ISSN 0005-1179, Automation and Remote Control, 2014, Vol. 75, No. 2, pp. 188–199. © Pleiades Publishing, Ltd., 2014.
Original Russian Text © A.G. Alexandrov, M.V. Palenov, 2014, published in Avtomatika i Telemekhanika, 2014, No. 2, pp. 16–30.

TOPICAL ISSUE

Adaptive PID Controllers: State of the Art and Development Prospects

A. G. Alexandrov and M. V. Palenov

Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia

Received February 28, 2013

Abstract—We give a survey of modern forms of PID controllers and algorithms for their automated tuning and adaptive control. We consider various autotuning and adaptation algorithms. We also consider the frequency adaptive control algorithm in more detail.

DOI: 10.1134/S0005117914020027

1. INTRODUCTION

Proportional-integral-derivative (PID) controllers are widespread in technical systems; they are used in 90–95% of controlling circuits [1, 2]. They achieve control objectives for the majority of technological objects, and their structure is simple and has a small footprint. Over the long history of its use and development, the PID control law has been augmented with new features that aim to improve its efficiency: different realizations for the differentiating part of the controller, a struggle against saturation in the internal component, and feedforward control. However, the key question in using a PID controller has always been about tuning its coefficients. At first, this problem was solved with a human operator who used her knowledge, experience, and intuition to tune the PID controller based on various methods of computing controller coefficients, e.g., the Ziegler–Nichols method [3]. When computers appeared, together with computers came programmable logic controllers (PLC), SCADA systems, and distributed control systems (DCS), and automated tuning (autotuning) methods for PID controllers appeared. Autotuning methods implemented in those systems were designed to tune the controller, either once or at an operator's request, in automatic mode with probing influences. Usually, these probing influences violate the normal operation mode of the control object (CO). Therefore, autotuning is done during the time specifically set aside for it (when the processor boots up or when operation with the current PID controller becomes impossible). However, many COs have nonstationary parameters that drift in time. Therefore, a controller that has been tuned only once cannot achieve the control objective over the entire CO operation. Thus, we need to constantly or periodically tune the coefficients of the PID controller in order to change CO parameters in such a way that the control objective is achieved. This problem can be solved with adaptive control which, depending on the adaptation algorithm, presupposes constant or periodic corrections in the PID controller coefficients. Existing adaptation algorithms fall into two wide classes: direct and indirect. Direct algorithms correct PID controller coefficients based on the analysis of the variable being controlled. Indirect algorithms are based on CO model identification and correcting PID controller coefficients based on the result of this model [1, 4–6].

We should note that apart from PID controllers, the industry picks up on alternative approaches to control: model predictive control (MPC) and algorithms based on fuzzy logic. We discuss these approaches in Sections 4 and 5.

2. VARIATIONS OF PID CONTROLLERS AND METHODS FOR THEIR SYNTHESIS

PID control has been a subject of study for more than 60 years. A huge number of papers, reports, and books have appeared that justify the need in various synthesis methods and give simple rules to compute PID controller parameters. Among the books, we make special notice of a reference on the tuning of PI and PID controllers [7, 8], whose second edition appeared in 2006, that contains a collection of 443 synthesis methods. In 2009, the third edition of this book already features 1731 methods that include probably all methods known from the times of Ziegler and Nichols.

Despite the simplicity of the basic notion of a PID controller, one can distinguish several different forms of implementing a PID control law. This is due to both historical reasons and new ideas coming from general control theory. Thus, PID controllers implemented in different PLC, SCADA systems, and DCS may differ in their structure, which is usually reflected in their documentation.

2.1. Structure and Forms of PID Controllers

The industry uses various forms of PID controllers, more than ten in whole [8]. Let us consider the most popular ones. To simplify our exposition, we will use transition functions.

(1) *Classical form*

$$w_{\mathrm{PID}} = k_c\left(1 + \frac{1}{T_I s} + T_D s\right) = k_c + k_I\frac{1}{s} + k_D s, \tag{1}$$

where s denotes the Laplace transform, k_c is the controller's amplification coefficient, T_I is the integration time constant, $k_I = \frac{k_c}{T_I}$ is the integration coefficients, T_D is the differentiation time constant, and $k_D = k_c T_D$ is the differentiation coefficient.

This form is used in the following industrial controllers:

Allen Bradley PLC5; Bailey FC19; Fanuc series 90-30 and 90-70; Intellution FIX; Honeywell TDC3000; Leeds and Northrup Electromax 5; Yokogawa Field Control Station (FCS); OVEN PLC 100, 150, 154, TRM10, 101, 148, 210; TECHNOKONT P.I.D.-Expert.

(2) *Sequential form* that arose in the use of PID controllers in systems with pneumatic devices:

$$w_{\mathrm{PID}} = k_c(\alpha + T_D s)\left(1 + \frac{1}{\alpha T_I s}\right), \quad T_I \geqslant 4T_D, \quad \alpha > 0. \tag{2}$$

For $\alpha = 1$, this form is used in the following industrial controllers:

Turnbull TCS6000; Alfa-Laval Automation ECA400; Foxboro EXACT 760/761.

Both classical and sequential forms of PID controllers are "unimplementable" since it is impossible to implement pure differentiation. Usually, implementations of these controllers use approximate computation for the derivative:

$$\dot{u}(t) = \lim_{\Delta t\to 0}\frac{\Delta u(t)}{\Delta t} \approx \frac{\Delta u(t)}{\Delta t},$$

for small Δt which is called the discretization period.

(3) *PID controller with a filter*. Both classical and sequential forms of controllers contain pure differentiation, which may lead to a number of problems related to implementation and a large amplification coefficient on high frequencies [1]. One often uses additional filters in two variations:

(i) *PID controller with a differential component filter*:

$$w_{\text{PID}} = k_c \left(1 + \frac{1}{T_I s} + \frac{T_D s}{1 + \frac{T_D}{N} s}\right), \tag{3}$$

where $N = 2 \div 20$ [1].

(ii) *PID controller with an input filter*:

$$w_{\text{PID}} = k_c \left(1 + \frac{1}{T_I s} + T_D s\right) \frac{1}{(T_f s + 1)^n}, \tag{4}$$

where T_f is the filter constant and n is the filter degree which is usually chosen to be $n = 1$.

These forms are used in the following industrial controllers:

Bailey Net 90 for N = 10 and FC156; Concept PIDP1 and PID1; Fischer and Porter DCU 3200 CON for N = 8; Foxboro EXACT I/A series; Hartmann and Braun Freelance 2000; Modicon 984 for 2 < N < 30; Siemens Teleperm/PSC7 ContC/PCS7 CTRL for N = 10 and S7 FB41 CONT_C.

(4) *PID controllers with modified structure.* The PID controllers shown above can be called standard. Alongside with them, there exist a number of ideas for modifications of the PID control law that are based on the so-called feedforward control. The following form of a PID controller is called "PID controller with input command weights," and it can be conveniently written in input–output form:

$$u(t) = k_c(\alpha y^*(t) - y(t)) + k_I \int_0^t (y^*(\tau) - y(\tau))\, d\tau + k_D \frac{d}{dt} (\beta y^*(t) - y(t)), \tag{5}$$

where $y^*(t)$ is the input command, $\alpha \in [0, 1]$ and $\beta \in [0, 1]$ are varied parameters. In case when $\alpha = 1$, $\beta = 0$, the controller is called PID; if $\alpha = 0$ and $\beta = 0$, IPD.

A more detailed survey of existing forms of PID controllers and products of various companies that use these forms is given in [8].

2.2. Methods for Computing PID Controller Coefficients

Existing methods for computing coefficients of PID controllers can be divided into the following groups.

- *Intuitive tuning.* This tuning method assumes that PID controller coefficients are changed independently of each other "intuitively" until the control objective is achieved.
- *Characteristic methods.* These methods arose from practical experience (Ziegler–Nichols methods [3] also belong to this group); in these methods, the controller is tuned with data obtained as a result of testing in an open circuit.
- *Analytic methods* (*algebraic synthesis*). Coefficients of a PID controller are computed from analytic or algebraic dependencies between the object model and control objective (e.g., the method of internal model (IMC) and lambda tuning [1]). Usually, analytic methods yield simple formulas and can be used in adaptive systems, but we need to represent the control objective in an analytic form and obtain a sufficiently accurate model of the control object.
- *Frequency methods.* Frequency characteristics of the control object can be used to tune a PID controller. Usually, these methods are resource–intensive, and they are used to synthesize robust PID controllers.

- *Optimal synthesis.* These methods can be considered as a special form of optimal control, where PID controller coefficients result from numerical optimization methods, computer heuristics, or evolutionary algorithms. Usually, optimization is rather resource-intensive. There also exists an analytic approach where PID controller coefficients receive analytic expressions.

This is not a complete classification. Some methods that are used in practice belong to several different groups at once. The most comprehensive list of existing synthesis methods for PI and PID controllers can be found in [8].

3. AUTOTUNING AND ADAPTATION ALGORITHMS FOR PID CONTROLLERS

Autotuning of PID controllers is more widespread at present than adaptation. This is due to the fact that adaptation algorithms are often more complex than autotuning algorithms and require more computational power. Besides, autotuning algorithms have already been tested since ideas they are based upon come from simple methods for tuning PID controllers similar to Ziegler–Nichols methods. Adaptation algorithms are only passing through this stage now, and the number of adaptive PID controllers in use grows with time. Below we consider autotuning methods and adaptation algorithms used in industrial controllers, SCADA systems, and DCS, including those that we believe may be widespread in the future.

Table 1 shows the currently most popular PLCs that implement automated tuning and adaptive PID control algorithms.

Table 1. Industrial controllers that use automated tuning and adaptation

Manufacturer	Industrial controller model	Automated tuning	Adaptive control
1	2	3	4
ABB	Bitric P	Yes	No
	Digitric 100	Yes	No
	COMMANDER 100	No	No
	COMMANDER 250	No	No
	COMMANDER 310	Yes	No
	COMMANDER 351	Yes	Yes
	COMMANDER 355	Yes	No
	COMMANDER 505	Yes	No
	COMMANDER V100	Yes	No
	COMMANDER V250	Yes	No
	ECA06	Yes	No
	ECA60	Yes	No
	ECA600	Yes	Yes
	MODCELL™ 2050R	Yes	No
	53SL6000	Yes	No
Foxboro	716C	Yes	Yes
	718PL, 718PR	Yes	Yes
	718TC, 718PR	Yes	Yes
	731C	Yes	Yes
	743C	Yes	Yes
	760C	Yes	Yes
	761C	Yes	Yes
	762C	Yes	Yes
	T630C	Yes	Yes

Table 1. (Contd.)

1	2	3	4
Honeywell	UDC100	No	No
	UDC700	Yes	Yes
	UDC900	Yes	Yes
	UDC1000, UDC1500	Yes	Yes
	UDC2300	Yes	Yes
	UDC3300	Yes	Yes
	UDC5000	Yes	Yes
	UDC6300	Yes	Yes
Yokogawa	US1000	Yes	No
	UT320, UT350,	Yes	No
	UT420, UT450	Yes	No
	UT520, UT550, UT750	Yes	No
	UP350, UP550, UP750	Yes	No
	YS150	Yes	Yes
	YS170	Yes	Yes
Siemens	Simatic S7-200,	Yes	Yes
	S7-300, S7-1200	Yes	Yes
OVEN	PLC 100, 150, 154	Yes	No

3.1. Autotuning Methods

Autotuning methods are based on the ideas put forward by Ziegler and Nichols [1, 3], but these ideas have been significantly transformed later. We can distinguish two approaches.

- *Autotuning with a CO transition characteristic.* In this approach, PID controller parameters are chosen based on the analysis of the CO transition characteristic obtained with a ladder-like influence. A drawback of this approach is that the ladder supplied as input must be of sufficient size so that one can distinguish the transition process component against the background of noise and external disturbances.
- *Autotuning with autooscillations.* This approach suggests that we artificially create autooscillations in the control circuit, which lets us identify the so-called limit point (point where the hodograph of the amplitude–phase frequency characteristic APFC of the closed system intersects the negative real axis (–1;0j)) by measuring the amplitude and frequency of autooscillations and using computational formulas to find coefficients for the PID controller. In [2], the authors propose to use a P controller and achieve established autooscillations by increasing its coefficient. In other works [1, 9–11], a two-positional relay or a relay with hysteresis is used for this purpose, which lets one cause autooscillations with bounded amplitude.

3.2. Adaptation Algorithms

Existing adaptation algorithms can be divided into two groups: direct and indirect. Let us consider them in more detail.

<u>3.2.1. Direct adaptive systems.</u> In direct adaptive control algorithms, controller parameters are updated immediately, according to a certain law that depends on the closed system's state. In order to estimate the system state, various approaches can be used.

Direct adaptive systems that use logical rules for tuning the controller (rule-based systems) represent an entire subclass of control systems [1]. Their operation is based on imitating the operation of a tuner that analyzes the state of system output as the input command changes and corrects PID controller parameters. During the operation, one always searches for a compromise between the shortest time of the transition process and stability reserves. Typical PID controller tuning rules are shown in Table 2. This method can also be used for PID controller autotuning.

Table 2. Influence of PID controller coefficients on overcontrolling and stability reserves

	TP time	Overcontrolling	Stability reserves
Increasing k_c	decreases	increases	decrease
Increasing k_I	slowly decreases	increases	decrease
Increasing k_D	slowly decreases	decreases	increase

To control COs whose behaviour is defined by complex and/or nonlinear models, one often uses a table of coefficients (Gain scheduling). For certain CO operation modes (e.g., flight altitude for planes or cranked shaft rotation speed for automobiles), one can find a PID controller that reaches the control objective. However, a PID controller acceptable for one mode may be completely unsuitable for another. Therefore, PID controller coefficients are determined in laboratory or experimental conditions during the control system's development. The CO operation mode is determined with state variables that are available for measurement or with the input command, while PID controller coefficients are chosen from a table depending on the current mode. Thanks to their simplicity, such adaptive systems are very widely used in industry [12, 13].

Among promising direct adaptive control algorithms we can emphasize the following.

- *Iterative gradient tuning* (iterative feedback tuning) algorithm can operate under unpredictable changes in CO parameters but small external disturbances. This method was first proposed in [13]. The main idea is to compute the gradient of PID controller coefficients with respect to tracking error. To compute the gradient, this method uses step-like probing influences that serve as input for the closed system. With the resulting gradient, the method corrects PID controller coefficients.
- The *method of recurrent objective inequalities* [14] is also a direct adaptive control algorithm. The main idea of this method is to use objective inequalities that depend on current and previous values of the measured variables, CO state and control, and that have been constructed based on the chosen control objective. In this algorithm's operation, objective inequalities are not available all at once, they arise during the operation and are therefore solved recurrently. To solve recurrent objective inequalities, one uses finitely converging algorithms with the idea that as the control objective is achieved, coefficients of the control law cease to be corrected.

3.2.2. Indirect adaptation algorithms. Many indirect adaptation algorithms in fact represent a further development of automated tuning algorithms. For instance, the works [10, 11] develop the ideas of identifying the end point and propose to use the least squares method (LSM) for this point's identification. This approach was further developed in [15], where the authors proposed to use several identifiers together with narrowband filters which led to an improvement in the algorithm. Numerous algorithms, e.g., [16, 17], are based on the least squares method that lets one identify various CO models. For PID controllers synthesis, some of the works use algebraic synthesis methods; others, optimization methods.

An adaptation algorithm based on the system's reaction to a step function has been developed in [6, 18, 19]. Initial tuning for the PID controller is done with the results of an experiment in the open circuit, i.e., we perform preliminary automated tuning for the controller. Then subsequent correction of the PID controller coefficients is done based on the results of analyzing the system's reaction to a step-like change in the input command or control that does not exceed 10% of its nominal value.

A similar approach has been described in [2], where adaptation algorithms that use the relay approach are proposed. Initial tuning for the controller is done by closing the CO with a two-step relay. We measure the amplitude and period of the resulting autooscillations, and these values then let us synthesise the PID controller. If it becomes necessary to tune PID controller coefficients, the two-step relay is connected in parallel with the controller.

A promising direction in the development of indirect adaptive control algorithms is frequency adaptive control [2]. The idea of this approach is to use a polyharmonic test signal which is input to the control system. This may become necessary if the input command is not "rich enough" with harmonics, e.g., if it is constant at all times. With a Fourier filter [20], one can single out the useful component that contains information about the CO and thus estimate CO model parameters even under intensive unknown external disturbances. There exist several variations of this approach. In the first variation [2], the system receives as input a single harmonic with frequency equal to the system's resonance frequency, we determine the amplitude and phase of established oscillations on the CO's input and output and then find CO model parameters with this information. For initial measurement of the system's resonance frequency, the method proposes to excite autooscillations with bounded amplitude in the control system with a two-step relay or by increasing the controller's amplification coefficient. In the second variation, one uses a two- and more frequency test signal [20]. In this case, frequencies of the test signal's harmonics must be located "far" from each other. This approach lets one estimate model parameters for COs with a more complex structure.

3.2.3. Frequency adaptive PID controller. The frequency adaptive control algorithm has been developed in [20–22], where the authors solve the key problems of choosing the frequencies and amplitudes for test signal harmonics. Let us consider this algorithm in more detail.

The CO model is represented as a first order link with delay:

$$T\dot{y}(t) + y(t) = k_0 u(t-\tau) + f(t), \tag{6}$$

where $y(t)$ is the CO output, $u(t)$ is the CO input formed by the controller (control signal), $f(t)$ is the unmeasured arbitrary external disturbance, and k_o, T, τ are the unknown amplification coefficient, time constant, and delay respectively.

CO model parameters k_o, T, τ change at arbitrary moments of time in an unknown way. In order for the adaptive controller to operate, it needs these changes to be not too frequent and it needs the changes themselves to be small.

We will use the PID controller in the following form:

$$T_f\dot{u}(t) + u(t) = k_c\left(\varepsilon(t) + \frac{1}{T_I}\int\limits_0^t \varepsilon(\tilde{t})d\tilde{t} + T_D\dot{\varepsilon}(t)\right), \tag{7}$$

where

$$\varepsilon(t) = y^*(t) - y(t) + v(t) \tag{8}$$

is the tracking error, and $v(t)$ is the test signal.

PID controller synthesis is done by estimating the parameters of the model (6) with internal model control (IMC) [1, 23]:

$$k_c = \frac{2\hat{T}+\hat{\tau}}{2\hat{k}_o(\lambda+\hat{\tau})}, \quad T_I = \frac{2\hat{T}+\hat{\tau}}{2}, \quad T_D = \frac{\hat{T}\hat{\tau}}{2\hat{T}+\hat{\tau}}, \quad T_f = \frac{\lambda\hat{\tau}}{2(\lambda+\hat{\tau})}, \tag{9}$$

where $\lambda = \frac{\hat{T}}{2\div 4}$ is the parameter that characterizes system performance, and $\hat{k}_o$, $\hat{T}$, and $\hat{\tau}$ are estimates of CO model parameters.

With this controller and with exactly known CO model parameters $\hat{k}_o = k_0$, $\hat{T} = T$, and $\hat{\tau} = \tau$, the behavior of the closed system (for $f(t) = 0$) up to first order Pade approximation can be described with the following equation [23]:

$$\lambda \dot{y}(t) + y(t) = y^*(t - \tau). \tag{10}$$

For preliminary PID controller tuning, one has to know estimates of model parameters at the initial time moment $\hat{k}_o\Big|_{t=0}$, $\hat{T}\Big|_{t=0}$, and $\hat{\tau}|_{t=0}$.

The test signal is chosen as a sum of sinusoidal signals:

$$v(t) = \rho_1 \sin \omega_1 t + \rho_2 \sin \omega_2 t, \tag{11}$$

where ρ_1, ρ_2, ω_1, and ω_2 are positive numbers.

The CO input $u(t)$ and output $y(t)$ are subject to a Fourier filter:

$$\begin{aligned} \hat{\alpha}_{yk} = \alpha_{yk}(\bar{t}) = \int_{t_F}^{t_F+\bar{t}} y(t) \sin \omega_k t dt, \quad \hat{\beta}_{yk} = \beta_{yk}(\bar{t}) = \int_{t_F}^{t_F+\bar{t}} y(t) \cos \omega_k t dt, \\ \hat{\alpha}_{uk} = \alpha_{uk}(\bar{t}) = \int_{t_F}^{t_F+\bar{t}} u(t) \sin \omega_k t dt, \quad \hat{\beta}_{uk} = \beta_{uk}(\bar{t}) = \int_{t_F}^{t_F+\bar{t}} u(t) \cos \omega_k t dt, \end{aligned} \quad k = 1, 2, \tag{12}$$

where t_F is the filtering start time and $\bar{t}$ is the filtering duration.

Fourier filter outputs $\alpha_{yk}(\bar{t})$, $\beta_{yk}(\bar{t})$ and $\alpha_{uk}(\bar{t})$, $\beta_{uk}(\bar{t})$ converge to the system's frequency parameters as $\bar{t} \to \infty$ [22]:

$$\begin{aligned} \alpha_{yk} + j\beta_{yk} &= \frac{w_{\text{PID}}(j\omega_k) w_o(j\omega_k)}{1 + w_{\text{PID}}(j\omega_k) w_o(j\omega_k)}, \\ \alpha_{uk} + j\beta_{uk} &= \frac{w_o(j\omega_k)}{1 + w_{\text{PID}}(j\omega_k) w_o(j\omega_k)}, \end{aligned} \quad k = 1, 2, \tag{13}$$

where $w_{\text{PID}}(j\omega_k)$ and $w_o(j\omega_k)$ are frequency transition functions of the PID controller and the CO respectively.

The numbers

$$\alpha_k = \text{Re } w_o(j\omega_k), \quad \beta_k = \text{Im } w_o(j\omega_k), \quad k = 1, 2, \tag{14}$$

are called *frequency parameters* of the CO.

Estimates of CO frequency parameters can be easily found as (13):

$$\hat{\alpha}_k = \frac{\alpha_{yk}\alpha_{uk} + \beta_{yk}\beta_{uk}}{\alpha_{uk}^2 + \beta_{uk}^2}, \quad \hat{\beta}_k = \frac{\beta_{yk}\alpha_{uk} - \alpha_{yk}\beta_{uk}}{\alpha_{uk}^2 + \beta_{uk}^2}, \quad k = 1, 2. \tag{15}$$

Now we can derive expressions to find the CO model parameters from (14) [22]. To do so, we rewrite (14) as

$$\alpha_k + j\beta_k = \frac{k_o}{T(j\omega_k) + 1} e^{-j\omega_k \tau}, \quad k = 1, 2, \tag{16}$$

and multiply each kth expression by its complex conjugate; then we get

$$\alpha_k^2 + \beta_k^2 = \frac{k_o^2}{T\omega_k^2 + 1}, \quad k = 1, 2, \tag{17}$$

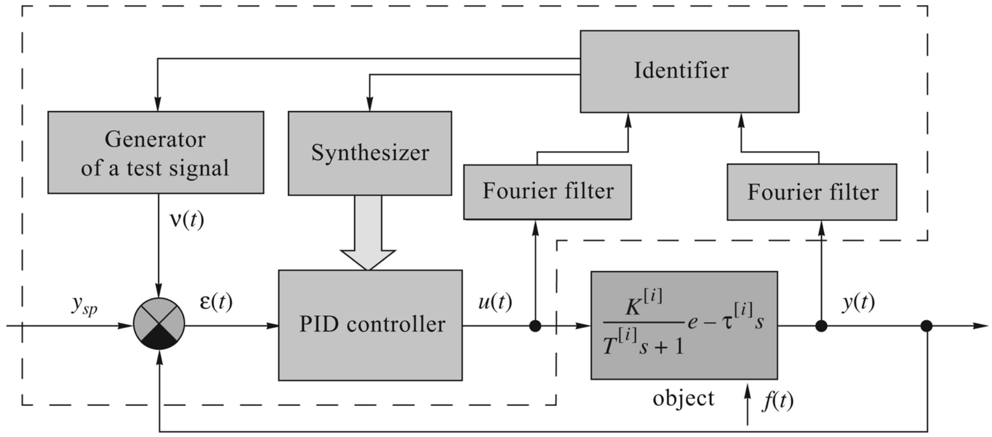

Fig. 1. Structural scheme of a system with frequency adaptive PID controller.

which easily yields expressions for the relation between CO model parameters k_0 and T and CO frequency parameters. To get an expression for the delay, we write (16) with Euler's formula for $k = 1$:

$$(\alpha_1 + j\beta_1)(T(j\omega_1) + 1) = k_o(\cos\omega_1\tau - j\sin\omega_1\tau), \tag{18}$$

which implies that

$$k_o\cos\omega_1\tau = \alpha_1 - T\beta_1\omega_1, \quad k_o\sin\omega_1\tau = \beta_1 + T\alpha_1\omega_1. \tag{19}$$

The latter expressions easily yield a relation between CO frequency model parameters with the value of the delay.

Substituting estimate of CO frequency model parameters into these expressions, we get

$$\hat{T}^2 = \frac{(\hat{\alpha}_2^2 + \hat{\beta}_2^2) - (\hat{\alpha}_1^2 + \hat{\beta}_1^2)}{\omega_1^2(\hat{\alpha}_1^2 + \hat{\beta}_1^2) - \omega_2^2(\hat{\alpha}_2^2 + \hat{\beta}_2^2)}, \quad \hat{k}_o^2 = (\hat{\alpha}_2^2 + \hat{\beta}_2^2)(\hat{T}^2\omega_2^2 + 1), \tag{20}$$
$$\hat{\tau} = -\frac{1}{\omega_1}\operatorname{arctan}\frac{\hat{\beta}_1 + \hat{T}\hat{\alpha}_1\omega_1}{\hat{\alpha}_1 - \hat{T}\hat{\beta}_1\omega_1}.$$

Thus, the adaptation algorithm works as follows.

(1) Compute the PID controller with known estimates $\hat{k}_o\big|_{t=0}$, $\hat{T}\big|_{t=0}$, and $\hat{\tau}|_{t=0}$ for CO model parameters at the initial time moment;

(2) Construct the test signal (11), apply it to the input of the closed system (6), (7), and apply CO input and output to the Fourier filter (12); the filter's outputs for a given filtering time $\bar{t} = \bar{t}^*$ after substituting them into (15) yield estimates for CO frequency model parameters $\hat{\alpha}_k$, $\hat{\beta}_k$, $k = 1, 2$;

(3) Substituting the estimates of CO frequency model parameters into (20), get estimates for CO model parameters and then, substituting them into (9), compute coefficients for the new PID controller, replace the old one with it, and then go to stage (2).

The system's general structure is shown on Fig. 1.

4. MODEL PREDICTIVE CONTROL (MPC)

MPC [24–26] has found many applications in chemical and petrochemical industries; it has also been comprehensively suited for control over slow processes. The idea of this approach is to look for optimal control on a bounded interval.

There exists a large number of MPC strategies (DMC, PFC, PCT, SMCA, OPC, APCS, GPC), but they are all based on the same idea, the difference is only in various CO models. The most general approach is the GPC method that uses difference equations to describe the object, so we consider this approach in more detail.

Predictions of CO output with the GPC method are based on the following CO model:

$$A(z^{-1})y(k) = B(z^{-1})z^{-d}u(k-1) + C(z^{-1})\frac{e(k)}{\Delta}, \tag{21}$$

where $A(z^{-1})$, $B(z^{-1})$, $C(z^{-1})$ are object polynomials, $y(k)$ is the model's output, $u(k-1)$ is the model's input, $e(k)$ are measured external disturbances, $\Delta = 1 - z^{-1}$, $k = 0, 1, 2, \ldots$.

The control is constructed by minimizing the following functional:

$$J(N_1, N_2, N_u) = \sum_{j=N_1}^{N_2} \delta(j)\,[\hat{y}(t+j|t) - w(t+j)]^2 + \sum_{j=1}^{N_u} \lambda(j)[\Delta u(t+j-1)]^2, \tag{22}$$

where N_1, N_2, N_u are horizons of prediction start, prediction end, and control respectively.

Apart from the functional, one must also specify constraints on the controlled variable, the growth of the controlled variable, and control. If there are no constraints, the optimization problem can be solved analytically, and one can write down an explicit solution, but under constraints the optimization problem can be solved only numerically. After solving the problem, we get a vector of parameters for each subsequent time tick, up to the control horizon N_u. The MPC's idea is to apply only the first of these controls and then repeat the same process on the next discrete time step, i.e., solve the optimization problem again, taking current measurements of the object state as initial conditions. This approach is justified by the fact that the model gives its most accurate predictions for the first discrete time step.

It becomes clear why MPC algorithms are well suited for slow technical processes. First, convergence conditions for the control algorithm are not proven under constraints. Second, a numerical solution of the optimization problem requires significant computational resources. For fast processes we would have to solve the optimization problem very often, which would require a very powerful industrial controller.

5. CONTROL ALGORITHMS BASED ON FUZZY LOGIC

Fuzzy control (i.e., control based on the methods of fuzzy set theory) [27] is used when we do not know enough about the control object but already have experience in controlling it. This approach is most often used to control objects whose exact mathematical models are either unknown or so complex that they are hard to linearize or reduce in order to synthesize a controller analytically. Expert knowledge is used to construct the control algorithm. Examples may include a blast furnace or a fractionating tower whose mathematical model contains many empirical coefficients that vary over a wide range and present significant obstacles for identification. At the same time, a qualified operator can control such objects sufficiently well using sensor readings and her accumulated experience.

It is a hard problem to construct a fuzzy controller with a large set of rules, so one often uses PI- or PID variations of a fuzzy controller. PID controllers with fuzzy logic are currently used in commercial systems for homing TV cameras in sports broadcasting, in air conditioning systems, controlling car engines, in automated control over a vacuum cleaner's engine, and many other applications.

One of the most widely used structures of a fuzzy controller (fuzzy PI controller) is shown on Fig. 2. The controller receives the error e as input and computes its time derivative $\frac{de}{dt}$. Then both

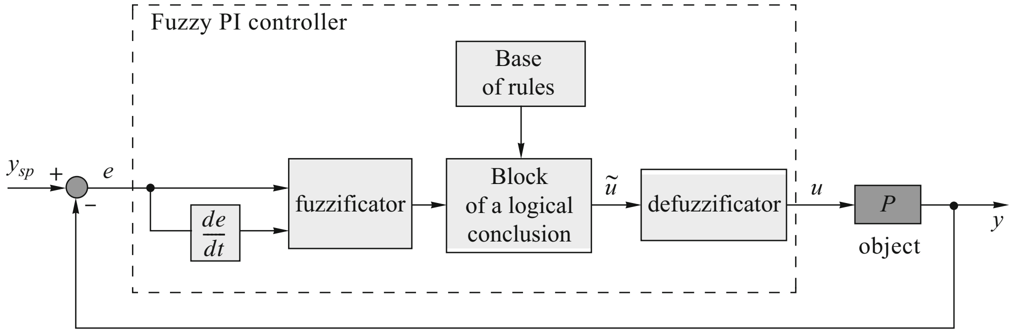

Fig. 2. Fuzzy PI controller.

values are first fuzzyfied (transformed into fuzzy variables) and then the resulting fuzzy variables are used in the fuzzy inference unit to get the controlling influence for the object that will then, after defuzzification (an inverse transform of fuzzy variables into exact ones), serve as input for the controller in the form of a controlling influence u.

In fuzzy controllers, the control is computed with rules based on fuzzy logic. A sample rule might go as follows: if the error is zero and the derivative is positive, the control must be zero.

These rules are formulated from the operator's experience. It is easy to construct a small set of rules. If we need better precision from the object, we will have to specify a more complex fuzzy controller, i.e., specify more rules since we need to cover all possibilities. As a result, fuzzy controllers are used for objects where we do not need to be too precise in the control.

6. CONCLUSION

Despite the existence of a collection of sufficiently diverse and nonuniform adaptive control algorithms, there still remains a rather large gap between theory and practice. It is mostly related to the use of automated tuning algorithms that have already stood the test of time; their modified versions are used in adaptation algorithms. On the other hand, technological objects are complicated enough so that complex adaptation algorithms with a large number of assumptions are hard to apply. Nevertheless, requirements stemming from the industry grow each year, driven by the increase in production rates, technological changes, and increasing flexibility. These factors will, in the end, create a favourable background for universal use of adaptive PID controllers.

ACKNOWLEDGMENTS

This work was supported by the Russian Foundation for Basic Research, project no. 09-08-00338-a.

REFERENCES

1. Åström, K.J. and Hagglund, T., *Advanced PID Control*, North Carolina: ISA, 2006.

2. Rotach, V.Ya., *Teoriya avtomaticheskogo upravleniya: uchebnik dlya VUZov* (Automated Control Theory: University Textbook), Moscow: MEI Publishing House, 2008, 5th ed.

3. Ziegler, J.G. and Nichols, N.B., Optimum Settings for Automatic Controllers, *Trans. ASME*, 1942, vol. 64, pp. 759–768.

4. Denisenko, V.V., PID Controllers: Problems of Implementation, *Sovremen. Tekhnol. Avtomatiz.*, 2008, no. 4, pp. 86–99.

5. Shteinberg, Sh.E., Serezhin, L.P., Zalutskii, I.E., et al., Problems of Construction and Operation of Efficient Control Systems, *Prom. ASU Kontrollery*, 2004, no. 7, pp. 1–7.

6. Shubladze, A.M., Gulyaev, S.V., and Shubladze, A.A., Optimal Automatically Tuned Controllers for General Industry, *Prib. Sist. Upravlen., Kontr., Diagnost.*, 2002, no. 10, pp. 30–33.

7. O'Dwyer, A., *Handbook of PI and PID Controller Tuning Rules*, London: Imperial College Press, 2006, 2nd ed.

8. O'Dwyer, A., *Handbook of PI and PID Controller Tuning Rules*, London: Imperial College Press, 2009, 3nd ed.

9. Astrom, K.J., Hagglund, T., Hang, C.C., and Ho, W.K., Automatic Tuning and Adaptation for PID Controllers—A Survey, *Control Eng. Practice*, 1993, vol. 1, pp. 699–714.

10. Hagllund, T. and Astrom, K.J., Industrial Adaptive Controllers based on Frequency Response Techniques, *Automatica*, 1991, vol. 27, pp. 599–609.

11. Hang, C.C., Astrom, K.J., and Ho, W., Relay Auto-Tuning in the Presence of Static Load Disturbance, *Automatica*, 1993, vol. 29, pp. 563–564.

12. Blevins, T.L., McMillan, G.K., and Wojsznis, W.K., *Advanced Control Unleashed*, North Carolina: ISA, 2003.

13. Hjalmarsson, H., Geers, M., Gunnarsson S., and Lequin O., Interactive Feedback Tuning: Theory and Applications, *IEEE Control Syst. Mag.*, 1998, vol. 18, pp. 26–41.

14. Fomin, V.N., Fradkov, A.L., and Yakubovich, V.A., *Adaptivnoe upravlenie dinamicheskimi ob"ektami* (Adaptive Control for Dynamic Objects), Moscow: Nauka, 1981.

15. Ho, W.K., Hang, C.C., Wojsznis, W., and Tao, Q.H., Frequency Domain Approach to Self-Tuning PID Control, *Control Eng. Practice*, 1996, vol. 4, pp. 807–813.

16. Sato, T. and Kameoka, K., Self-Tuning Type-2 PID Control System and its Application, *Proc. 17th IFAC World Congr.*, Seoul, Korea, 2008, pp. 11287–11292.

17. Suda, N., *PID Control*, Japan: Asakura Shoten Co. Ltd., 1992.

18. Shubladze, A.M., Gulyaev, S.V., and Shubladze, A.A., Adaptive Automatically Tuned PID Controllers, *Prom. ASU Kontrollery*, 2003, no. 6, pp. 24–26.

19. Shubladze, A.M. and Kuznetsov, S.I., Automatically Tuned Industrial PI and PID Controllers, *Avtomatiz. Promyshl.*, 2007, no. 2, pp. 15–17.

20. Alexandrov, A.G., *Metody postroeniya sistem avtomaticheskogo upravleniya* (Design Methods for Automated Control systems), Moscow: Fizmatlit, 2008.

21. Aleksandrov, A.G., Frequency Requlators, *Autom. Remote Control*, 1991, vol. 52, no. 1, part 1, pp. 17–26.

22. Aleksandrov, A.G. and Palenov, M.V., Self-Tuning PID/I Controller, *Autom. Remote Control*, 2011, vol. 72, no. 10, pp. 2010–2022.

23. Visioli, A., Improving the Load Disturbance Rejection Performances of IMC-tuned PID Controllers, *Preprints 18th IFAC World Congr.*, Barcelona, Spain, 2002.

24. Camacho, E.F. and Bordons, C., *Model Predictive Control*, New York: Springer, 1999.

25. Wang, L., *Model Predictive Control System Design and Implementation Using MATLAB*, London: Springer, 2009.

26. *Metody robastnogo, neiro-nechetkogo i adaptivnogo upravleniya. Uchebnik* (Methods of Robust, Neuro-fuzzy, and Adaptive Control. Textbook), Egupov, N.D., Ed., Moscow: Mosk. Gos. Tekh. Univ., 2002, 2nd ed.

27. Rutkovskaya, D., Pilin'skii, M., and Rutkovskii, L., *Neironnye seti, geneticheskie algoritmy i nechetkie sistemy* (Neural Networks, Genetic Algorithms, and Fuzzy Systems), Moscow: Goryachaya Liniya-Telekom, 2006.

This paper was recommended for publication by B.T. Polyak, a member of the Editorial Board