IFAC Conference on Advances in PID Control
PID'12
Brescia (Italy), March 28-30, 2012

Intelligent PID Product Design

Willy K. Wojsznis*, Terrence L. Blevins*, John Caldwell*, Peter W. Wojsznis*, Mark J. Nixon*

*Emerson Process Management, 12301 Research Blvd., Austin, TX 78759
USA (Tel: 512-418-7475; e-mail: willy.wojsznis@emerson.com).

Abstract: This paper outlines intelligent PID design for DCS. The design includes a PID algorithm with diverse standard options and algorithm extensions for wireless/event-driven control and for surge control. The core of the PID intelligence is adaptive process modeling based on model switching and parameter interpolation. The developed process model is applied to loop tuning, adaptive control, loop performance evaluation and valve diagnostics. A user-friendly interface provides insight into a loop's current state and history events. The interface also provides advice about how to improve loop performance.

Keywords: Intelligent control, Adaptive algorithms, Feedback control, Feedforward control, First order systems, Performance monitoring, Diagnostic programs, User interfaces

---

1. INTRODUCTION

The label *intelligent* is attached to many control products. Some products are named *intelligent* because they perform self-diagnostics or use more complex algorithms. It seems the name *intelligent* is justified for many products when a collection of simple features improve product functionality and make it easy to use. In this paper we adhere to the more demanding definition of *intelligent.* An intelligent control system has the ability to learn about processes, disturbances and operating conditions – Åström et al. (1992). The key features for such an intelligent control system are adaptive process modelling, adaptive control and enhanced overall functionality. Adaptive control for industrial applications started from non-model based PID tuning, like the pattern recognition technique – Åström et al. (2006a), where PID parameters are adjusted by an automated procedure used by an experienced process engineer. Another technique applied for auto-tuning in industry is the relay oscillation technique – Åström et al. (2006b). The controller tuning applies Ziegler-Nichols rules or its variations and is characterized by its simplicity. A modification of this technique made it possible to identify a first order plus time delay process model – Blevins et al. (2003). This modified technique has been applied to auto-tuning in the PID controller design described in this paper.

The adaptive process identification technique with model switching was found to be robust and reliable in various applications - Morse et al. (1994), Narendra et al. (1997). Reducing the number of models and applying parameter interpolation to improve accuracy simplified the technique and made it suitable for the industrial implementation – Wojsznis et al. (2002), (2003). Several variations of the model switching technique have been compared in controlling simulated pH – Böling et al. (2005), confirming the advantages of the model switching adaptation. The modelling accuracy of the switching technique has been further improved by running the the algorithm recursively with the same data – Wojsznis et al. (2003). The enhanced algorithm is a main adaptive technology for the design.

The design implements performance monitoring based on the concept by Harris (1989), Desborough et al. (1992). The concept has been extended on the tuning index, which indicates potential variability reduction due to the improved tuning. This type of tuning index for PI controllers has been explored by Ko et al. (1998).

Valve diagnostics for control loops has been the focus of research by Horch (1999) and Choudhury et al. (2004). They present practical results in detecting valve stiction and in providing qualitative assessment in a graphical form. In the discussed PID product, the process model obtained from the adaptation is used for valve diagnostics. This novel approach allows automatic identification of the valve dead band and resolution.

The PID controller has an option for wireless operation with event-driven mode of operation for optimizing performance and saving energy.

The PID control design provides an automatically configured user interface for setting loop operation and observing the results of auto-tuning, adaptive tuning and performance evaluation. The paper is structured as follows – section 2 outlines the PID design, section 3 - adaptive tuning and control, section 4 - loop performance evaluation and fault detection. Section 5 presents valve diagnostics concepts followed by conclusions and acknowledgments.

2. PID DESIGN OUTLINE

2.1 PID Function Block

The PID algorithms are implemented in the fieldbus function block, which combines all of the necessary logic to perform analog input channel processing, proportional-integral-derivative control with the option for nonlinear control (including error-squared and notched gain), and analog output channel processing.

The PID function block supports mode control, signal scaling and limiting, feedforward control, override tracking, alarm limit detection, and signal status propagation. To support testing, a simulation mode is available.

Two PID equation forms are supported in the block: standard and series. Both forms include external reset and feedforward control.

The PID block provides enhanced control for applications in the following areas:

- nonlinear (notch gain) control
- exception reporting execution
- enhanced saturation recovery

Figure 1 illustrates how the *nonlinear* tuning parameters are used to establish notched gain - KNL.

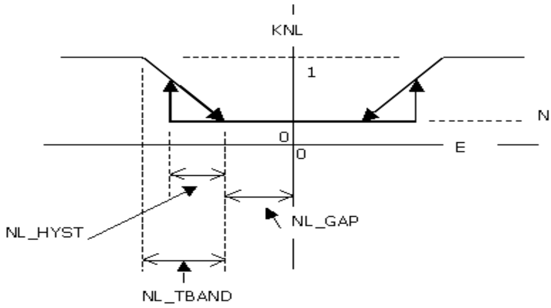


Fig. 1. Notch and notch gain (KNL) parameters

The PID product uses a positive feedback network to create the reset contribution to the PID output. For the standard mode of operation, the time constant of the filter included in the positive feedback network is the reset time. For the *exception reporting execution* (wireless application) the filter is restructured to provide integral action to match the process response in the elapsed time as in (1). The derivative calculation is modified to compute a rate of change over the elapsed time from the last update (2). Reset and rate action are only computed when there is a new value.

$$F_N = F_{N-1} + \left(O_{N-1} - F_{N-1}\right) * \left(1 - e^{\frac{-\Delta T}{T_{Reset}}}\right) \qquad (1)$$

where F_N - new filter output

F_{N-1}- filter output for the last execution

O_{N-1}- controller output for the last execution

ΔT - elapsed time since a new value was communicated

$$O_D = K_D \bullet \frac{e_N - e_{N-1}}{\Delta T} \qquad (2)$$

where e_N - current error

e_{N-1} - last error

O_D - controller derivative term

ΔT - elapsed time since a new value was communicated

Performance of the algorithm has been reported in Kaltiokallio et al. (2010).

Due to the use of the positive feedback network for the reset component, each of the elements in the forward path must contain a filter component $1/T_i s + 1$. However, when a process is operating using a *saturated condition*, the PID is at its output limits. Under these conditions the filter in the forward path limits response to process conditions that require the PID to move from its output limits. The point at which the valve will start to open as the PV approaches setpoint depends on the PV rate of change and the magnitude of the error. A better response to major upsets can be achieved by reducing the filtering that is applied in the forward path calculation for this process saturation condition. Such reductions are possible if the process measurement is relatively noise free. Thus, there is merit in allowing the user to select the amount of filter applied in the forward path when the PID output is limited.

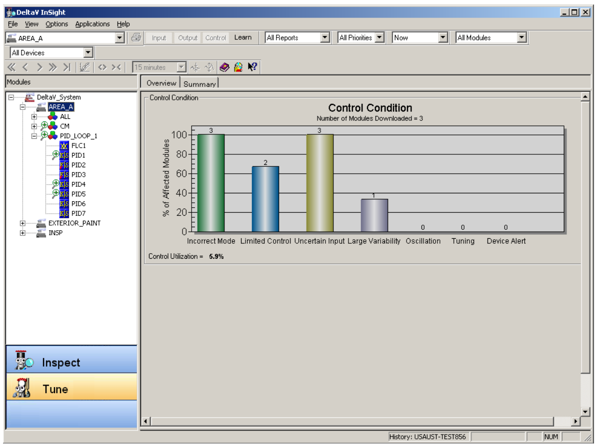
The extended PID algorithms constitute a solid base for implementing advanced functionality: auto-tuning, adaptive tuning and control, loop performance evaluation and fault detection.

The PID function block operates in the DCS controller (*the DCS is a DeltaV System*) and uses all the standard DCS tools for graphical control design and debugging (Control Studio) and on-line operation monitoring and adjusting (dynamos, faceplates and detailed displays).

2.2 PID application

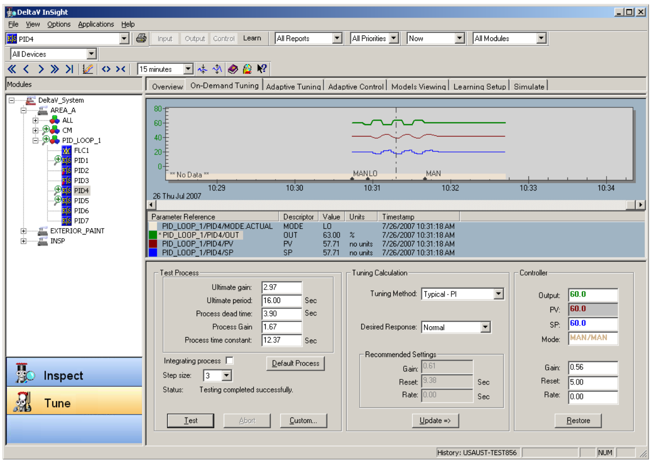
The PID application (*Insight*) is installed on the workstation and provides support for advanced functionality. The entry application screen provides a summary overview of the control loop’s operation and statistics on incorrect mode, limited control, large variability, loop oscillation and incorrect tuning – Figure 2. Further details for every loop through various tabs. The *Tune* tab provides a complete interface for on-demand tuning, adaptive tuning, adaptive control, and models viewing for learning setup and for testing loop operation in simulation – Figure 3. The simulation allows the user to explore loop performance with new tuning parameters, adjust parameters for a desired performance or to design the loop in a unique graphical way using the robustness plot in the gain-phase margin plane – Figure 4.

The robustness plot presents an area of acceptable gain and phase margin for a specific process model.



Control Utilization = 5.9%

Fig. 2. Summary screen of the advanced PID application.



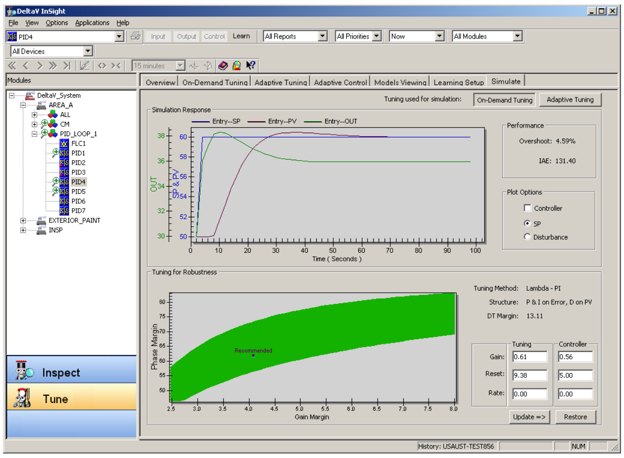
Parameter Reference	Descriptor	Value	Units	Timestamp
PID_LOOP_1/PID4/MODE.ACTUAL	MODE	LO		7/26/2007 10:31:18 AM
* PID_LOOP_1/PID4/OUT	OUT	63.00	%	7/26/2007 10:31:18 AM
PID_LOOP_1/PID4/PV	PV	57.71	no units	7/26/2007 10:31:18 AM
PID_LOOP_1/PID4/SP	SP	57.71	no units	7/26/2007 10:31:18 AM

Test Process
Ultimate gain: 2.97
Ultimate period: 16.00 Sec
Process dead time: 3.90 Sec
Process Gain 1.67
Process time constant: 12.37 Sec
Integrating process ☐
Step size: 3
Status: Testing completed successfully.

Tuning Calculation
Tuning Method: Typical - PI
Desired Response: Normal
Recommended Settings
Gain: 0.61
Reset: 9.38 Sec
Rate: 0.00 Sec

Controller
Output: 60.0
PV: 60.0
SP: 60.0
Mode: MAN/MAN
Gain: 0.56
Reset: 5.00
Rate: 0.00

Fig. 3. Auto-tuning and adaptive tuning screen.



Performance
Overshoot: 4.59%
IAE: 131.40

Plot Options
☐ Controller
● SP
○ Disturbance

Tuning Method: Lambda - PI
Structure: P & I on Error, D on PV
DT Margin: 13.11

	Tuning	Controller
Gain:	0.61	0.56
Reset:	9.38	5.00
Rate:	0.00	0.00

Figure 4. Loop simulation screen with gain-phase margin robustness plot.

3. ADAPTIVE MODELING, TUNING AND CONTROL

Three approaches to loop tuning are implemented.

On-Demand Tuning - Uses an on-demand test of the process to automatically provide tuning recommendations. On-demand tuning is available for PID and Fuzzy Logic Control (FLC) blocks. Tuning recommendations are available on-demand by initiating automatic testing of the process. When testing is requested using on-demand tuning, the PID or FLC block's actual mode switches to Local Override (LO). Once in LO mode, the operation of the loop's primary control algorithm is suspended and the controller resident relay control adjusts the control block output (OUT). The original algorithm has been enhanced for identifying process dead time and time constant in addition to the ultimate gain and ultimate period. The controller settings are then computed using modified Ziegler-Nichols, Lambda, or Internal Model Control tuning rules.

Adaptive Tuning - Uses normal operator changes in setpoint or output to identify process models and provide tuning recommendations.

Adaptive Control and Model Scheduling **-** Includes all of the Adaptive Tuning capabilities plus the ability to create models in up to 5 regions and to automatically change control loop tuning. The model quality is validated by taking into account the most recent adaptation and the last adaptation. A high quality model and the expected control performance with the recommended tuning are used as criteria for switching to adaptive control.

Relay-based auto-tuning has been presented in many publications Åström et al. (2005b), Blevins et al., (2003), therefore in this paper we focus on the adaptive modeling and control.

The adaptive model identification uses switching strategy with parameter interpolation and model re-centering. This approach makes it possible to dramatically reduce the number of models used for adaptation. The salient features of the technology include: shorter adaptation time, complete process model identification, and reduction in required process excitation.

The structure of the PID with model parameter interpolation is shown in Figure 5.

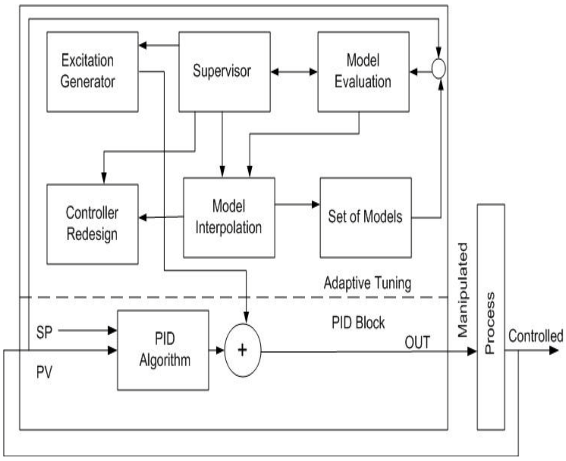


Fig. 5. Adaptive Feedback Control Design.

The process identification is based on the evaluation of multiple sets of models. Each model consists of three parameters (p=3, gain, dead time, lag). Assigning n values for every parameter, the model set has $M = n^3$ models.

The Adaptive Control operates in the following way:

The supervisor detects changes in the process output (i.e. the controlled variable (y-PV)), the setpoint or the manipulated process input (u-OUT).

If any changes exceed a minimum level, model evaluation starts. This involves:

- model initialization and adjustment of the model's output with the current process output
- model incremental update based on the changes of the manipulated input to the process
- computing for every model i squared error

$$e_i^2(k) = \left(y(k) - \hat{y}_i(k)\right)^2 \text{ for each scan } k \text{, where:}$$

$y(k)$ - the process output at the time k

$\hat{y}_i(k)$ - i-th model output at the time k

The squared error is assigned for every parameter value of the model $\boldsymbol{i}$ if the parameter value is used in the evaluated model. A zero is assigned to any parameter value that is not part of the evaluated model. Next, the model $\boldsymbol{i+1}$ is evaluated. Calculated squared error is assigned for every parameter value and previously assigned squared error for every parameter value is updated. The model's evaluation continues until all models are evaluated. As a result of the evaluation, every parameter value is assigned a sum of squared errors from all models in which this specific parameter value has been used. In the one scan *k* therefore, every parameter type *p* with value *v* has assigned squared error $SE_p^v(k)$.

$p=1,2,\ldots,P \qquad v=1,2,\ldots,V$

$$SE_p^v(k) = \sum_{i=1}^{M} \gamma_p^v e_i^2(k) \tag{3}$$

where:

M – the number of models

$\gamma_p^v = 1$ if parameter *p* with value *v* is used in the model, otherwise $\gamma_p^v = 0$. Model evaluation is repeated in the scan *k+1* and the sum of the squared errors for every parameter value is added to the sum of the appropriate parameter value accumulated in the previous scans. The adaptation cycle continues through a declared number of scans (1 to N), or until there is enough excitation on the inputs. As a result of this procedure, every parameter type *p* with value *v* has an assigned sum of squared errors SSE_p^v over a period of evaluation:

$$SSE_p^v = \sum_{k=1}^{N} SE_p^v(k) \tag{4}$$

At the end of adaptation cycle, the inverses of SSE_p^v are calculated:

$$F_p^v = {}^{1}\!\big/_{SSE_p^v} \tag{5}$$

An adaptive parameter value a_p for the parameter *p* is calculated as a weighted average of all values of this parameter $v_p(v) \qquad v = 1,2,\ldots,V$.

$$a_p = v_p(1) f_p^1 + v_p(2) f_p^2 + \ldots + v_p(V) f_p^V$$

where:

$$f_p^v = F_p^v \left(\sum_{1}^{V} F_p^v \right)^{-1} \tag{6}$$

Calculated parameters define a new model set with center parameter values a_p and the range of parameter changes defined as $\pm\Delta\% a_p$, $p=1,2,\ldots,P$. Within that range, two parameters at a minimum should be defined. As soon as a model has been updated, controller tuning takes place based on updated a_p model parameters.

In the PID product a first order plus dead time process model is used (in a discrete form).

$$\Delta y(k) = a\Delta y(k-1) + b\Delta u(k-1-L) \tag{7}$$

$$a = e^{-\frac{h}{\tau}} \qquad b = k\left(1 - e^{-\frac{h}{\tau}}\right) = k(1-a) \tag{8}$$

where:

h = loop scan period

L – process dead time, $L=hk$.

Δu - process input incremental change since adaptation started

Δy - model output incremental change

τ = model time constant

With three values for every parameter of the model, there are $3^3 = 27$ models for evaluation.

Improved convergence and reduction in the number of models is achieved by the following amendments to the basic algorithm:

- Performing parameter adaptation sequentially, one parameter at a time. In this way, the number of model combinations for the first order plus dead time model has been reduced to 3*3=9.
- Performing adaptation for only two parameter values with minimal errors.
- Using the original data set and performing adaptation iteratively by running the algorithm several times.

In a sequential procedure in which one parameter is updated over a calculation cycle, the updating is performed in the sequence process gain, dead time and time constant – Figure 6.

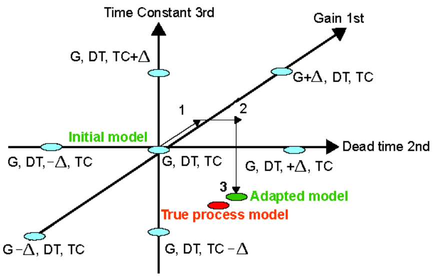


Fig.6. Sequence of parameters adaptation and interpolation

Figure 7 illustrates model adaptation, showing how process output is aligned with three models outputs.

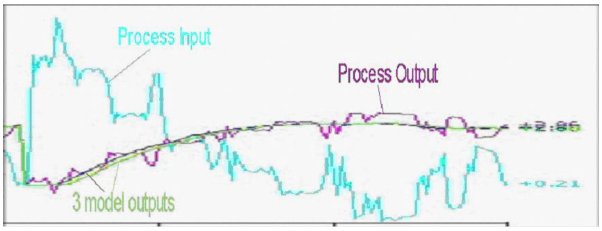
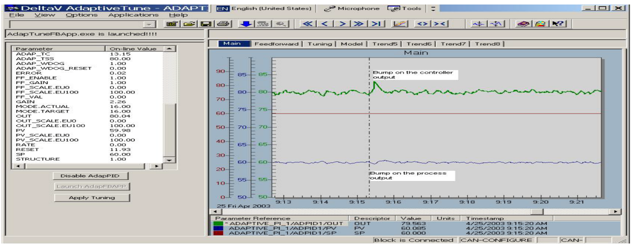


Fig. 7. Adaptive models validation plots

After model adaptation completes, controller redesign begins using the first order plus dead time process model. Any tuning rules, typically Lambda or IMC, can be applied. If there are infrequent changes in the manipulated input, external excitations can be automatically injected into the manipulated input in Automatic modes. PID output pulse of the amplitude 3-7% and duration of several scans is normally enough for a process model identification – Figure 8.



AdapTuneFBApp.exe is launched!!!!

Parameter	On-line Value
ADAP_TC	13.15
ADAP_TSS	80.00
ADAP_WDOG	1.00
ADAP_WDOG_RESET	0.00
ERROR	0.02
FF_ENABLE	1.00
FF_GAIN	1.00
FF_SCALE.EU0	0.00
FF_SCALE.EU100	100.00
FF_VAL	0.00
GAIN	2.26
MODE.ACTUAL	16.00
MODE.TARGET	16.00
OUT	80.04
OUT_SCALE.EU0	0.00
OUT_SCALE.EU100	100.00
PV	59.98
PV_SCALE.EU0	0.00
PV_SCALE.EU100	100.00
RATE	0.00
RESET	11.93
SP	60.00
STRUCTURE	1.00

Parameter Reference	Descriptor	Value	Units	Timestamp
🟩 *ADAPTIVE_PI_1/ADPID1/OUT	OUT	79.563		4/25/2003 9:15:20 AM
🟦 ADAPTIVE_PI_1/ADPID1/PV	PV	60.085		4/25/2003 9:15:20 AM
🟥 ADAPTIVE_PI_1/ADPID1/SP	SP	60.000		4/25/2003 9:15:20 AM

Fig. 8. An example of minimal excitation on the process input (upper plot) and output required for the process model identification.

4. PERFORMANCE MONITORING AND FAULT DETECTION

The ability to inspect control and measurement loops quickly is of primary importance. The PID design provides advanced process monitoring that allows under-performing loops and malfunctioning field devices to be identified instantly. In providing this advanced capability, it takes full advantage of the fieldbus block architecture. A *Bad*, *Uncertain*, or *Limited* measurement, downstream *limitations* in control, and *incorrect* mode of operation are automatically determined based on block mode and the status of block parameters. The user interface provides indices that quantify loop utilization, measurements with abnormal status, limitations in control action, process variability, and availability of recommended tuning. Other parameters used within the block are standard deviation, variability and tuning index.

The existence of an abnormal condition is determined within the controller based on function block parameters in the controller (such as measurement status, back calculation input status, standard deviation, tuning index, and mode). The state of these monitored conditions is automatically reported to the DCS server on an exception basis. Loop performance is indicated by the variability and tuning indexes.

Variability index is a measure of the quality of control, while *the tuning index* indicates the potential for improving control performance by updating controller tuning according to the identified process model and the selected or default tuning rule.

To support the performance calculations performed in the function blocks, capability and total standard deviations are calculated every scan ($\sigma_{cap}, \sigma_{tot}$).

The total standard deviation and the capability standard deviation is calculated using a moving time window as in Shunta (1995):

$$\sigma_{cap}(K) = \sigma_{cap}(K-1) + f\left(MR/_{1.128} - \sigma_{cap}(K-1) \right) \quad (9)$$

where:

$\overline{MR} = 1/_{(K-1)} \sum_{2}^{K} |y(k) - y(k-1)|$ - average moving range

K – the current window, K-1 – the previous window

The *variability index* (*VI*) in % is computed then as:

$$VI = 100\left(1 - (\sigma_{mvc} + s)/_{(\sigma_{tot} + s)} \right) \quad (10)$$

where

s is the sensitivity factor that makes calculations stable. The default value is 0.1% of the variable scale.

σ_{tot} is the actual measured standard deviation.

σ_{mvc} is the standard deviation that can be achieved with minimum variance feedback control defined as:

$$\sigma_{mvc} = \sqrt{2 - \sigma_{cap}/_{\sigma_{tot}}} \quad (11)$$

The *tuning index* is based on the variability difference estimate for the current controller tuning (*PID1*) and the desired controller tuning (*PID2*).

$$\Delta\sigma_{res}^2 = \sigma_{PID1}^2 - \sigma_{PID2}^2 \quad (12)$$

The *tuning index TI* indicates the potential for improving control performance and is defined as the ratio of the potential PID variability reduction to the actual PID variability:

$$TI = \frac{\Delta\sigma_{res}^2}{\sigma_{PID1}^2} \quad (13)$$

5. VALVE DIAGNOSTICS

It is commonly recognized that the undesirable behaviour of control valves is the biggest single contributor to poor control loop performance and the destabilization of process operation. It is normal practice to perform loop special testing in manual mode and use the test results for calculating valve resolution and dead band – Gerry et al. (2001). There are several effective techniques for complete valve diagnostics when internal valve parameters, including positioner pressure are available – Grumpstrup et al. (2001). The loop diagnostics are extremely important for the adaptive loops. Loop adaptation may bring instability caused not by the failed adaptation but by the sticky valve or measurement failure. On the other hand, manual or semi-manual diagnostic procedures as cited above are not adequate for the adaptive loop working with no supervision. The on-line techniques by Horch (1999) and Choudhury et al. (2004) provide qualitative valve evaluation. For the PID product, valve diagnostics techniques has been developed that do not assume access to any valve parameters. Availability of the valve stem position improves the diagnostics, though this parameter is optional. The approach uses the process model gain $\boldsymbol{K}_p$ and is the best suited for the adaptive control loops or automatically tuned loops where process gain is known.

The valve dead band and resolution is reflected by a characteristic loop oscillation – Figure 9. After calculating oscillation amplitudes on the controller input $A(pv)$ and output $A(out)$ valve hysteresis is defined directly as:

$$h = 2A(out) \quad (14)$$

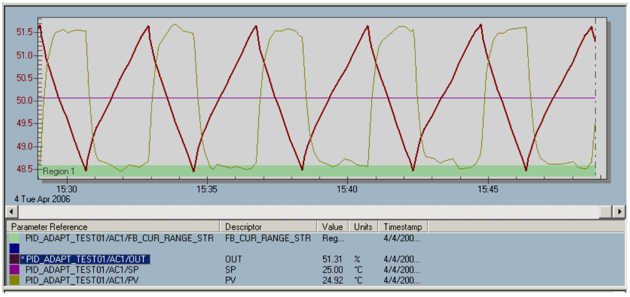
The valve resolution r defines effective valve stem movement during valve induced loop oscillations, which applied to the process with gain K produce oscillation on the process output with the amplitude $A(pv)$

$$2Ampl(PV) = Kr \quad (15)$$

$$r = {2Ampl(PV)}\Big/{K} \tag{16}$$

Dead band is calculated as

$$b = h - r \tag{17}$$



Parameter Reference	Descriptor	Value	Units	Timestamp
🟩 PID_ADAPT_TEST01/AC1/FB_CUR_RANGE_STR	FB_CUR_RANGE_STR	Reg...		4/4/200...
🟦				
*PID_ADAPT_TEST01/AC1/OUT	OUT	51.31	%	4/4/200...
PID_ADAPT_TEST01/AC1/SP	SP	25.00	°C	4/4/200...
PID_ADAPT_TEST01/AC1/PV	PV	24.92	°C	4/4/200...

Figure 9. A loop oscillation caused by the valve dead band 2% and the resolution 1% , noise level is 0.25%.

The technique can be refined by using the back calculation signal from the valve. Other than imperfect process gain sources of error include difficulty with separation of the loop oscillation caused by the valve from the oscillation caused by poor tuning, external disturbances or set point changes. The oscillation shape factor is applied for identifying oscillations caused by the valve. Some of the lab test results are in the table 1.

Table 1 – Valve stickness (resolution) and backlash diagnostics: real value /diagnosed value

Dead band %	1/1	0.4/ 0.4	0.8/ 0.93	3.0/ 3.6	2/ 2.1	2/2	2/2	2.5/ 2.1	3/ 2.3
Resolution %	.5/ .49	.1/ .098	0.1/ 0.09	0.1/ 0.1	.2/ .23	.5/ .49	1/ .99	1/.9	1/ .96

The technique will be included into product after full field test trial.

6. CONCLUSIONS

The intelligent PID design described here has been widely accepted by the users. The main factors that contributed to the acceptance are the robust process model identification technique and a user interface that provides full insight into control loop operation, control performance, faults and tuning recommendations.

Acknowledgments

Our communication with professors Karl Åström, Dale Seborg and Thomas Edgar greatly improved the product concepts and design. The final shape of the product and its quality is the result of contributions from many control software developers including Dennis Stevenson, John Gudaz, Peter Wojsznis, Mike Ott, Yan Zhang and Ron Ottenbacher.

REFERENCES

Åström, K.J. and McAvoy, Th. J. (1992). Intelligent control. J. Proc. Cont., Vol. 2, No. 3. 115-127.

Åström, K.J. and Hägglund, T. (2006a) *Advanced PID Control*, pp. 316-318. ISA, USA.

Åström, K.J. and Hägglund, T. (2006b) *Advanced PID Control*, pp. 321-322. ISA, USA.

Blevins, T.L., McMillan, G.K., Wojsznis, W. K. and Brown, M. W. (2003) *Advanced Control Unleashed*, pp. 210-218. ISA, USA.

Böling, J.M., Seborg , D.E. and Hespanha, J. P. (2005) Multi-model control of a simulated pH neutralization process, *IFAC World Congress,* Prague, Czech Republic.

Choudhury, M., Shah, S.L. and Thornhill, N.F. (2004) Diagnosis of poor control-loop performance using higher-order statistics, *Automatica,* 40, 1719-1728.

Desborough, L. and Harris, T.J. (1992) Performance assessment measures for univariate feedback control, *Can. J. Chem. Eng.,* 70, 1186-1992.

Gerry, J., Ruel, M. (2001) How to measure and combat valve stiction on line, *ISA Technical Conference*, Houston, USA.

Grumstrup, B.F., Junk, K.W., Snowberger, J.L. and Taylor, E.R. (2001) A method and apparatus for deterministically obtaining measurements, *US Patent 6,192,321.*

Harris, T.J. (1989) Assessment of control loop performance, *Can. J. Chem. Eng.,* 67(10), 856-861.

Horch, A. (1999) A simple method for detecting of stiction in control valves, *Control Engineering Practice,* 7, 1221-1231.

Kaltiokallio, O., Eriksson, L.M. and Bocca, M. (2010) On the performance of the PIDPLUS controller in wireless control systems, *18th Mediterranean Conference on Control and Automation,* Marrakech, Morroco.

Ko, B.S. and Edgar, T.F. (1998) Assessment of achievable PI control performance for linear processes with dead time, *ACC,* 7, USA.

Morse, A.S., Pait, F.M. and Weller, S.R. (1994) Logic-based switching strategies for self-adjusting control, *33rd IEEE Conference on Decision and Control,* Workshop no. 5, Lake Buena Vista, USA.

Narendra, K. S. and Balakrishnan, J. (1997) Adaptive Control Using Multiple Models, *IEEE Transactions on Automatic Control,* Vol. 42, No.2, 177-187.

Shunta, J. (1995) *Achieving World Class Manufacturing Through Process Control*, pp. 28-33, Prentice Hall, USA.

Wojsznis, W. K. and Blevins, T. L. (2002) Evaluating PID adaptive techniques for industrial implementation, *Proc. of ACC,* Anchorage, Alaska, USA, pp. 1151-1155.

Wojsznis, W. K., Blevins, T. L. and Wojsznis P.W. (2003) Adaptive feedback/feedforward PID controller, *ISA Technical Conference,* P137, Houston, USA.