2.29 Self-Tuning Controllers

P. E. WELLSTEAD (1995) **B. G. LIPTÁK, S. RENGANATHAN** (2005)

Types of Products on the Market:	A. Stand-alone hardware mostly for temperature control B. Stand-alone hardware for tuning all process loops C. Offered as part of DCS supplier's package D. Self-tuning software package
Partial List of Suppliers (Product Names):	ABB Process Automation, U.S. (MC 5000) (B) (www.abb.com/us/instrumentation) Asea Brown Boveri, Sweden (Novatune) (B, C) (www.abb.com) CAL Controls, U.S. (9900 Autotune) (A) (www.cal-controls.com) Control Techniques, U.K. ("Expert") (B) (www.controltech.co.uk) Eagle Controls, U.K. (Eagle Mini 948) (A) (www.applegate.co.uk) Emerson Process Measurement, U.S. (RS3) (B) (www.emersonprocess.com) Eurotherm, U.S. (815, 8181) (A) (www.eurotherm.com) FGH Controls, U.K. (5900, 556) (A) (www.fgh.co.uk) Fuji Electric, Japan (CC-S) (B) (www.fujielectric.co.jp) Honeywell, U.S. (UDC 6000, E-Max, D-Max) (B) (www.iac.honeywell.com) Invensys, Foxboro, U.S. ("Exact" 761, 760) (B) (www.foxboro.com) Jumo, U.S. (Dicon 5) (A) (www.jumousa.com) Philips, U.K. (KS 4290) (A) (www.philips.co.uk) Self-Tuning Friend, U.K. (D) (www.csc.umist.ac.uk) Siemens, Germany (Sipart dr) (B) (www.sea.siemens.com) Toshiba, Japan (EC300, 2150) (B) (www.tic.toshiba.com) Yokogawa, U.S. (UP25, YEW80) (A) (www.yokogawa.com/us)

INTRODUCTION

In Section 2.18, the methods of process modeling using artificial neural networks (ANN) were described. Some of the concepts used in designing self-tuning controllers use the concepts of the ANN family (Figure 2.29a). Before reading this section, it is advisable to become familiar not only with ANN, but also with the basics of PID controllers, their tuning, and the related subjects of model-based and adaptive controls, which are all covered in this section.

The selection of controller settings that will provide optimum performance is called controller tuning. If the controller is manually tuned and the dynamics of the controlled process change, the tuning has to be done again. If on the other hand, the tuning is done automatically periodically, then it is known as self-tuning. Some DCS and other control system suppliers provide the means to allow the operator or a timer to initiate controller tuning.

Self-tuning controllers are capable of automatically readjusting the controller tuning settings. They are also referred to as auto-tuning controllers and can be stand-alone products, integral parts of distributed computer control systems (DCS), or software packages. The market is dominated by stand-alone products, most of which are able to communicate with other systems or other controllers.

The standard operating principles include self-tuning regulators (STR) and self-tuning temperature controllers using feature-extraction–type tuning methods, which are obtained from step response information taken at startup or when load changes occur. General-purpose stand-alone units can tune by pattern recognition methods (e.g., Foxboro Exact) or by model-based methods (e.g., ABB Novatune).

EVOLUTION

The first self-tuning controllers came from Europe in the early 1970s. The original methods used optimal regulation[1,2] with optimal control,[3] followed by pole-placement[4] somewhat later. The subsequent development involved new algorithms,[5,6] stability theory,[7] and the industrial development of commercial self-tuning controllers. A comprehensive treatment of algorithms

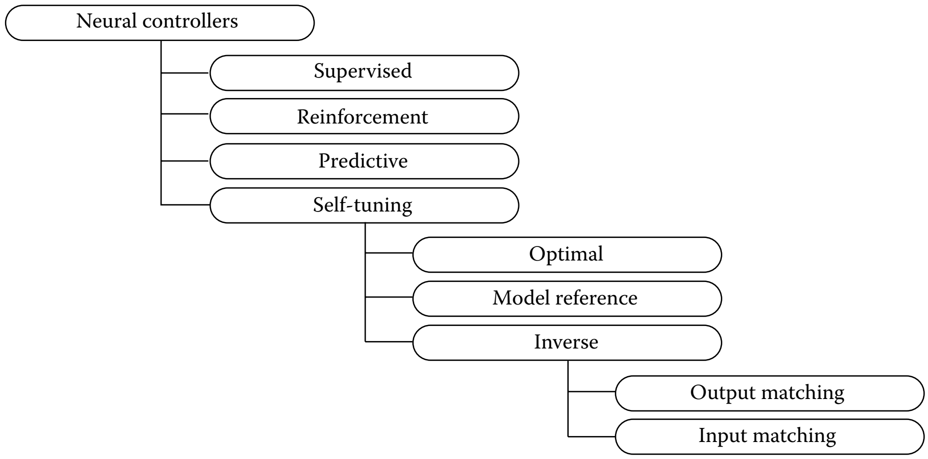


FIG. 2.29a
A hierarchical classification of ANN-based control algorithms. Self-tuning controllers belong to the family of neural controllers, which are distinguished from each other by the way they use the error signal to train the ANN.

for self-tuning control is provided in Reference 8, while nonlinear and adaptive controls are covered in another section of this chapter.

The general principle of operation of the family of self-adaptation is described in Figure 2.29b. This approach is applied when the cause of changes in the control loop response is either unknown or unmeasurable and the adaptation therefore must be based on the response of the loop itself. Several self-adaptive systems are described in the paragraphs that follow, including the self-tuning regulator, the model reference controller, and the pattern-recognizing adaptive controller.

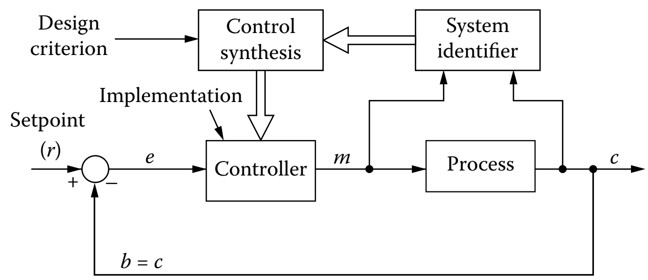


FIG. 2.29c
The main components of a self-tuning regulator.

Self-Tuning Regulator (STR)

The main components of a self-tuning regulator are shown in Figure 2.29c. They are as follows:

1. *A system identifier* — This element consists of a process parameter estimation algorithm, which estimates the parameters of the process.
2. *A controller synthesizer* — This synthesizer calculates the new controller parameters as a function of the estimated process parameters specified by the control objective function.
3. *A controller implementation block* — This is the controller whose parameters are updated at periodic intervals by the controller parameter calculator.

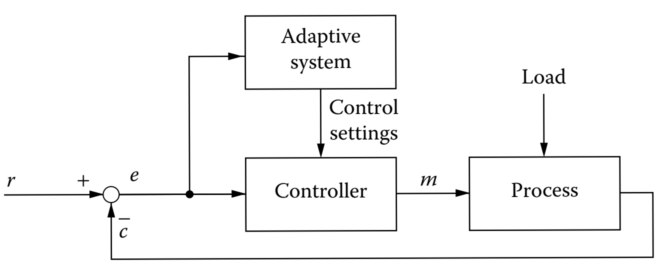


FIG. 2.29b
In self-adaptation, the correction is based on the response of the loop itself. One can view a self-adaptive system as a control loop around a control loop.

The STRs are distinguished on the basis of their identifiers and synthesizers. Popular varieties include the minimum-variance, generalized minimum-variance, detuned minimum-variance, dead-beat, and generalized pole-placement controllers.

As shown in Figure 2.29c, the system identifier determines the response of the controlled variable (c) to a change in the manipulated variable (m). Model-based self-tuning algorithms use some method based on recursive estimators. Many commonly used industrial products, however, use pattern recognition or expert/fuzzy logic methods to extract key dynamic response features from a transient excursion in the system dynamics.

The transient excursion can be deliberately introduced by the controller, or preferably, it is the start-up transient or a normally occurring transient of the process. The desired tuning settings for PID, optimal, or other types of control are determined by the control synthesis block, which is also called the controller parameter calculator. The control synthesizer can be simple or sophisticated depending on the rules used.

MODEL-BASED METHODS

There are two classes of model-based self-tuners: optimal and response specification types. Both of these controllers are composed of a reference model, which specifies the desired performance; an adjustable controller, which sets the manipulated so as to bring the process performance as close as possible to that of the reference model; and an adaptation mechanism.

Figure 2.29d schematically shows the organization of the reference model controller.

Optimal Self-Tuning Optimal self-tuning algorithms use optimal regulation theory as their design rule in the controller parameter calculator. The design methods used are minimum variance (MV), generalized minimum variance (GMV), and generalized predictive control (GPC). The MV and GMV methods are, as the names imply, aimed at minimizing the mean square deviation of a process variable from its set point. The minimum variance methods have deficiencies when the process time delay is unknown or variable because they can, if unprotected, become unstable.

The related optimal method GPC was developed to overcome this deficiency by using prediction horizon ideas that occur in DMC (dynamic matrix control).[9] These methods are used in turn-key applications by research and consulting companies. Few industrial controllers use them because of the sophisticated knowhow required, although the ABB Novatune is a widely installed and respected product that uses optimal methods.

Transient Response Self-Tuner Another class of model-based self-tuners uses the desired closed-loop frequency or the transient response characteristics of the loop as the basis for operating the self-tuning algorithm. The accepted technique for this is pole-placement (PP) or pole-assignment (PA). This procedure asks the designer to select the desired closed-loop pole positions, and the self-tuner selects a controller that does this.

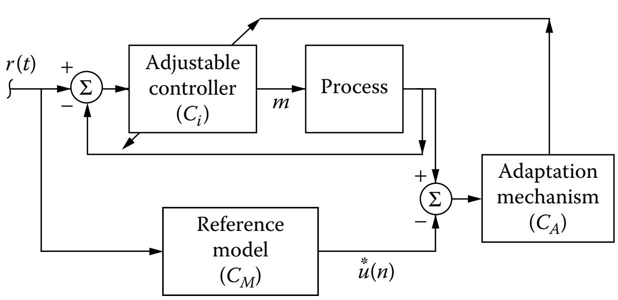


FIG. 2.29d
The main building blocks of a model reference adaptive controller.

The response specification methods can be linked to the optimization methods via various techniques.[8] Pole-placement can handle systems with unknown and variable dead times but requires more computational effort than most optimization-based methods do. Several commercial products offer these methods in one form or another, usually tailored to the requirements of the specific applications.

The model-based methods require that a persistent excitation or "dither" signal be injected into the process. This is normally done in the open loop or at startup, but start-up transients alone are not normally considered sufficient for these self-tuners. The form of the control law is usually a discrete time difference equation that is suitable for digital implementation.

Pattern Recognition Methods

The self-tuning of controllers that utilize pattern recognition procedures is widely used in industrial applications. This method relies upon the introduction of a perturbation of a specified form into the process. This usually is a step or pulse signal. Often the start-up transient is used, which is particularly useful in systems where a special start-up procedure exists, such as the one with temperature controllers.

The process response is then analyzed to extract the key transient performance indicators of the process, which are then used to select the correct controller tuning values. The transient performance indicators can be estimates of rise time, dead time, peak-overshoot, and so on. Some methods use fuzzy or qualitative performance measures, such as the criterion of whether "the time to first overshoot" is very fast, fast, medium, or slow.

The main difference between pattern recognition and the model-based methods is that in the case of pattern recognition no model of the process is constructed. This eliminates problems due to incorrect modeling but limits the usefulness of these methods to processes that can be handled in this way. For example, some of these methods might only work if a dead time and time constant can fully describe the controlled process.

Self-tuners of this type are restricted to three-term (PID) control algorithm applications. In these systems, the PID tuning constants are adjusted according to some tuning rules based upon the measured open-loop transient behavior of the process. The tuning rules are often proprietary variations of such well-known process tuning procedures as the Zeigler Nichols method or others.

A subclass of algorithms is based on testing the stability of the loop. These methods will usually cause the loop to oscillate as in the Zeigler Nichols periodic oscillation (closed-loop) method. Special precautions are applied to avoid runaway oscillation. The period and the controller gain that caused the oscillations are then measured and used to tune the controller.

Eliminating Process Upsets Self-tuning controllers that operate by introducing pulses or other forms of perturbations

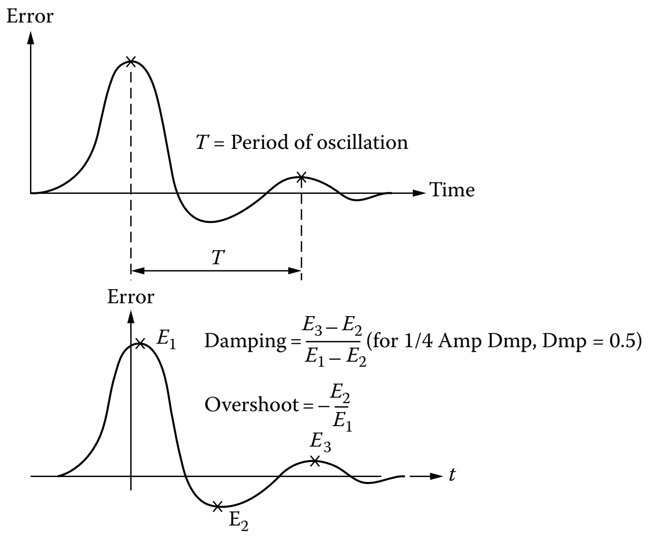


FIG. 2.29e
The control loop tuning is diagnosed by evaluating the period of oscillation (T), the decay or damping ratio, and the amount of overshoot to determine whether the controller is correctly tuned.[10]

are not widely accepted in the process control industry. It is felt that the control loops are unstable enough as it is, and nobody wants additional sources of perturbation.

Therefore, preference is given to self-tuning controllers that do not introduce any upsets but evaluate the controller's response to set-point changes or to other, naturally occurring load variations and other upsets as they take place. In such cases the self-tuning algorithm is usually kept "dormant" (inactive) until an error of some predetermined value (usually at least 1%) develops.

At that point the self-tuning subroutine is activated. After the error has evolved, the self-tuning subroutine checks the response of the controller in terms of the controlled variable's period of oscillation, damping, and overshoot (Figure 2.29e).

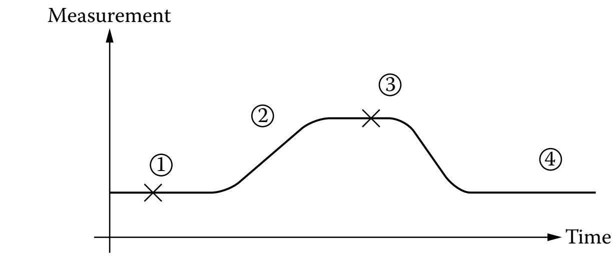


FIG. 2.29g
After evaluating the process response to a step change that increased the process load, another step change is introduced to return the process load to its original value.[10]

Start-Up Sequence As shown in Figure 2.29f, when starting up a control loop for the first time (the pretune phase), the controller is first manually brought to its normal load level of operation. After that, an adjustable-size step change in its output signal to the control value (m) is introduced. This step change in the manipulated variable (m) will cause the controlled variable (c) to draw an S-shaped steady-state reaction curve. From this response curve one can obtain the steady-state process dead time, process sensitivity, and process gain in response to a load increase.

The pretune sequence is continued, as shown in Figure 2.29g, by moving the controller output (m) through another step change back to its original value and determining again the steady-state gain, dead time, and process sensitivity in response to a decrease in load. After these data have been collected and after the "noise-band" of the measurement been identified, these readings are used to automatically determine the start-up PID settings of the controller.

After this pretune or start-up sequence, the self-tuning controller reevaluates its performance after each upset, using the criteria given in Figure 2.29e.

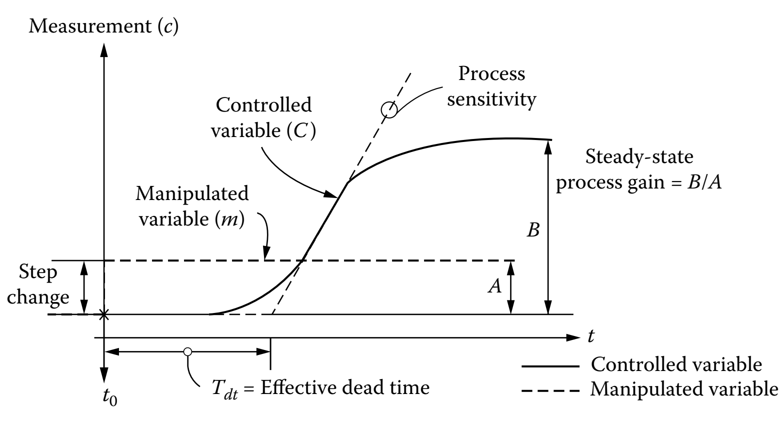


FIG. 2.29f
In the start-up or pretune phase, the self-tuning controller determines the dead time, gain, and sensitivity of the controlled process variable by evaluating its response to a step change.[10]

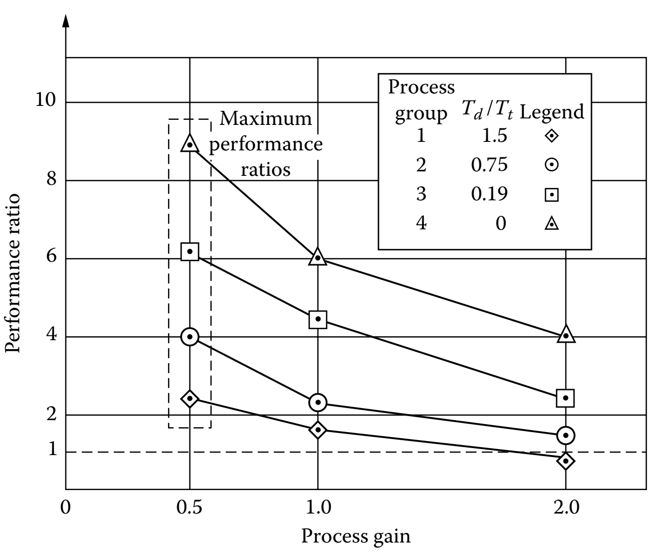
PERFORMANCE

The performance of such self-tuning controllers has been evaluated on different types of processes. The test results are shown in Figure 2.29h, where the performance ratio between conventionally tuned and self-tuned loops is on the vertical.

The conventionally tuned controllers were tuned for the worst condition, and their performance relative to the self-tuning controllers was evaluated on the basis of the integrated absolute error (IAE), which is the total area under the error curve in Figure 2.29e. In evaluating the test results, one can conclude that the improvement in control quality is impressive for slow processes (low gain) having little or no dead time. These are easy processes to control because they start responding to a correction immediately and can be controlled with high gain (narrow proportional band) controllers.

On the other hand, as the ratio of dead time to time constant rises and as the process becomes faster (process gain increases), the performance of self-tuning controllers deteriorates. When the process gain exceeds two and the dead time exceeds the time constant, the performance of the self-tuning controller actually drops below that of the conventionally tuned loop.

This is not surprising because as the dead time rises, the time it takes for evaluating a response also gets longer, which in turn increases the area under the error curve. For this reason, the use of PID control is not recommended for mostly dead time processes. In such cases a sample-and-hold algorithm (see Section 2.2 in this chapter) is recommended.



T_d = Process dead time
T_t = Process time constant

$$\text{Performance ratio} = \frac{\text{Static PID integrated error}}{\text{Self-tuning PID integrated error}}$$

FIG. 2.29h
For slow processes with little dead time, the self-tuning controller outperforms the statically tuned PID controller.[10]

CONCLUSIONS

Process controllers with self-tuning capabilities are mostly stand-alone units that are used for single-loop applications, but self-tuning features are also available in DCS installations or in software packages. The software can be resident in a DCS, or it can be used in general-purpose computers, or perhaps as an external tuning aid.

A special class of self-tuning controllers has evolved for temperature-control applications. These are provided with special tuning procedures (usually based on the response during the warm-up phase or to a set-point change) and include constraints such as minimum overshoot.

Self-tuning controllers used for general-purpose process applications are more sophisticated and are often provided with provisions for multiple loop tuning, feedforward control, gain scheduling, and compensation for actuator and sensor nonlinearity. Most self-tuning controllers will self-tune during startup and will retune both on request and also when upsets occur naturally.

A few self-tuning controllers operate on the basis of continuously introducing excitations into the process. Such excitations can be deterministic set-point steps or doublets (used in pattern recognition methods) or can be persistent set-point variations (often pseudo noise, and usually on the order of 5% of set point).

Self-tuning controllers have gone through substantial development and improvement during the last decades. The better systems no longer depend on pulses or set-point changes for detecting the response of the loop but instead analyze the response to naturally occurring process upsets. These self-tuning controllers give control performance that is superior to that of manually tuned, static PID controllers when the process gain and the process dead time are both low. As the process gain and/or the dead time–to–time constant ratio rises, their performance deteriorates.

References

1. Peterka, V., "Adaptive Digital Regulation of Noisy Systems," Second IFAC Symposium on Identification and Process Parameter Estimation, Paper 6.2, Academica, Prague, 1970.
2. Astrom, K. J., and Wittenmark, B., "On Self-Tuning Regulators," *Automatica*, 9, pp. 185–199, 1973.
3. Clarke, D. W., and Gawthrop, P. G., "Self-Tuning Control," *Proceedings of IEE,* 122, pp. 929–934, 1975.
4. Wellstead, P. E., Edmunds, J. M., Prager, D. L., and Zarrop, P. M., "Pole Zero Assignment Self-Tuning Regulator," *International Journal of Control,* 30, pp. 11–26, 1979.
5. Clarke, P. W., Moktadi, C., and Tuffs, P. S., "Generalized Predictive Control," *Automatica*, 23, pp. 137–448, 1987.
6. Lelic, M. A., and Zarrop, M. B., "Generalized Pole-Placement Self-Tuning Controller," *International Journal of Control,* 46, pp. 569–607, 1987.
7. Sastry, S., and Bodson, M., *Adaptive Control: Stability, Convergence and Robustness,* Englewood Cliffs, NJ: Prentice Hall, 1989.
8. Wellstead, P. E., and Zarrop, M. B., *Self-Tuning Systems,* New York: Wiley, 1991.

9. Seborg, D., Edgar, E., and Mellichamp, D., *Process Control,* New York: Wiley, 1990.
10. Kraus, T. W., and Myron T. J., "Self-Tuning PID Controller Uses Pattern Recognition Approach," *Control Engineering,* June 1984.

Bibliography

Abonyi, J., *Fuzzy Model Identification for Control,* Boston: Birkhauser, 2003.

Astrom, K. J., and Hagglund, T., *Automatic Tuning of PID Controllers*, 2nd ed., Research Triangle Park, NC: ISA Press, 1995.

Astrom, K. J., and Wittenmark, B., *Adaptive Control*, Singapore: Pearson Education, 2001.

Astrom, K. J., and Wittenmark, B., *Adaptive Control*, 2nd ed., Reading, MA: Addison-Wesley, 1995.

Badgwell, T. A., and Qin, S. J., "Review of Nonlinear Model Predictive Control Applications," *Nonlinear Predictive Control: Theory and Practice,* IEEE Publishing, 2001.

Chindambram, M., *Computer Control of Processes*, New Dehli: Narosa Publishing, 2002.

"Distributed Controllers," *Measurements and Control,* April 1991.

Ender, D. B., "Troubleshooting Your PID Control Loop," *InTech,* May 1992.

Fabri, S. G., and Kadirkamanthan, V., *Functional Adaptive Control: An Intelligent Systems Approach,* Heidelberg: Springer-Verlag, 2001.

Howell, J., and Rost, D., "PC Software Tunes Plant Startup," *InTech,* June 1988.

Hunt, K. J., and Sbarbaro, D., "Neural Networks for Nonlinear Internal Model Control," *IEE Proceedings/D,* Vol. 138, 1991.

Maciejowski, J. M., *Predictive Control with Constraints,* New York: Addison-Wesley Longman, 2000.

McMillan, G., "A New Era for Model Predictive Control," *Control,* Vol. 13, No. 5, 2000.

Qin, S. J., and Badgwell, T. A., "An Overview of Industrial Model Predictive Control Technology," *Chemical Process Control,* 5th International Conference on Chemical Process Control, AIChE and CACHE, 1997.

Rossiter, J. A., *Model-Based Predictive Control,* Boca Raton, FL: CRC Press, 2003.

Seborg, D. E., Edgar, T. F., and Shah, S. L., "Adaptive Control Strategies for Process Control," *AIChE Journal,* 32, 1986.

Stephanopoulos, G., and Han, C., "Intelligent Systems in Process Engineering: A Review," *Computers and Chemical Engineering,* Vol. 20, 1996.

Suykens, J. A. K., et al. *Artificial Neural Network for Modelling and Control of Non-Linear Systems,* New York: Kluwer Academic, 1996.

Tao, G., *Adaptive Control Design and Analysis,* New York: John Wiley & Sons, 2003.

VanDored, V., *Techniques for Adaptive Control,* Oxford: Butterworth-Heinemann, 2002.

Wade, H. L., "High-Capability Single-Station Controllers: A Survey," *InTech,* September 1988.

Wellstead, P. E., and Zarrop, M. B., *Self-Tuning Systems,* New York, Wiley, 1991.

Widrow, B., and Stearns, S. D., *Adaptive Signal Processing,* Englewood Cliffs, NJ: Prentice Hall, 1985.

Wojsznis, K. W., Blevins, L. T., and Nixon, M., "Easy Robust Optimal Predictive Controller," Advances in Instrumentation and Control, New Orleans, ISA TECH 2000.

Wojsznis, K. W., et al., "Practical Approach to Tuning MPC," *ISA Transactions,* Vol. 42, January 2003.