ of 0.0 using a successive substitution technique.

\[ I_1 = \int_0^{t_{est}} m_{SS} m(t) dt \]  \hspace{1cm} (8)

\[ I_2 = \int_0^{t_{est}} \left[ \frac{m_{SS} m(t)}{\Delta SP} \right] e^{a \tau_{est} b} dt \]  \hspace{1cm} (9)

\[ k_{est} = \frac{m_{SS}}{m_{init}} \]  \hspace{1cm} (10)

\[ \tau_{est} = \frac{1}{\alpha} \left[ a \tau_{est} + b \right] \]  \hspace{1cm} (11)

\[ \tau_{est} = \frac{1}{\alpha} \left[ c \right] \]  \hspace{1cm} (12)

where

$\Delta SP$, $I_1$, Control Area Integrals

$m_{SS}$, Final Value, Control Output

$m_{init}$, Initial Value, Control Output

$m(t)$, Control Output at time $t$

$\Delta SP$, Change in Setpoint

$k_{est}$, $\tau_{est}$, $T_{est}$, Estimated Parameters

$k_m$, $\tau_m$, $T_m$, Controller Parameters

$\alpha$, weighting parameter $(\tau_m + T_m)^{1/2}$

and

\[ z - \frac{k_m (\tau_f + \tau_m)}{k_{est}} \left[ \begin{array}{c} \tau_f \cr \tau_f \end{array} \right] - \frac{k_{est} a (\tau_m a + 1)}{k_m (k_{est} - 1) a^{1/2}} \left[ \begin{array}{c} \alpha \tau_m a + 1 + e^{T_m a} \cr k_m (k_{est} - 1) a^{1/2} \end{array} \right] \]

\[ a = \alpha \]

\[ b = \frac{a \tau_m a + 1}{\alpha} \]

\[ c = \frac{k_{est}}{k_m} (\tau_m + 1) \]
Simulation Results

A computer simulation of the model based controller, deadtie/parameter estimator and two generic processes was used to examine the capabilities and limitations of the deadtie/parameter estimator. The simulation on the model based controller was designed to closely parallel the implementation of this controller in Bailey's 90% equipment. This required the approximation of the deadtime within the controller through the use of a finite element buffer and linear interpolation on these buffer values. Two generic processes were used for the simulation testing: a third order lead-lag network with deadtime and a second order system with deadtime specified in terms of static gain, natural frequency, and damping ratio. Both linear processes operated around a bias of 50 and with a gain of 10. Zero mean, Gauss white noise with a specified standard deviation was added to the output of each process. The simulation was designed to run the process at a sample rate of 3 times that of the controller to approximate the behavior of a digital controller operating on an analog process. A supervisory program controlled each simulation run to ensure repeatability and consistency between all tests.

Performances used a first order lag process with deadtime to determine the effect of the ratio between process time constant and deadtime and the model based controllers filter time constant on estimator performance. These tests were also used to validate the robustness of the same gains change used and the ability of the estimator to detect the characteristics of processes with both positive and negative gains. Table 1 summarizes the results of initial testing by showing the worst estimates generated for 5 first order processes with deadtime to the constant ratios ranging from 1/6 to 4. The controller parameters were specified to match those of the actual process with filter time constants ranging between 10 and 90 seconds. The filter time constant, \( \tau_f \) used by the controller that produced the worst case estimate was also tabulated.

Table 1: Estimator performance for first order process with controller/process match and no noise

<table>
<thead>
<tr>
<th>( \tau_p )</th>
<th>( \tau_f )</th>
<th>( \tau_m )</th>
<th>( \tau_{est} )</th>
<th>( \tau_{est} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>60</td>
<td>10</td>
<td>45</td>
<td>59     18</td>
<td>10     41</td>
</tr>
<tr>
<td>30</td>
<td>30</td>
<td>90</td>
<td>32     10</td>
<td>27     98</td>
</tr>
<tr>
<td>30</td>
<td>60</td>
<td>90</td>
<td>34     50</td>
<td>58     07</td>
</tr>
<tr>
<td>30</td>
<td>120</td>
<td>90</td>
<td>34     70</td>
<td>113    79</td>
</tr>
<tr>
<td>30</td>
<td>5</td>
<td>90</td>
<td>31     50</td>
<td>36     61</td>
</tr>
</tbody>
</table>

Smaller values of the filter time constant typically resulted in improved estimator performance down to the point at which controller saturation occurred for the required setpoint change. For small filter time constants the value of \( e_{est} \) in (9) remains large for a greater percentage of the total process response, increasing the amount of information captured by the integral \( I_m \), thus improving estimator accuracy.

The error between the estimated and actual parameters increases as a function of the deadtime value used in the controller results from inaccuracy in the controller deadtime parameter on the controller. When estimator parameters were implemented in the controller, the error was found to have no impact on the overall performance.

Additive output noise with a standard deviation of less than 2% of the magnitude of the setpoint change had minimal impact on estimator performance. The impact of noise, however, cannot be fully understood by analyzing the results of simulation testing alone. It is reasonable to expect that the type of noise present and the magnitude of the noise to the size of the setpoint change used for estimation and the process gain will affect estimator performance in ways that can only be understood through field testing.

A second set of simulations were used to determine the effect of initial parameter estimates on estimator performance. A first order plus deadtime process with \( \tau_m = 30.0 \) seconds and \( T_p = 30.0 \) seconds was used. Controller parameters were varied as follows:

- \( \tau_{est} = 10.0 \) 60.0 seconds
- \( T_m = 10.0 \) 60.0 seconds
- \( \tau_f = 15.0 \) 60.0 seconds

The deadtime/parameter estimator identified both the process time constant and deadtime to within ±1.5 seconds or 5% of actual process values. For cases when either the model time constant or the model deadtime of the controller were significantly larger than the correct value, a small filter time constant led to closed loop instability and no estimates were obtained. The integral measure \( I_m \) did not converge in this situation.

The above simulation studies validated the performance of the deadtime/parameter estimator for the identification of parameters of first order processes. Excellent results were obtained independently of initial controller settings and the presence of moderate levels of process noise. The estimator experienced difficulty only when the controller saturated during the course of the closed loop test or when the initial controller tuning resulted in a marginally stable or unstable closed loop system response.

Further simulation testing was conducted to explore the ability of the deadtime/parameter estimator to approximate high order dynamic systems with a first order process model. Two processes were used for this testing:

- \( G_1(s) = \frac{10s + 1}{(30s + 1)^3} \)
- \( G_2(s) = \frac{10s}{s^2 + 2cw_n + w_n} \)

The deadtime/parameter estimator was tested on \( G_1(s) \) using controller settings:

- \( \tau_m = 30.0 \) 90.0 seconds
- \( T_m = 30.0 \) 90.0 seconds
- \( \tau_f = 30.0 \) 120.0 seconds

For all controller settings giving a stable closed loop response, the estimator approximated the third order process with:

- \( G_{est}(s) = \frac{10s + 1}{(30s + 1)^3} \)
- \( G_{est}(s) = \frac{10s}{s^2 + 2cw_n + w_n} \)
Figure 3 shows the open loop response of the estimated process $G_f(s)$ and compares it with the closed loop response of the process $G_f(s)$ using the mode based controller based on $G_{ies}$ and a filter time constant of 15 seconds.

Similar on testing with the process $G_p(s)$ revealed an inability of the deadtime/parameter estimator to approximate most underdamped systems using a first order model. Nevertheless the estimator did provide suitable approximations for processes with a damping ratio above 0.8. Similarly, the first order model based controller provided stable closed loop response on processes with a damping ratio below this value only with the use of a larger filter time constant. This type of tuning resulted in sluggish closed loop performance.

**Implementation Details**

The deadtime/parameter estimator is designed to detect naturally occurring step setpoint changes and to estimate process parameters when they occur. The estimator can may calculate the average of the process variable over a fixed time horizon using a first order filter. When a step setpoint change is detected, the estimator compares this average with the old setpoint to determine if the process was at steady state conditions. If so, the estimator algorithm is run. Should a second setpoint change occur before completion on the estimator is aborted. For reasons of practicality the estimator will treat ramp setpoint changes of less than 5 second duration as a step setpoint change. Use of a ramp setpoint change will lead to a slight degradation of the deadtime estimate.

In addition, the user may select to have the a go on/off on demand from the operator or on an event or time dependent basis. In these additional modes of operation, the user specifies the duration and magnitude of the setpoint change to be used for the test. Testing the estimator on a go on/off test is the following sequence of events:

1. The estimator waits until it determines that the process is at steady state conditions.
2. The setpoint is changed by the specified amount.
3. Parameters are estimated to characterize the process from observations of the closed loop response. If the estimation is go on/off change the filter time constant in the mode based controller. The increase in the filter time constant will stabilize the process response. Owing the estimator to obtain a valid model of process dynamics. After validation of the parameter estimates, they are made available for automatic or manual updating of the control system.
4. The setpoint is returned to its pre-test value.
5. A second set of parameters are estimated to characterize the response of the process moving in the opposite direction of the first test. The second test helps to identify overall process response so as to diminish effects caused by nonlinearities.
6. The deadtime/parameter estimator outputs the average of the estimates obtained during the two tests. If the filter time constant was changed after the first test, it is returned to its original pre-test value.

The parameter estimates obtained in response to both natural and induced setpoint changes can be used to automatically update the model based controller or can be presented to the operator for review and manual updating.
Many questions still remain to be answered about when the 
deadtime/parameter estimator should be run and how the 
estimator should interact with Bailey's existing self tuning 
controller. Current development efforts are focussing on the 
following topics:

- Development of an algorithm to monitor the stability 
of the closed loop system. This information will 
clarify the adjustment of the filter time constant to 
stabilize the system until new estimates of the process 
parameters are obtained.

- Development of a general measure of process 
performance. This information will 
avoid the estimation of a 
good fit to the process and 
run automatically when the closed loop performance 
degrades below a specified threshold.

- Integration of the deadtime/parameter estimator with 
Bailey's existing self-tuning controller. This will 
provide the continuous parameter tracking of a 
recursively evaluated estimator with an accurate 
dead time estimate supplied from the 
deadtime/parameter estimator.

Conclusion

This paper presents implementations and simulation on study 
details of a new technique for estimating on-process parameters 
including deadtime. It was shown that this technique 

- Is less sensitive to parameter estimates for the 
parameters.

- Is very sensitive to process noise.

- Is able to develop a first order process model that 
adjust estimates higher order dynamics for the purposes of control. 
This does not indicate processes with underdamped open loop responses for which the described controller structure is inadequate.

Acknowledgements

This contribution to this work of M. Buchner and K. 
Loparo of CWRU's Systems Engineering Department are 
acknowledged. They are responsible for the 
development of the deadtime estimation technique employed 
and for preliminary derivations and studies on testing of this 
algorithm in conjunction with the model-based controller.

References

1. R. Chen, "Identification of Time Delays in Linear 

Adaptive Control of Processes with a Slowly Varying 
Dead Time," Automatica, 17(1), 1981

3. De Souza E., Goodwin G., Mayne D., and 
Palazanawam M., "A New Adaptive Control Algorithm for 
Linear Systems Having Unknown Time Delay," 
Automatica, 24(3), 1988

4. H. Itakura, "Parameter Identification in a Process 
Control System," IEEE Transactions on Automatic 
Control, AC 31(12), 1986

5. Y. Nishikawa et al., "A Method for Auto-Tuning of 
PID Control Parameters," Automatica, 20(3), 1984

6. T. Matsko, "Internal Model Control for Chemical 
Recovery," Chemical Engineering Progress, 1985

7. J. Lane, "Description of a Modular Self-Tuning 
Control System," Proceedings of the American Control 
Conference, 1986

8. J. Parsh, "The Use of Model Uncertainty in Control 
System Design with Application to a Laboratory Pica 

9. Popel L. Matsko, J. and Brosow C., "Coordinated 
Control," Chemical Process Control CPC III, 1986

10. Parsh, J., and Brosow C., "Inferential Control 
Applications," Automatica, 21(5), 1985

11. J. Lane, "Self-Tuning Control," Power Controls Co 
Application Guide, AG-0000-953-01, 1987