

Chapter 2

Model-Based PI(D) Autotuning

Alberto Leva and Martina Maggio

2.1 Introduction

Any autotuning procedure starts by taking input/output measurements from the process. This can be done in open- or closed-loop, by deliberately injecting some *stimulus* or simply relying on the excitation provided by standard manoeuvres like set point changes, and in a number of different manners, see e.g. the discussions in works like [5, 6, 33] and the references therein.

No matter how the said measurements were obtained, in some cases they are directly employed to determine the regulator parameters. In other cases, the same measurements are conversely used to first obtain a process *model*, i.e. something that permits to simulate the closed-loop system and, sticking to the linear time-invariant single-variable context, virtually always takes the form of a transfer function. That model, together with conveniently expressed specifications, is subsequently used to tune the regulator. Autotuning procedures involving a model in the sense just shown are said to be *model-based*, and model-based autotuning (hereinafter MBAT for short) is the subject of this chapter.

In the literature, many classifications of PI/PID (auto)tuning techniques were proposed, see e.g. again [5] or works like [15, 16, 48, 65, 66], and throughout such a vast research *corpus*, the word “model” is used with various meanings. For example, referring to the synthetic and clear scheme of [52, p. 904], the subject of this chapter would fall into the “parametric model methods” set. It is therefore worth stressing that the distinctive character of MBAT as intended herein is the presence of a process model in a form suitable for analysing and simulating the closed-loop system. The typical workflow of MBAT is summarised in Fig. 2.1.

A. Leva (✉) · M. Maggio

Dipartimento di Elettronica e Informazione, Politecnico di Milano, via Ponzio 34/5,
20133 Milano, Italy

e-mail: leva@elet.polimi.it

M. Maggio

e-mail: maggio@elet.polimi.it

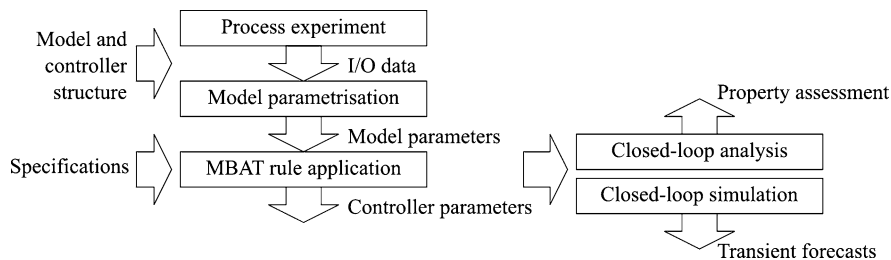


Fig. 2.1 Summary of the typical MBAT workflow

The distinctive character of MBAT just pointed out results in some relevant strengths, but also in substantially two weaknesses. The main strong point is that, as anticipated, the closed-loop system can be simulated before applying the tuned regulator to the real process. Hence, not only structural properties such as stability, performance, and robustness can be checked, but also the most relevant characteristics of the obtained transients, like peak values and durations, can be forecast. Another strength of MBAT is that the control specifications can be stipulated with reference to the model, for example requiring that the closed-loop set point step response be “ n times faster” than that of the model in open loop. In general, the MBAT approach allows one to make the specification more readable and easy to interpret, thus comprehensible also by non-specialist personnel, to the advantage of industrial acceptability.

The mentioned weaknesses, on the other hand, both come from the interplay between the model identification (or better *parameterisation*, since the model structure is typically fixed a priori) and the subsequent regulator tuning. In MBAT, the model structure is almost invariantly fixed a priori—the word “almost” being removable if industrial applications are considered—based on that of the regulator to be tuned (e.g. PI or PID). This unavoidably gives rise to process/model mismatches that can adversely affect the closed-loop transients’ forecasts and sometimes even the assessment of structural properties such as stability. In addition, the process stimulation used to produce the input/output measurements has to frequently obey potentially strict technological limits, and tuning time is frequently a relevant issue for industrial acceptance. In one word, the model needs obtaining from finite (often quite short) sets of noisy data, in the virtually ubiquitous presence of poor excitation. As a result, the model is typically obtained with ad hoc techniques involving some heuristics, such as the method of areas, of moments, and numerous others.

A chapter of this book is entirely devoted to the “identification for PID” subject, so we do not further delve into the matter here and just point out its two most relevant consequences as seen from the MBAT standpoint. First, even if the used tuning rule is well suited for the particular problem at hand, different model identification methods may cause that rule to produce quite significantly different results, and for the same reason, apparently, the same couple of model identification method and tuning rule may behave very differently in different control problems. In the opinion of the authors, this is maybe the toughest difficulty that MBAT has to face

in order to achieve the wide acceptance it potentially deserves. Second, not only the identified model will be nothing more than a (hopefully adequate) approximation of the process dynamics, but the available data will hardly ever—to put it mildly—be sufficient to set up a robustness problem in a formally sound manner. In fact, as shown later on, identification/tuning integration and robustness (or “nonfragility”) are nowadays two relevant research lines in the MBAT domain.

In order to discuss the issues sketched above, this chapter is organised as follows. First, Sect. 2.2 proposes a trivial introductory example to establish the chapter’s perspective. Section 2.3 then provides a panorama of the major MBAT methods proposed in the literature, referring the reader to more extensive works for the details that cannot fit herein. At the same time, suggestions are given on how to organise the presented methods into a simple yet reasoned taxonomy that can easily incorporate and classify the numerous methods not quoted for space reasons. Section 2.4, in part elaborating from the mentioned taxonomy suggestions, addresses the problem of assessing the tuning results and monitoring the control loop, and also of determining when a new autotuning operation is advisable. Up to this point, reference is made to well-assessed MBAT methods and, wherever possible, to methods that do have an industrial realisation and therefore a backlog of use experience. Section 2.5 conversely deals with modern research issues, not yet or only sparingly reflected in the applications, and illustrates the topics that appear more promising based on the experience and the opinion of the authors. Here it is also impossible to exhaust the matter, and thus references are provided for the subjects not touched here. Finally, in Sect. 2.6 some conclusions are drawn, and possible future research perspectives are briefly sketched.

2.2 A Simple Introductory Example

To get a rapid idea of what MBAT is from *both* the methodological and the engineering point of view, it is useful to start with a simple example and some comments. To this end, the following is a deliberately trivial MBAT procedure to tune a PI in the form $R(s) = K(1 + 1/sT_i)$ for an asymptotically stable process.

1. Lead the process to steady state.
2. Set the PI to manual mode and apply a step control variation.
3. Wait for the controlled variable to settle.
4. Attempt to describe the process with a first-order model with delay. Set the gain μ_M to the ratio between the difference of the values of the final and initial controlled variables and the control step amplitude, the delay D_M to the time needed to complete 5% of the overall transient, and the time constant T_M to the time needed to go from 5% to 70% of the same transient.
5. Attempt to enforce a prescribed cutoff frequency ω_c and phase margin φ_m , privileging the latter if its achievement prevents to attain the former and using a cancellation strategy. Omitting simple computation, this means using the MBAT

rule that

$$T_i = T_M, \quad K = \min\left(\frac{\omega_c T_M}{\mu_M}, \frac{T_M}{\mu_M D_M} \left(\frac{\pi}{2} - \varphi_m\right)\right). \quad (2.1)$$

The reader is now encouraged to re-consider the example and notice the following facts. There is a MBAT *rule*, namely (2.1), that realises a *tuning policy* (pole/zero cancellation) with an *objective* or *tuning desire* (a given response speed if possible under a stability degree constraint) expressed by *specifications* (ω_c and φ_m). This is done *assuming a model structure* (first order plus delay) that will attempt to explain the measured data and depends inherently of the chosen *controller structure*. Finally an *experiment* is designed to *parameterise* the model, and various (reasonable but arbitrary) decisions are taken on how to process the data in order to parameterise the model (e.g. the 5% and 70% thresholds, but in real-life cases, also filtering, detrending, outlier removal, and much more).

It should be clear that one thing is a MBAT *rule*, and another is a MBAT *procedure*, as in fact the process of turning the first into the second requires some decision (arbitrary, as *not stemming from the rule*) practically at each step. Correspondingly, discussing the first is merely a methodological fact, while turning it into the second requires a lot of engineering effort, where any of the mentioned decisions has a potentially significant impact on the obtained results and thus the product's applicability and success. This is probably the main reason why MBAT research is difficult, despite prescribing some property for a system with overall less than ten parameters may seem a sinecure. In fact, a major challenge is how to give the mentioned “engineering” facts a methodological dignity, so as to be capable of treating them in a formal manner, and not just like implementation-related incidentals.

This chapter will try to take such an attitude, assuming some reader's familiarity with PID tuning (although no fundamental facts will be totally omitted), with the aim of serving as a guide in a huge corpus of literature that would be impossible even to summarise here.

2.3 Tuning Methods in the Literature

This section reports an extremely synthetic overview of the MBAT history, limited to its major milestones, by describing a few methods that in the opinion of the authors provided some theoretical advance or were particularly successful in the applications. The choice of organising the section in this way was dictated by the enormous extent of the matter: entire books are devoted at listing and sometimes evaluating PID tuning methods [5, 55], and if this section were structured in the same way, it would be nothing more than a shallow rough copy of such works. On the contrary, after detailing the considered model and controller structures, the following two Sects. 2.3.3 and 2.3.4 deal with the PI and the PID controller, telling the MBAT story in extreme synthesis, from the dawn up to recent times but excluding modern research issues. Then, Sects. 2.3.5 and 2.3.6 summarise the *scenario* and propose the announced minimal taxonomy guidelines for MBAT methods, while Sect. 2.3.7 provides a few samples of what was not possible to treat herein.

2.3.1 The Typical Model Structures

As anticipated, models for MBAT need to be simple. One reason is the necessity of parameterising them based on a limited amount of data in the presence of poor excitation. A second one is the need for simple and preferably explicit tuning rules, to the advantage of a safe autotuner operation. A third reason is the opportunity of summarising the main (and hopefully control-relevant) process dynamics with few parameters, to allow for an easy interpretation of their values, also on the part of the typical plant personnel.

In this chapter, the decision was taken to omit treating unstable models. The motivation is that asymptotically stable or integrating ones cover the great majority of the cases of interest, although relevant ones (especially for example in the domain of chemical reactions) are left out. Also, discussing MBAT for unstable processes, especially for the involved robustness issues, would require a large amount of space, and it was considered preferable to disregard the matter completely instead of reporting a necessarily partial treatise.

Given the considerations above, the structures employed in virtually all the literature are the First Order Plus Dead Time (FOPDT), the Integrator Plus Dead Time (IPDT), the First Order and Integrator Plus Dead Time (FOIPDT), and the Second Order Plus Dead Time (SOPDT) in the OverDamped (OD) and UnderDamped (UD) subtypes, the latter appearing typically in mechanical systems. In fact different acronyms are frequently used instead of those reported above, but the meaning of such numerous variants in the literature should be obvious to the reader. For the purpose of this work, the mentioned model structures are thus expressed in the form of SISO transfer functions as follows.

$$\text{FOPDT} \quad M(s) = \frac{\mu_M e^{-sD_M}}{1 + sT_M}, \quad (2.2a)$$

$$\text{IPDT} \quad M(s) = \frac{\mu_M e^{-sD_M}}{s}, \quad (2.2b)$$

$$\text{FOIPDT} \quad M(s) = \frac{\mu_M e^{-sD_M}}{s(1 + sT_M)}, \quad (2.2c)$$

$$\text{SOPDT-OD} \quad M(s) = \frac{\mu_M e^{-sD_M}}{(1 + sT_{M1})(1 + sT_{M2})}, \quad (2.2d)$$

$$\text{SOPDT-UD} \quad M(s) = \frac{\mu_M e^{-sD_M}}{(1 + 2\frac{\xi_M}{\omega_M}s + \frac{s^2}{\omega_M^2})}. \quad (2.2e)$$

2.3.2 The Considered Controller Structures

For apparent space reasons and for the purpose of the chapter, we limit here the scope to the one-degree-of-freedom PI and the PID controllers, and for uniformity, we refer to their so-called ISA forms, i.e. the error-to-control transfer functions

$$\text{PI} \quad R(s) = K \left(1 + \frac{1}{sT_i} \right), \quad (2.3a)$$

$$\text{PID} \quad R(s) = K \left(1 + \frac{1}{sT_i} + \frac{sT_d}{1 + sT_d/N} \right), \quad (2.3b)$$

where K is the gain, T_i and T_d the integral and derivative times, respectively, and N the ratio between T_d and the time constant of the “derivative filter” in the real PID case, the ideal PID corresponding to $N = \infty$.

2.3.3 Some MBAT Methods for the PI

The first and historical tuning method for PI (and PID) controllers is by common opinion that proposed by Ziegler and Nichols in 1942 [75]. In fact, however, previous works by Callender and coworkers [7, 25] in 1935–36 proposed a similar method, thereby constituting the first known “tuning rule”. The work by Callender et al. actually dates back to an internal report of ICI (Alkali) Ltd., Northwich, written in 1934—that is, eight years before the paper by Ziegler and Nichols—and was discovered by Aidan O’Dwyer in 2004. The reader interested in the fascinating story of that discovery can find it in [56]. The Callender rule for the PID is presented later on in Sect. 2.3.4.

Coming back to the Ziegler–Nichols PI rule, it refers to the FOPDT model (2.2a) and the ISA PI (2.3a) and takes the form

$$K = \frac{0.9T_M}{\mu_M D_M}, \quad T_i = 3.33T_M. \quad (2.4)$$

The goal is to obtain a quarter decay ratio for the nominal closed loop, which makes (2.4) applicable for $D_M/T_M \leq 1$. The model is parameterised based on the application of the tangent method to an open-loop process unit step response, whence other published rules equivalent to (2.4) that refer to the tangent parameters directly. It is worth noticing that the ancestor of most MBAT methods *did* specify the model parameterisation procedure—a habit not maintained in several subsequent proposals. Finally, of course the same concept just sketched can be applied to different model structures. For example, there exist a rule analogous to (2.4) for the IPDT model (2.2b) that reads

$$K = \frac{0.9}{\mu_M D_M}, \quad T_i = 3.33D_M, \quad (2.5)$$

and again employs a tangent-based model parameterisation procedure.

Along the idea of constraining the closed-loop damping, or somehow equivalently the maximum overshoot of the set point step response, many methods were introduced in the following years. Two notable proposals are that by Chien, Hrones and Reswick [9] and by Cohen and Coon [10]. The Chien et al. method has the merit of being (probably) the first to distinguish “servo” control problems (where the goal is to track the set point) and “regulatory” ones (where the set point is substantially considered constant, and the problem is to effectively reject load disturbances). Notice, incidentally, that the work by Ziegler and Nichols did not propose such a distinction but looked particularly at disturbance responses. In many subsequent works the statement can be found that the Ziegler and Nichols rules provide “too oscillatory a set point response”: true, but tracking was not their primary goal.

The Chien rules refer to (2.2a) and (2.3a) and read

$$\text{Servo, 0\% overshoot} \quad K = \frac{0.35T_M}{\mu_M D_M}, \quad T_i = 1.17T_M, \quad (2.6a)$$

$$\text{Servo, 20\% overshoot} \quad K = \frac{0.6T_M}{\mu_M D_M}, \quad T_i = T_M, \quad (2.6b)$$

$$\text{Regulatory, 0\% overshoot} \quad K = \frac{0.6T_M}{\mu_M D_M}, \quad T_i = 4D_M, \quad (2.6c)$$

$$\text{Regulatory, 20\% overshoot} \quad K = \frac{0.7T_M}{\mu_M D_M}, \quad T_i = 2.33D_M, \quad (2.6d)$$

and are applicable for $0.1 < D_M/T_M < 1$. Notice the structural similarity to the Ziegler and Nichols ancestor and also that the integral time is made dependent on the model time constant in the servo case and on the model delay in the regulatory one. With the same model, a servo problem thus results in a smaller PI gain than a regulatory one. On the contrary, the servo integral time is smaller than the regulatory one only for processes where the rational dynamics definitely dominates the delay, i.e. for D_M/T_M smaller than $1.17/4 \approx 0.29$ and $1/2.33 \approx 0.43$ in the 0% and 20% overshoot cases, respectively. Observe also that the work by Chien et al. still specifies the model parameterisation procedure (here too, the tangent method).

Also, the Cohen and Coon method refers to (2.2a) and (2.3a). It takes the form

$$K = \frac{1}{\mu_M} \left(0.083 + 0.9 \frac{T_M}{D_M} \right), \quad T_i = T_M \left(\frac{3.33 \frac{D_M}{T_M} + 0.31 \left(\frac{D_M}{T_M} \right)^2}{1 + 2.22 \frac{D_M}{T_M}} \right), \quad (2.7)$$

and is applicable for $0 < D_M/T_M \leq 1$. The goal is quarter closed-loop damping, and once again, the model parameterisation procedure is specified to be tangent-based.

Observing (2.4) through (2.7), a progressively more articulated use of the model parameters can be observed, corresponding to a deeper structuring of the possible control problems. In particular, the D_M/T_M ratio, sometimes also called the “controllability index”, emerges as a quantity with particular relevance. Such an idea

is probably one of the starting points for subsequent elaborations that will be formalised later on when the Internal Model Control (IMC) principle will be applied to MBAT.

In the following years a number of rules similar to those just quoted were proposed and are omitted here for brevity. In parallel, however, a stream of methods started emerging, the goal of which is to optimise some integral index referring to a convenient closed-loop response, instead of punctual quantities such as a decay ratio or an overshoot. Among the most used indices are the Integral of the Squared Error (ISE), the Integral of the Absolute Error (IAE), and the Integral of Time times the Absolute Error (ITAE), defined as

$$\text{ISE} = \int_0^\infty e(t)^2 dt, \quad \text{IAE} = \int_0^\infty |e(t)| dt, \quad \text{ITAE} = \int_0^\infty t |e(t)| dt, \quad (2.8)$$

where $e(t)$ is the error (many other indices are used in the literature, but there is no space here for a discussion). Two pioneering works on the matter are those by Murrill [54] and Rovira [60]. Both refer to (2.2a) and (2.3a) and use a tangent-based parameterisation procedure. The Murrill rules aim at minimising the ISE, IAE or ITAE in the regulatory case, i.e., referring to the unit load disturbance closed-loop step response. Denoting from now on by θ_M the normalised delay expressed as D_M/T_M , those rules are valid for $0.1 \leq \theta_M \leq 1$, taking the form

$$\text{ISE} \quad K = \frac{1.305}{\mu_M \theta_M^{0.959}}, \quad T_i = \frac{T_M \theta_M^{0.739}}{0.492}, \quad (2.9a)$$

$$\text{IAE} \quad K = \frac{0.984}{\mu_M \theta_M^{0.986}}, \quad T_i = \frac{T_M \theta_M^{0.707}}{0.608}, \quad (2.9b)$$

$$\text{ITAE} \quad K = \frac{0.859}{\mu_M \theta_M^{0.977}}, \quad T_i = \frac{T_M \theta_M^{0.68}}{0.674}. \quad (2.9c)$$

The Rovira rules conversely minimise the IAE or the ITAE in the servo case, i.e., for the closed-loop unit step set point response. They are valid for $0.1 \leq \theta_M \leq 1$ and read

$$\text{IAE} \quad K = \frac{0.758}{\mu_M \theta_M^{0.861}}, \quad T_i = \frac{T_M}{1.02 - 0.323\theta_M}, \quad (2.10a)$$

$$\text{ITAE} \quad K = \frac{0.586}{\mu_M \theta_M^{0.916}}, \quad T_i = \frac{T_M}{1.03 - 0.165\theta_M}. \quad (2.10b)$$

The stream of rules aiming at optimising integral indices is still flourishing, also thanks to the availability of computational resources that one could hardly dare to dream at the Murrill and Rovira times. For the scope of this chapter, it is however more interesting to notice how the type of control problem started (further) calling for structurally different relationships between the model and the controller parameters. Quite intuitively, although up to here the history has touched almost exclusively

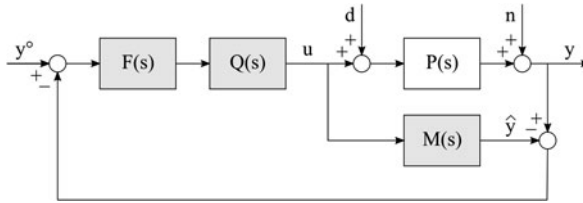


Fig. 2.2 The basic Internal Model Control (IMC) scheme: y^o is the set point, y the controlled variable, u the control signal, d and n respectively a load disturbance and an output noise

FOPDT-based methods, there are extensions to other structures, basically the IPDT and the FOIPDT. There is however no space to comment the matter with sufficient detail, and the conceptual aspects that this chapter aims at discussing can emerge even with such an omission. The reader interested in a complete panorama can refer e.g. to [55].

In fact, more or less in the 1960s, the idea started emerging of tuning the controller by impressing not some characteristics of the obtained closed-loop responses, be it a punctual value or an integral index, but the form of the closed-loop dynamics in the whole, for example by requiring that the transfer function from set point (or load disturbance) to controlled variable be “as similar as possible” to some reference one. A historical MBAT rule conceived that way and still used in many applications under the name of “ λ -tuning” was proposed in 1968 by Dahlin [12]. Starting from the FOPDT model (2.2a), Dahlin’s idea was to make the transfer function from the set point to the controlled variable resemble a first-order one with unity gain, the same delay as the process model, and a specified time constant, which becomes the method’s design parameter (sort of another innovation, notice). Denoting the said time constant with λ , whence the method’s name, the idea corresponds to tuning the regulator so as to approximate the ideal (but not rational) transfer function

$$R_{ID}(s) = \frac{1 + sT_M}{\mu_M(1 + s\lambda - e^{-sD_M})}. \quad (2.11)$$

If e^{-sD_M} is replaced by its (1,0) Padé approximation $1 - sD_M$, the resulting controller is a PI, and the Dahlin rules are thus

$$K = \frac{T_M}{\mu_M(D_M + \lambda)}, \quad T_i = T_M. \quad (2.12)$$

In fact and more in general, having the Dahlin rule as an anticipation, between the 1970s and the 1980s, time became ripe for the introduction in MBAT of the already mentioned IMC principle that originated a vast family of methods. There is quit a bit of debate on which was the first work on that matter, and since we do not intend here to enter said debate, we prefer to illustrate the IMC principle in a view to its usefulness in MBAT, then show some of the most widely used methods, and later on employ the principle for some methodological discussions on model error and robustness.

In a nutshell, the IMC idea is explained by the block diagram of Fig. 2.2 and thinking for now to the case of an asymptotically stable process $P(s)$: if the model $M(s)$ of that process is perfect and if there is neither load disturbance d nor output noise n , then apparently the process output y and the model output $\hat{y}$ are equal, the scheme is open-loop, and the transfer function from the set point y° to the controlled variable y is $F(s)Q(s)M(s)$. If *in addition* $Q(s)$ can be taken as the inverse of $P(s)$, then the desired closed-loop dynamics from set point to controlled variable can be chosen arbitrarily by selecting $F(s)$. Finally, the IMC controller (the grey blocks in Fig. 2.2) is immediately shown to be equivalent to the feedback one given by the error-to-control transfer function

$$R(s) = \frac{F(s)Q(s)}{1 - F(s)Q(s)M(s)}. \quad (2.13)$$

Of course the above hypotheses are in general unrealistic, and if the IMC principle is to be used with a high-fidelity process model, and thereby accepting a controller the structure of which depends on that of said model, there are a number of aspects to address and a vast literature. If the IMC principle is to be applied to MBAT, however, things are somehow simpler, in that the principle has to be viewed basically as a flexible means to obtain explicit tuning rules for the model structures of interest, provided that the said structures allow (exactly if possible, and in the opposite case with reasonable approximation) to derive a controller of the desired form. For example, if one takes as $M(s)$ the FOPDT model (2.2a), as $Q(s)$ the inverse of its minimum-phase part, i.e. $(1 + sT_M)/\mu\mu_M$, and sets $F(s) = 1/(1 + s\lambda)$, the Dahlin rule is re-obtained and could be called the first IMC-PI one. The IMC principle however allows for more insight into the problem, leading to the “improved” rules mentioned here in the following and to some robustness-related considerations reported later on. On the other hand, however, as the literature began focusing on IMC-based and similar methods, attention was progressively shifted on the characteristics of the control problem involving the process *model*, and works accounting for the particular model parameterisation procedure used (or even mentioning it) ceased to be the majority.

Coming back to the mainstream history, a notable “improved IMC” rule for the PI was proposed by Rivera, Skogestad and Morari in 1986 [59] in a view to improve performance especially in the case of significant process delays, which reads

$$K = \frac{T_M + D_M/2}{\mu_M\lambda}, \quad T_i = T_M + \frac{D_M}{2} \quad (2.14)$$

and is presented in the quoted paper together with a wealth of considerations on IMC-based tuning that is impossible to summarise here but is highly advisable for reading. Another successful IMC improvement, known as “SIMC”, was proposed by Skogestad [63] with the aim (simplifying the matter a bit) of avoiding excessive values of the integral time, thus sluggish transients due to poor control activity. The SIMC rules for the PI are

$$K = \frac{T_M}{\mu_M(D_M + \lambda)}, \quad T_i = \min(T_M, 4(D_M + \lambda)). \quad (2.15)$$

Another relevant research line, closely related to IMC-based tuning, is the so-called “Direct Synthesis” (DS), exemplified by [8], that shares with the IMC the possibility of being employed with different model structures, thereby unifying in a single design framework a matter that with older approaches needed treating on a per-structure basis.

To end this section, at least another tuning *rationale* needs mentioning, namely that based on interpolation of results obtained via numerical optimisation. A successful result of the said approach is the AMIGO PI proposed by Hägglund and Åström in 2002 [24] and is somehow borderline with respect to classical MBAT since it extends the MIGO method previously proposed by the same authors. The MIGO assumes the process transfer function to be known, while the AMIGO tunes based on three parameters (the gain, the apparent dead time and time constant, and the inflection point slope) deducible from a step response under the hypotheses that it is essentially monotonic and is therefore considered by the authors a “revisitation” of the Ziegler and Nichols rule. The AMIGO goal is to maximise the integral gain subject to a constraint on the maximum sensitivity, in an attempt to solve the performance/robustness tradeoff—a subject that has been gaining increasing interest in the last years. The method is applicable to asymptotically stable and integrating processes, see [24] for its description.

The last tuning method here mentioned, based again on interpolation, is the so-called “kappa–tau” one [23] that obtains the PI parameters as

$$K = \frac{A_0}{k_M} e^{(A_1 \tau_M + A_2 \tau_M^2)}, \quad T_i = B_0 D_M e^{(B_1 \tau_M + B_2 \tau_M^2)}, \quad (2.16)$$

where $k_M = \mu_M \theta_M$, and $\tau_M = D_M / (D_M + T_M)$ is another definition of “normalised delay”, differing from θ_M as it lies in the 0–1 range; the constants $A_{0,1,2}$ and $B_{0,1,2}$, tabulated in [23], are obtained by numerically optimising performance under a robustness constraint expressed on the magnitude margin M_s , defined as

$$M_s = \max_{\omega} \left| \frac{1}{1 + L(j\omega)} \right|, \quad (2.17)$$

where $L(s)$ is the open-loop transfer function, and for which the values of 1.4 and 2.0 provide two sets of constants A and B , respectively corresponding to conservative and aggressive tunings.

2.3.4 Some MBAT Methods for the PID

Coming to the PID controller, more or less the same story just followed for the PI can be told, with two main differences. First, given the richer controller structure, more model structures are present. Second, some methods refer to the ideal PID—i.e. to (2.3b) with $N = \infty$ —and some to the real one, the first type of methods being not the totality, but a significant majority.

Here too, the ancestor is the work by Callender et al. [7]. It refers to the ideal PID and the FOPDT model (2.2a) and takes the form

$$K = \frac{1.066}{\mu_M D_M}, \quad T_i = 1.418 D_M, \quad T_d = a D_M, \quad a = [0.353, 0.47]. \quad (2.18)$$

Much more known, as for the PI case, is however the method by Ziegler and Nichols [75], also referring to the ideal PID and providing its three parameters as

$$K = \frac{a T_M}{\mu_M D_M}, \quad T_i = 2 D_M, \quad T_d = \frac{D_M}{2}, \quad a = [1.2, 2] \quad (2.19)$$

for a FOPDT model parameterised with the tangent method, aiming at quarter decay ratio. Similar proposals can be found in the following years for other model structures. An example for the IPDT is the work by Ford [17], whose rules are

$$K = \frac{1.48}{\mu_M D_M}, \quad T_i = 2 T_M, \quad T_d = 0.37 D_M, \quad (2.20)$$

with the goal of a 2.7:1 decay ratio, the model being assumed known (i.e. the parameterisation procedure not being thought of as part of the tuning). The quoted work by Chien et al. [9] provides overshoot-related rules also for the PID that we omit here for brevity.

The use of integral indices emerged a few years later, like for the PI, and still continues. An “old” example given by the Murrill [54] is minimum IAE rules for the FOPDT model in the regulatory case

$$K = \frac{1.4835}{\mu_M \theta_M^{0.981}}, \quad T_i = \frac{T_M \theta_M^{0.749}}{0.878}, \quad T_d = 0.482 T_M \theta_M^{1.137} \quad (2.21)$$

with a tangent-based parameterisation procedure. A more recent example was proposed for both the servo and regulatory case and the IPDT model by Visioli [72], referring to the various indices. For example, the minimum servo ISE rules are

$$K = \frac{1.37}{\mu_M D_M}, \quad T_i = 1.69 D : M, \quad T_d = 0.59 D_M \quad (2.22)$$

with the model assumed already parameterised.

The idea of tuning the controller by impressing the form of the closed-loop dynamics in the whole emerged also for the PID, and here too the work by Dahlin [12] provides historical MBAT rule for the FOPDT model. If (2.11) is rationally approximated by using not a (1,0) but a (1.1) Padé approximation of the delay term, i.e. $(1 - s D_M/2)/(1 + s D_M/2)$, the resulting controller is a PID, whence the Dahlin (or “PID λ -tuning”) rules

$$K = \frac{T_M + D_M/2}{\mu_M (D_M + \lambda)}, \quad T_i = T_M + \frac{D_M}{2}, \quad T_d = \frac{T_M D_M/2}{T_M + D_M/2}, \quad (2.23)$$

referring to the ideal controller.

The IMC principle came then into play, and in the PID case it was exploited more extensively than in the PI one. As before, in fact, the Dahlin rule can be considered the first (ideal) IMC-PID one, but much more has been done. For example, a wide variety of applications of the IMC principle to PID MBAT can be found in [59], where all the model structures (2.2a)–(2.2e), plus several others, are considered. For some structures, the PID is augmented with a lag term and thus turned into a real one. This is not done on a general basis, however. On the other hand, Leva and Colombo proposed a tuning rule invariantly producing a real PID [43], referring (2.2a) and (2.3b), and composed of the relationships

$$\begin{aligned} T_i &= T_M + \frac{D_M^2}{2(D_M + \lambda)}, & K &= \frac{T_i}{\mu_M(D_M + \lambda)}, \\ N &= \frac{T_i(D_M + \lambda)}{\lambda T_i} - 1, & T_d &= \frac{\lambda D_M N}{2(D_M + \lambda)}, \end{aligned} \quad (2.24)$$

to be used in sequence; the design parameter λ is interpreted as the desired closed-loop dominant time constant. The number of IMC-derived tuning rules is impressive and impossible to review, and the same applies to the similar DS principle, some applications of which to the PID can be found in the paper [8] quoted above. A major merit of both approaches, but in particular—see the comparative discussion in [8]—of the IMC, is the possibility of conducting robustness analysis in a sense compatible with MBAT: some words on that important matter will be spent later on.

Continuing the panorama, the interpolation-based approach was applied to the PID too, and there exist corresponding versions of the quoted AMIGO [24] and “kappa–tau” [23] methods, plus many others.

As can be seen, the PI and PID stories are very parallel. It is now time to discuss the announced differences. First, the two zeroes of the PID make it more suited to the PI for second- and even higher-order systems, see e.g. the discussions on “which controller is adequate” in many works such as [5]. This was recognised long ago and gave rise to many considerations on the role of unmodelled dynamics that general frameworks such as the IMC and the robust control theory, see e.g. [14, 53] as references of more or less that period, subsequently comprehended in a unitary treatise.

From this point of view, two historical methods are worth mentioning, with some considerations on their potential and pitfalls as expressed by their authors at the time the research was published. The first method was proposed by Haalman in 1965 [20]. The method has the goal of making the nominal open-loop transfer function $L(s) = R(s)M(s)$ resemble the desired one

$$L_{ID}(s) = \frac{2e^{-sD_M}}{3D_M s}, \quad (2.25)$$

which corresponds to achieving a cutoff frequency ω_c of $2D_M/3$ and a phase margin of 50° approximately. If the (non-integrating) process model is in the form (2.2a), it

is advised to select a PI, while if it is in the form (2.2d), an ideal PID is chosen, and the tuning formulæ are

$$K = \frac{2(T_{M1} + T_{M2})}{3\mu_M D_M}, \quad T_i = T_{M1} + T_{M2}, \quad T_d = \frac{T_{M1}T_{M2}}{T_{M1} + T_{M2}}. \quad (2.26)$$

Haalman (correctly) stated, right from the paper's title, that the method is very suited for processes with overdamped response and significant delay. In fact, being ω_c inversely proportional to D_M , the requested response might become too fast if D_M is small. It is additionally worth noticing that the method does not use any model mismatch information, thus there is no way of guiding its operation on the basis of the model accuracy.

Such a problem had however started receiving attention a few years before, as testified by the second method here mentioned, namely that introduced by Kessler in 1958 [38, 39] and known as the ‘‘Symmetric Optimum’’ (SO). In fact, despite being older than the Haalman method, the SO one contains several ideas that have been widely developed in the following years. The most important one is to assume that the (non integrating) process model be

$$M(s) = \frac{\mu e^{-sD}}{\prod_{k=1}^m (1 + sT_k) \prod_{h=1}^n (1 + sT_h)}, \quad (2.27)$$

i.e. either in the form (2.2a) if $m = 1$ or (2.2d) if $m = 2$, but with some other poles accounting for the process dynamics not described by the model. It is furthermore assumed that the time constants T_k dominate the process dynamics, i.e. that

$$T_k \gg D + \sum_{h=1}^n (1 + sT_h) \quad \forall k. \quad (2.28)$$

The quantity $L + \sum_{h=1}^n (1 + sT_h)$ can then be interpreted as the time constant of a transfer function representing the unmodelled process dynamics, which is still rough but extremely foreseeing way to account for model mismatch. Denoting by T_{um} the above quantity, the SO method takes as approximate model

$$M'(s) = \frac{\mu e^{-sD}}{(1 + sT_{um}) \prod_{k=1}^m (sT_k)} \quad (2.29)$$

and designs the controller so that the nominal cutoff frequency be $1/2T_{um}$ (thus reducing the demand as the mismatch increases, i.e. automating a very wise practice) and that the nominal open-loop magnitude $|R(j\omega)M'(j\omega)|$ have a slope of -20 dB/dec in the frequency interval from $1/4mT_{um}$ to $1/T_{um}$. The SO tuning formulæ (also the PI ones for completeness) are given in Table 2.1.

The SO method was immediately recognised to have both strengths and weaknesses. It performs very well, provided that the process delay is small since the time constants T_k must dominate also the delay—see (2.28)—thus it is especially suited for electromechanical systems, despite being keen to generate low-frequency

Table 2.1 The SO tuning formulæ

	K	T_i	T_d
PI	$\frac{T_1}{2\mu T_{um}}$	$4T_{um}$	
PID ($T_2 \geq 8T_{um}$)	$\frac{T_1 T_2}{8\mu T_{um}^2}$	$16T_{um}$	$4T_{um}$
PID ($T_2 \geq 8T_{um}$)	$\frac{T_1(T_2 + 4T_{um})}{8\mu T_{um}^2}$	$T_2 + 4T_{um}$	$\frac{4T_2 T_{um}}{T_2 + 4T_{um}}$

regulator zeroes, i.e. overshoots in the set point responses. In any case the idea of “tuning also on the basis of unmodelled dynamics” is very powerful. Apart from the methodological consequences discussed later on, for example, the SO method has led to several evolutions, maybe the most successful being a similar one called BO or “Betrag Optimum”, often translated as “magnitude optimum” [68]. The application of the PID to higher-order model structures has also received attention in general, as shown e.g. by [34].

The second peculiarity of the PID story is the presence or absence of the “derivative filter”, the term $1/(1 + sT_d/N)$ in (2.3b), in the tuning formulæ. Curiously enough, such a relevant issue was not brought to a systematic attention, at least to the best of the authors’ knowledge, until quite recent years with respect to the overall MBAT history, see e.g. [51]. As noticed in subsequent more extensive works such as [35], the derivative filter effect is invariantly a modification of the high-frequency PID behaviour, which improves noise insensitivity and robustness versus model errors acting “slightly above” the cutoff frequency, at a generally affordable cost in terms of stability degree (e.g. phase margin). However, if this is the sole effect in the case of “series” PID realisations, where the filter is cascaded to *all* the controller dynamics, the same is not true for parallel ones like the ISA (2.3b), where the derivative filter also causes the zeroes of the real PID to not coincide with those of the ideal form anymore. Hence, in the PID form probably most widely used in the applications for a number of reasons (e.g. the ease of switching on and off the individual control actions) that it is impossible to discuss here, default values for the derivative filter parameter N , quoting from the abstract of [35], “are much less natural” and can sometimes cause undesired behaviours that are definitely hard to understand for the typical personnel using those controllers.

2.3.5 Summarising the Story

Apart from the recent research issues discussed in Sect. 2.5, the story told so far can be assumed to represent quite well the panorama of assessed and industrially used MBAT methods. In the said story, also in view to better relate modern issues to previously emerging problems, three periods can be broadly distinguished.

At the dawn of MBAT, methods tended (and quite intuitively *needed*) to be extremely simple, owing both to the limited computational resources available and to

the fact that much control-theoretical subjects nowadays familiar were at the time being studied, and even if some methodological results were available, their industrial application was far to come—think of optimal and robust control, just to give a sample. In that period specifications were typically expressed as punctual quantities (damping, overshoot, and so on), or it was required to minimise some integral index or cost function, rules were almost invariantly structure-specific, but (probably owing at least partially to a research attitude privileging application-related aspects with respect to formal analysis) the model parameterisation procedure was often treated as an integral part of the overall MBAT method. Industrial realisations appear.

In the subsequent period, a transition can be observed towards specifications expressed, broadly speaking, in the form of some reference model (sometimes mentioned explicitly, like in works denoted by the “model matching” or similar concepts like [1]). The idea of accounting somehow for the limited capability of the model to represent the process emerges with increasing strengths, and a visible convergence starts with methodologies (such as the mentioned robust control) that have reached a sufficient maturity. At the same time, however, the focus tends to shift away from the model parameterisation phase, concentrating instead on the analysis of the nominal control problem, i.e. that containing the regulator and the model used for its tuning. Industrial realisations become more frequent, especially integrating MBAT in existing controllers. The end of this period may be considered to coincide with the introduction of DS and especially IMC-based techniques.

More recently, the DS and IMC analysis capabilities have led to revisit older methods in view to use more expressive specifications (e.g. gain or phase margins), and the increasing availability of computing power has permitted the use of interpolation. More attention to robustness is paid, and some discussions start emerging on what is to be meant by “robustness” in the MBAT context.

As a result of such an evolution, many MBAT industrial products exist, while however many questions are still open for modern research, as will be discussed in the following.

2.3.6 Guidelines for a Minimal Taxonomy

Based on the discussion above, a minimal taxonomy is easily provided that classifies the methods, and the following four axes can be proposed.

The first one is the type of regulator addressed (PI, ideal or real PID, and so on): such information allows us to initially relate the method to preferred classes of problems. To do so, one can use the considerations on controller selection given e.g. in [5] and/or more implementation-related knowledge, e.g. the poor suitability of an ideal PID to highly noisy measurements, and so forth.

The second axis is the type of model used, bearing in mind (the consequences of this will emerge later on) that the structure of the said model de facto dictates that of the regulator or, taking an alternative viewpoint, is dictated by it, making in any

case the model/regulator structure choice a unitary fact. Again, the capability of the used model to represent some particular dynamics (e.g. the FOPDT one is poorly suited to oscillatory processes) helps relate the method to classes of problems from another point of view.

The third axis is the way specifications are accepted, and basically one can distinguish characteristics of some closed-loop response, parameters of some reference closed-loop model, and the objective of minimising some integral index (e.g. the ISE). This aspect of the proposed taxonomy allows one to relate the *rationale* of a method—a third point of view—to the types of control problems for which it is best suited (for example, set point tracking versus disturbance rejection).

The fourth axis is the mechanism behind the tuning rules, that can be broadly classified into analytical synthesis, optimisation, and possibly interpolation (and also soft computing methodologies, although space limitations oblige to omit them here). Such information helps us estimate the computational weight and the type of information needed to use the method, i.e., helps us judge on its suitability for the particular application at hand.

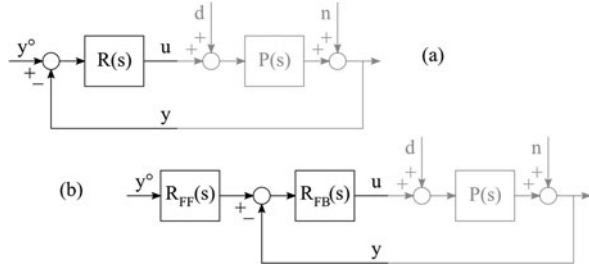
One could also think of a fifth axis, distinguishing whether or not (and in the affirmative case in what sense) the model parameterisation phase is considered a part of the overall tuning procedure, but such a subject has quite recently appeared in the literature, see e.g. some considerations in [30], although first noticed practically at the origins of MBAT, and is thereby deferred to the following of this chapter.

It is finally worth noticing that other taxonomies were attempted in the literature, e.g. [55], as well as a number of works evaluating and comparing (thus implicitly classifying or helping classify) PI/PID tuning rules in several ways, see e.g. [3, 15, 16, 66] and many others. The authors do not believe that the one proposed herein is in any sense superior but find this way of thinking to be slightly better as a means to select the “best” method to use in a certain type of control problems. It has of course to be acknowledged that subjective opinions play a relevant role in statements like that just proposed.

2.3.7 A Few Samples of What Was Left out Here

As shown e.g. in the introductory sections of [55], there is a full population of PID controller forms: series, parallel, interacting, noninteracting and so forth. Although here the ISA form was adopted for brevity, some rules were conceived and designed having a certain PID form in mind. Apart from the need of converting parameters from one form to another, which is totally straightforward, the *operation* of an MBAT procedure can sometimes be affected by the way the PID is written, especially for what concerns the commutation between the “normal operation” and the “tuning” modes. The matter has potentially relevant implications, in that if not accounted properly in the controller engineering, it can cause undesired and hard to explain malfunctions but is of scarce conceptual interest, thus it was omitted (suffice to include here a *caveat* not to underestimate such facts, directed to anybody possibly wishing to implement an autotuner).

Fig. 2.3 Schemes with a 1-dof (a) and a 2-dof (b) controller



Also, there are numerous tuning methods (also of the MBAT type) that employ soft computing techniques such as neural networks, fuzzy logic, genetic algorithms and so on. On an average basis such techniques are less used in the applications than in more classical ones, and in any case too impossible to analyse in detail in a chapter like this. That matter was thus omitted.

Abandoning soft computing, it is then to be noticed that a number of other techniques were necessarily left out such as pole placement, prescription of gain and phase margins, and much more. The main reason is that all of them fall into the defined categories and would just have lengthened the list, but the choices made here must in no sense be considered a “ranking” of techniques, and, as anticipated, are largely subjective.

A final word is worth spending on two-degree-of-freedom (2-dof) controller tuning methods, corresponding to the scheme of Fig. 2.3 and that were omitted only for space reasons.

The idea of exploiting the 2-dof nature of many industrial PIDs, also frequently called the “set point weighing” functionality, is however very interesting, as it allows us to focus the tuning of the feedback path on stability and disturbance rejection, subsequently employing the feedforward path to recover and/or improve set point tracking. Also, if the 2-dof ISA form

$$U = K \left(bY^\circ - Y + \frac{1}{sT_i}(Y^\circ - Y) + \frac{sT_d}{1 + sT_d/N}(cY^\circ - Y) \right) \quad (2.30)$$

is adopted, where b and c are the set point weights in the proportional and derivative actions, the feedback and feedforward blocks denoted in Fig. 2.3 as R_{FB} and R_{FF} become respectively a standard 1-dof ISA PID in the form (2.3b)—or which standard rules are readily applied—and a unity-gain set point pre-filter, namely

$$R_{FF}(s) = \frac{1 + s(bT_i + T_d/N) + s^2T_iT_d(c + b/N)}{1 + s(T_i + T_d/N) + s^2T_iT_d(1 + 1/N)}. \quad (2.31)$$

This allows us to set up two-step MBAT procedures, first tuning R_{FB} for stability, robustness and disturbance rejection, and then R_{FF} for tracking. In other words, using MBAT in this way is an effective means to counteract the “overemphasis on the set point response” that is observed in very engineering-oriented works like [61].

2.3.8 A Final Remark

The reader may have noticed that the review of methods shown so far practically stops some years ago. This is however consistent with the idea of separating “well assessed” and “innovative” methods. From a similar point of view, it is extremely interesting to look at the great classification effort published in 2005 and 2006 by Ang, Chong and Li [4, 47]. The quoted papers list the main products for PI/PID tuning, in the form of both modules for control environments and independent software packages, together with the patents filed on PID tuning in the United States, Japan, Korea, and by the World Intellectual Property Organisation; also, there is a list of autotuning PID hardware modules.

From so extensive a survey, three facts are worth noticing. First, the release years of the mentioned hardware modules span the range 1985–2001 [4, Table V]. Second, out of 42 software packages listed, 32 support MBAT [*ibid.*, Table IV]. Third, in patents from 1971 to 2000 one can observe an increasing percentage of methods based on “non-excitation” (i.e. that do not deliberately stimulate the system); also in patents, the most used tuning approach is “formula” (i.e. simplifying a bit explicit tuning relationships) followed by “rule” (more complex mappings from process information to controller parameters, from example heuristic relationships up to fuzzy logic) and with an increasing percentage of “optimisation” [*ibid.*, Figs. 6 and 7].

If one assumes that (a) hardware objects tend to incarnate well-established technology, and only a significant methodological advance results in a new hardware generation, (b) independent software packages are typically conceived so as to deliver advanced functionalities, not convenient and/or impractical to realise in hardware with sufficient flexibility and ease of use, and (c) patents are clearly filed before the contained claims are industrially exploited, the following conclusions—in the opinion of the authors—can be drawn.

- MBAT is felt as a promising technology, but its use is mostly limited to “quite advanced” tools, typically requiring some user interaction—thus competence—especially to drive the identification phase, see e.g. the LOOP-PRO[®] product suite¹ [11].
- Many hardware modules do encompass MBAT, typically in the form of very simple methods, and quite often complement it with tuning maps or rule-based synthesis. At the same time, however, a significant amount of industrial products (especially, but not only, low-end ones) stick to alternative approaches like relay-based tuning. A probable reason for that is the practical absence, in the relay-based context, of any arbitrariness and/or ambiguity on how the process information—frequency response points—is obtained and treated [74].
- Much MBAT industrial research is underway on the model identification front, aiming especially at reducing the process upset and the tuning time. However, see also the item above, the said research has not (yet) fully unleashed its potential.

¹Loop-Pro is a registered trademark of Control Station, Inc., One Technology Drive, Tolland, CT 06084.

- Modern computational capabilities make it affordable to include optimisation in the loop controllers directly—another recently started research field.

Projecting the above remark onto the scope of this chapter, it is therefore not surprising that among current research lines, particular effort is devoted to the possible effects of a model selection and/or parameterisation that is in some sense “improper” for the used tuning rule in the addressed problem. This motivates the choice of the topics discussed in Sect. 2.5.

2.4 Tuning Assessment, Monitoring and Retuning

Before treating such modern issues, however, three aspects of (MBAT) autotuners’ operation, the latter two deeply intertwined, need discussing.

The first aspect is how the obtained tuning can be assessed before making it effective. Given the simple models used, it is not complicated to check the tuning in nominal conditions. The real question is whether or not the desired properties (think for example to some stability and performance level) carry over from the nominal system—that containing the model—to the real one. This clearly leads to consider unmodelled dynamics, and apart from the mentioned SO historical example, the natural theory to be brought in is that of robust control.

When declining the idea of robustness in MBAT, however, an important distinction is necessary. To set up a robustness problem, one has to specify which property has to be (made) robust, with respect to the variation of what, in which set. Assuming that the property is specified (e.g. stability) and the varying object is the process transfer function, in MBAT the set is not available by definition. In real-life cases, one makes just a single experiment, and any model error estimate (the matter will be discussed soon) can at most measure the model’s inability to explain the data. No information can be obtained on the effects of a process variation, because in a single experiment of acceptable duration, there can be no such variation. In MBAT, therefore, the problem of “robustness” needs splitting into two. One is to guarantee that the PID tuned on the model will control “well enough” the real process as it was when data was collected, and this can be tackled by using model error information. The other is to quantify the “amount of model error” that can be tolerated while still preserving the property, and this can only be done *a priori*. It is important to bear such a distinction in mind, although the literature (no criticism intended) is quite often silent on the matter.

Model error estimates can only be obtained by analysing the identified model response in conjunction with the identification data. In order to simplify a potentially complex matter, attempts were made to treat the model error as a parametric error for the model, or more generally, to assume some structure for the model error itself, deduced by some reference response, see e.g. [50]. However, for a rigorous treatise, the model error description must be nonparametric, as shown in [42] together with a procedure to obtain a magnitude overbound for the additive error in the frequency domain. Based on nonparametric overbounds, works like [43], referring to the IMC

PID method, indicate how to guarantee that at least the real process as it was at the time of the experiment will be well controlled. On a similar front, the recent paper [70] provides a very neat problem formulation, based on two tuning parameters for performance (specifically, closed-loop time constant) and robustness, also clarifying how complex the underlying optimisation problem can be and how to obtain a simple tuning rule [*ibid.*, p. 66] serving the purpose.

Coming to the a priori quantification of the tolerable model error, in the past years this was practically the only problem addressed, and the typical tools were the classical stability degree indicators, namely the gain and phase margins, see e.g. [18, 27–29, 73]. More recently, a shift can be observed towards better indicators like the peak (nominal) sensitivity M_s , defined as

$$M_s = \max_{\omega} |S(j\omega)|, \quad S(j\omega) = \frac{1}{1 + R(j\omega)M(j\omega)}, \quad (2.32)$$

and the performance/robustness tradeoff is managed both by evaluating existing rules, as in [71], and proposing a new one with that specific purpose. Incidentally, accounting for robustness poses an interesting identification-related question: does the model have reproduce the identification data as closely as possible, thus minimising the error in the frequency domain on an average basis, or concentrate its fidelity on some band, for example to loosen the acceptable error bound in the vicinity of the cutoff? Some words on the matter will be spent later on when discussing the so-called “contextual tuning”.

The second aspect touched here is how to monitor the tuning on-line, which is strongly connected to the third one, i.e. how to decide when a re-tuning is advisable. The subject has significant connection with fault detection, as recognised since many years [36], but in the MBAT context it is more frequent, for implementation convenience, to encounter “detectors” for specific problems. Two notable examples, covering by the way the most relevant facts to detect, are the methods proposed by Hägglund [21] to detect oscillations by studying the magnitude of the IAE between subsequent zero crossings of the error, and in [22] to spot sluggish loops by the so called “idle index” that (roughly) indicates that the controlled and control variables’ derivatives have the same sign for too long during a step-generated transients. Another interesting work is [64] that uses normalised (dimensionless) settling time and IAE to detect sluggish or non-optimal loop behaviours, referring to the IMC rules, while from a more methodological standpoint, [32] attempts to cast the matter into a comparative framework with minimum variance control. Recent evolutions of the loop monitoring idea, including suggestions for controller retuning, can be found e.g. in [69].

2.5 Modern Research Issues

Despite various decades of past research, in the MBAT context many questions still stand open. In the impossibility of even just mentioning all of them, the choice

is made here to discuss those that the authors feel as more relevant, admitting in advance that the choice is somehow influenced by their research interests.

Specifically, whilst up to some years ago MBAT was basically viewed as a handy methodology to devise tuning recipes, modern research is progressively concentrating on the high-level information conveyed by the presence of a model. Stretching the lexicon a bit, one could say that an MBAT-based autotuner is “more conscious” of the controlled process, in the same way as possessing a model of some object implies deeper a conscience of that object than just knowing how it responds to some stimulus. This “consciousness” is being nowadays devoted a great effort, and the fronts on which the said effort is exerted are (in the opinion of the authors) the major actual research lines.

In the first place, right from the beginning MBAT rules were conceived for some tuning objective (quarter damping, 0% overshoot and so on). As anticipated above when proposing taxonomic guidelines, the type of specifications accepted is strongly connected to the tuning objectives, thereby helping classify the method. An interesting issue is to conversely classify the *control problem* and then act accordingly for the regulator tuning. A proposal in this sense can be found in [41], where first some quantities are defined that characterise the nominal control problem, i.e., the (model PID) couple as emerging from the chosen MBAT method. The suggested quantities, apart from the nominal cutoff frequency ω_c and phase margin φ_m , are the PID high-frequency gain, the ratio between the integral time and the model dominant time constant, the PID frequency response magnitude at ω_c , the maximum PID phase lead, the ratio between the frequency corresponding to the said maximum lead and ω_c , and finally the ratio between the closed-loop settling time, roughly estimated as $5/\omega_c$, and that of the model (assumed asymptotically stable) in open loop. Based on those quantities and a set of weights characterising the tuning desires (e.g. fast response versus high damping) and the qualitative amount of noise, a decision mechanism allows to choose, within a pre-specified set of MBAT rules, the one that best suits the tuning desires in the case at hand. Further discussions on the idea of automatically selecting a tuning rule, and correspondingly of quantitatively characterising a tuning problem, can be found in [44]. Most likely, endowing industrial autotuners with the capability of selecting the “best” rule to use will enhance their success.

On a similar front, research is addressing “multi-objective” tuning methods as a way to manage the typical MBAT tradeoffs, namely (to quote the most relevant) that opposing “servo” to “regulatory” tuning, i.e. set point tracking to disturbance rejection, and performance to robustness. As noticed since some years, see e.g. [19], this problem has quite a lot to do with optimisation, since a natural way of trading two conflicting figures of merit versus one another is to optimise one of the two subject to a constraint on the latter. The arising optimisation problems are however difficult to formalise, since many of the involved entities are more keen to a qualitative than a quantitative descriptions. Attempts were thus made to use soft computing techniques: for example, [40] proposes a method based on clonal selection accounting for diversity, distributed computation, adaptation and self-monitoring function and contains interesting comparisons to similar proposals. Alternatively, the quoted paper [19] introduces an LMI-based optimisation framework, while [67] numerically

maximises “the shortest distance from the Nyquist curve of the open-loop transfer function to the critical point -1 ” and thus optimises the peak sensitivity as per (2.32).

Avoiding a lengthy list of references, in the particular context of multi-objective tuning a significant presence of soft computing techniques can be observed, with particular reference to genetic algorithms, and it can also be noticed that several works directly refer to a specific application, see e.g. [57]. More classical techniques, at least in the first years in which the subject received interest, tend conversely to focus on the mentioned LMI idea, since it was shown a few years before that well-assessed formalisms, such as the $\mathcal{H}_\infty$ or the μ -synthesis ones, give rise to intractable problems when the controller structure has to be constrained into the PID form [37].

In parallel, however, works such as the book [13] or [62] had systematically exploited, among others, the concept of “stabilising regions”. Given a (nominal) model, the idea is to analytically determine the region of the PID parameter space that guarantees stability, thus providing the search space for possible subsequent optimisations. Similar ideas can be found in [26], where the problem of designing for robust performance is cast into the simultaneous stabilisation of a family of complex polynomials.

More recently, a quest for simpler approaches to the multi-objective tuning problem can be observed, somehow as a consequence of the research path just sketched, together with some *renaissance*, in this new and better formalised context, of old tuning indices. A very interesting example is the paper [49] that addresses different model structures by minimising the IAE for step and load responses (which is well aligned with traditional research) in view however to manage “robustness, performance and control effort” in a coordinated manner; the interested reader can refer in particular to Sect. 4 of the paper.

In this scenario, modern trends are finally well exemplified by the mentioned works [70, 71]. In the first one, an ISA PID MBAT rule is presented in which two design parameters govern respectively the closed-loop time constant and the degree of robustness, i.e. of acceptable error, and default values for the said parameters are also devised—an important matter to enhance simplicity and thus industrial acceptability. The second paper analyses how widely used MBAT rules comply with their claimed robustness specification, indicating that for some set of plants, the said specifications are hardly or not attained, while for others, there is even too wide a margin. The outcome, and certainly a direction to research upon, is that not only robustness specifications need including in MBAT (in the sense just discussed, and somehow envisaged in previous works such as [42, 43]) but checked a posteriori.

Indeed, all the above considerations could be collectively grouped in what appears to be a re-consideration of the role of the tuning model in MBAT. As pointed out e.g. in [45], when a tuning parameter is related to robustness, its choice cannot be done on the sole basis of the nominal model—quite intuitive but curiously e.g. in the FOPDT-based IMC context there are several proposals to select λ based on the model parameters. Along the same reasoning, it was also shown that the said “default” parameter choices normally tend to be excessively conservative, to the

detriment of performance—an idea that subsequent and more complete investigations such as the quoted paper [71] did confirm.

In synthesis, the idea has been emerging that one relevant MBAT issue is the necessity for the model to be precise especially near the cutoff, which however is a *result* of the tuning, not a prior information. From a different but analogous perspective, in fact, the model should be evaluated based on the tuning results it produces rather than on the capability of reproducing the identification data. As shown in the quoted work, a major MBAT problem is that the identification phase contains numerous decisions that are substantially arbitrary: the model structure, the way the process is possibly stimulated and data is obtained, and the parameterisation procedure. Since relay feedback was introduced, it become clear that the said procedure produces local information—typically, one or a few Nyquist curve points—but is practically free of such arbitrary choices. As such, attempts were made to join relay-based identification and MBAT, an interesting and somehow pioneering paper being [31]. Such research is still ongoing, and a recent result is the so-called “contextual tuning” approach proposed in [46] as a more systematic treatise. The quoted paper shows that, in principle, any MBAT rule can be used by fitting the model to be exact at a certain frequency, determined by suitably driving a relay experiment. That frequency will become the cutoff one, and if the selected MBAT rule is used in such a way to nominally produce exactly that cutoff frequency, the result is the simultaneous parameterisation of the PID and the tuning model. The contextual approach has proven to produce tendentiously better degrees of robustness with respect to those of the same MBAT rule used with non-contextual parameterisation methods such as the method of areas and appears to be a promising research subject.

Another relevant topic, particularly in recent years, is that of “fragility”, see e.g. [2]. Quoting from that reference, “if robustness of the control loop indicates the margin of variation in which the plant characteristics with a fixed controller may vary, the controller fragility has a similar meaning in terms of the variation of its own parameters”, based on previous research reported e.g. in [13], the maximum ℓ_2 norm of the PID parameter vector can be suggested as a measure of “controller parametric stability”, and from that, [2] proposes an *absolute fragility index* and shows that, besides robustness as traditionally meant, also fragility is a quantity that may vary significantly among different MBAT rules when applied to similar cases. While the mainstream research on fragility concentrates on the non-nominalities that can be introduced by imperfections in the controller components if analogue, or numerical errors in its realisation if digital, the obtained results can *de facto* be considered applicable whatever the cause is for the PID having “slightly different parameters than it should”, owing e.g. to some measured outliers adversely affecting the model parameterisation.

Concerning again the model identification phase, a relevant topic is the investigation of automatic structure selection. Given the poor excitation typical of MBAT, classical techniques based on the prediction error whiteness, and also most methods coming from the identification for control domain, are in fact difficult to apply and often inadequate. A promising idea is to detect the “correct” model structure based on the recognition of some patterns in reference response, for example, an oscillating step response calls (in the continuous time domain) for a couple of complex

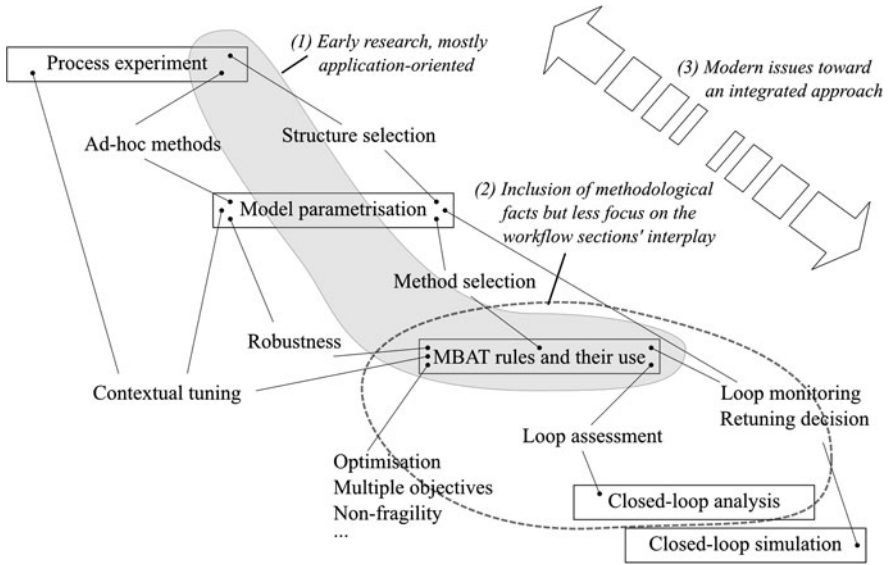


Fig. 2.4 The MBAT workflow revisited in the light of research developments

poles. A way to accomplish the task can be the use of neural networks coupled to suitable data pre-processing, as in [58]. Incidentally, the “structural information” provided by such techniques is potentially richer than that coming from more classical ones: sticking to the example just sketched, in fact, not only can one say that a second-order model is advised, but also that its poles should be complex—another research issue to address.

Finally, it is worth noticing that modern computation tools allow for MBAT rules that would have been impossible to realise only a few years ago. In fact, many literature proposals are de facto based on the interpolation of results obtained with some optimisation mechanism that at present can be directly plugged into the process hardware. Needless to say, this gives rise to very interesting perspectives, although from more an engineering than a strictly methodological point of view.

To end this section, in Fig. 2.4 a scheme is proposed that shows again the typical MBAT phases, like Fig. 2.1 did very synthetically at the beginning of this chapter, but also outlines the main open problems and their collocation in the overall workflow, together with mapping onto it the main research phases outlined before.

2.6 Conclusions and Future Perspectives

After presenting a necessarily brief and partial review of MBAT, it is now the time to draw some conclusions that can be summarised in the following items and implicitly express the authors’ opinions on future perspectives.

- MBAT is a powerful tool because the presence of a process model allows one to tell more on the closed loop than it is possible with other tuning approaches.
- However, the model must be suitable to do so, or the conclusions drawn on the forecast loop behaviour, even if a more or less satisfactory tuning was actually achieved, may be heavily incorrect. In one word, the model can exert a great power but bears an equally great responsibility
- Better still, the *model identification* and the *tuning assessment* bear such responsibility. MBAT poses very peculiar issues as for both, and there is still room for a lot of research.
- From the methodological standpoint, beside improvements in the various phases of the MBAT process, an *integrated* approach said process is necessary, nowadays envisaged quite clearly and pursued by modern research lines.
- From the application-related standpoint, MBAT has not yet unleashed all of its potential. Most likely, one reason for that is some lack still present in the aforementioned integrated view of the process. Indeed, the interest for MBAT research definitely comes also from engineering issues.
- On the same front, the impressing increase in the computational power available “on the plant floor” will allow one to realise solutions that only some years ago were just wishful thinking, and possibly also to port the said solutions directly into the loop controllers—i.e., not only in centralised software packages used in the control room. There will be much to think and design also in the user interface and ergonomics of the so envisaged product, since a wide usage can be foreseen, also on the part of non-specialist personnel.

To conclude, the authors believe that, for sure, MBAT has some pitfalls, but within the various possible approaches to (PID) autotuning, it is probably the one that—while being researched upon—led to the deepest insight into the (auto)tuning problem and also into the way a tuning procedure has to be engineered. The authors’ hope is that the few and partial considerations reported herein may be helpful for the community who devote their effort to such a fascinating research and engineering subject.

References

1. Aguirre, L.A.: PID tuning based on model matching. *Electron. Lett.* **28**(25), 2269–2271 (1992)
2. Alfaro, V.M.: PID controllers’ fragility. *ISA Trans.* **46**(4), 555–559 (2007)
3. Alvarez-Ramirez, J., Monroy-Loperena, R., Velasco, A., Urrea, R.: Optimality of internal model control tuning rules. *Ind. Eng. Chem. Res.* **43**(24), 7951–7958 (2004)
4. Ang, K.H., Chong, G.C.Y., Li, Y.: PID control system analysis, design, and technology. *IEEE Trans. Control Syst. Technol.* **13**(4), 559–576 (2004)
5. Åström, K.J., Hägglund, T.: *Advanced PID Control*. Instrument Society of America, Research Triangle Park (2006)
6. Åström, K.J., Hägglund, T., Hang, C.C., Ho, W.K.: Automatic tuning and adaptation for PID controllers—a survey. *Control Eng. Pract.* **1**(4), 699–714 (1993)
7. Callander, A., Hartree, D.R., Porter, A.: Time lag in a control system. *Philos. Trans. R. Soc. Lond.* **235**, 415–444 (1935)

8. Chen, D., Seborg, D.E.: PI/PID controller design based on direct synthesis and disturbance rejection. *Ind. Eng. Chem. Process Des. Dev.* **41**(19), 4807–4822 (2002)
9. Chien, K.L., Hrones, J.A., Reswick, J.B.: On the automatic control of generalised passive systems. *Trans. Am. Soc. Mech. Eng.* **74**(2), 175–185 (1952)
10. Cohen, G.H., Coon, G.A.: Theoretical consideration of retarded control. *Trans. Am. Soc. Mech. Eng.* **75**(6), 827–834 (1953)
11. Cooper, D.: *Practical Process Control Using Control Station*. Control Station, Inc., Storrs (2004)
12. Dahlin, E.G.: Designing and tuning digital controllers. *Instrum. Control Syst.* **41**(6), 77–81 (1968)
13. Datta, A., Ho, M., Bhattacharyya, S.P.: *Structure and Synthesis of PID Controllers*. Springer, Berlin (2000)
14. Doyle, J.C., Francis, B.A., Tannenbaum, A.R.: *Feedback Control Theory*. MacMillan, Basingstoke (1992)
15. Foley, M.W., Julien, R.H., Copeland, B.R.: A comparison of PID controller tuning methods. *Can. J. Chem. Eng.* **83**, 712–722 (2005)
16. Foley, M.W., Ramharack, N.R., Copeland, B.R.: Comparison of PI controller tuning methods. *Ind. Eng. Chem. Res.* **44**(17), 6741–6750 (2005)
17. Ford, R.L.: The determination of the optimum process-controller settings and their confirmation by means of an electronic simulator. *IEE Proc., Part 2* **101**(80), 141–155 and 173–177 (1953)
18. Fung, H.W., Wang, Q.G., Lee, T.H.: PI tuning in terms of gain and phase margins. *Automatica* **34**(9), 1145–1149 (1998)
19. Ge, M., Chiu, M.S., Wang, Q.G.: Robust PID controller design via LMI approach. *J. Process Control* **12**(1), 3–13 (2002)
20. Haalman, A.: Adjusting controllers for a deadtime process. *Control Eng.* **65**, 71–73 (1965)
21. Hägglund, T.: A control-loop performance monitor. *Control Eng. Pract.* **3**(11), 1543–1551 (1995)
22. Hägglund, T.: Automatic detection of sluggish control loops. *Control Eng. Pract.* **7**(12), 1505–1511 (1999)
23. Hägglund, T., Åström, K.J.: Automatic Tuning of PID Controllers. In: Levine, S. (ed.) *The Control Handbook*, pp. 817–826. CRC Press, Boca Raton (1996)
24. Hägglund, T., Åström, K.J.: Revisiting the Ziegler–Nichols rules for PI control. *Asian J. Control* **4**(4), 364–380 (2002)
25. Hartree, D.R., Porter, A., Callander, A., Stevenson, A.B.: Time lag in a control system—II. *Proc. R. Soc. Lond. Ser. A* **161**, 460–476 (1937)
26. Ho, M.T., Lin, C.Y.: PID controller design for robust performance. *IEEE Trans. Autom. Control* **48**(8), 1404–1409 (2003)
27. Ho, W.K., Gan, O.P., Tay, E.B., Ang, E.L.: Performance and gain and phase margins of well-known PID tuning formulas. *IEEE Trans. Control Syst. Technol.* **4**(4), 473–477 (1996)
28. Ho, W.K., Hang, C.C., Cao, L.S.: Tuning of PID controllers based on gain and phase margin specifications. *Automatica* **31**(3), 497–502 (1995)
29. Ho, W.K., Hang, C.C., Zhou, J.: Self-tuning PID control of a plant with under-damped response with specifications on gain and phase margins. *IEEE Trans. Control Syst. Technol.* **5**(4), 446–452 (1997)
30. Huang, H.P., Chen, C.L., Lai, C.W., Wang, G.B.: Autotuning for model-based PID controllers. *AIChE J.* **42**(9), 2687–2691 (1996)
31. Huang, H.P., Jeng, J.C., Luo, K.Y.: Auto-tune system using single-run relay feedback test and model-based controller design. *J. Process Control* **15**(6), 713–727 (2005)
32. Hugo, A.J.: Performance assessment of single-loop industrial controllers. *J. Process Control* **16**(8), 785–794 (2006)
33. Gyöngy, I.J., Clarke, D.W.: On the automatic tuning and adaptation of PID controllers. *Control Eng. Pract.* **14**(2), 149–163 (2006)
34. Isaksson, A.J., Graebe, S.F.: Analytical PID parameter expression for higher order systems. *Automatica* **35**(6), 1121–1130 (1999)

35. Isaksson, A.J., Graebe, S.F.: Derivative filter is an integral part of PID design. *IEE Proc., Control Theory Appl.* **149**(1), 41–45 (2002)
36. Isermann, R.: Process fault detection based on modeling and estimation methods—a survey. *Automatica* **20**(4), 387–404 (1984)
37. Iwasaki, T., Skelton, R.E.: All fix-order $\mathcal{H}_\infty$ controllers: observer-based structure and covariance bounds. *IEEE Trans. Autom. Control* **40**(3), 512–516 (1995)
38. Kessler, C.: Das Symmetrische Optimum, Teil I. Regelungstechnik **6**(11), 395–400 (1958) (in German)
39. Kessler, C.: Das Symmetrische Optimum, Teil II. Regelungstechnik **6**(12), 432–436 (1958) (in German)
40. Kim, D.H., Cho, J.H.: Robust PID controller tuning using multiobjective optimization based on clonal selection of immune algorithm. *Knowl.-Based Intell. Inf. Eng. Syst.* **3213**, 50–56 (2004)
41. Leva, A.: Automated selection of model-based PID tuning methods based on the nominal control problem. In: *Proc. ECC2007*, Kos, Greece, pp. 5341–5347 (2007)
42. Leva, A., Colombo, A.M.: Estimating model mismatch overbounds for the robust autotuning of industrial regulators. *Automatica* **36**(12), 1855–1861 (2000)
43. Leva, A., Colombo, A.M.: On the IMC-based synthesis of the feedback block of ISA-PID regulators. *Trans. Inst. Meas. Control* **26**(5), 417–440 (2004)
44. Leva, A., Donida, F.: Quality indices for the autotuning of industrial regulators. *IET Control Theory Appl.* **3**(2), 170–180 (2009)
45. Leva, A., Schiavo, F.: On the role of the process model in model-based autotuning. In: *Proc. 16th IFAC World Congress*, Praha, Czech Republic (2005)
46. Leva, A., Negro, S., Papadopoulos, A.V.: PI/PID autotuning with contextual model parametrisation. *J. Process Control* **20**(4), 452–463 (2010)
47. Li, Y., Ang, K.H., Chong, C.Y.: Patents, software, and hardware for PID control—an overview and analysis of the current art. *IEEE Control Syst Mag* 42–54, February 2006
48. Lin, M.G., Lakshminarayanan, S., Rangaiah, G.P.: A comparative study of recent/popular PID tuning rules for stable, First-Order Plus Dead Time, Single-Input Single-Output processes. *Ind. Eng. Chem. Res.* **47**(2), 344–368 (2008)
49. Madhuranthakam, C., Elkelam, A., Budman, H.: Optimal tuning of PID controllers for FOPTD, SOPTD and SOPTD with lead processes. *Chem. Eng. Process.* **47**(2), 251–264 (2008)
50. Maffezzoni, C., Rocco, P.: Robust tuning of PID regulators based on step response identification. *Eur. J. Control* **3**(2), 125–136 (1997)
51. Mellquist, M., Crisafulli, S., MacLeod, I.M.: Derivative filtering issues for industrial PID controllers. In: *Proc. IEAust Control-97*, Sydney, Australia, pp. 486–491 (1997)
52. Moradi, M.H.: New techniques for PID controller design. In: *Proc. 2003 IEEE Conference on Control Applications*, Istanbul, Turkey, pp. 903–908 (2003)
53. Morari, M., Zafiriou, E.: *Robust Process Control*. Prentice-Hall, Upper Saddle River (1989)
54. Murrill, P.W.: *Automatic Control of Processes*. International Textbook Co., Scranton (1967)
55. O'Dwyer, A.: *Handbook of PI and PID Controller Tuning Rules*. Imperial College Press, London (2006)
56. O'Dwyer, A.: PID control: the early years. In: *Control in the IT Sector Seminar*, Cork Institute of Technology, Cork, Ireland, May 2005
57. Pedersen, G.K.M., Yang, Z.: Multi-objective PID-controller tuning for a magnetic levitation system using NSGA-II. In: *Proceedings of the 8th Annual Conference on Genetic and Evolutionary computation*, pp. 1737–1744 (2006)
58. Piroddi, L., Leva, A.: Step response classification for model-based autotuning via polygonal curve approximation. *J. Process Control* **17**(8), 641–652 (2007)
59. Rivera, D.E., Skogestad, S., Morari, M.: Internal model control 4: PID controller design. *Ind. Eng. Chem. Process Des. Dev.* **25**(1), 252–265 (1986)
60. Rovira, A.A., Murrill, P.W., Smith, C.L.: Tuning controllers for setpoint changes. *Instrum. Control Syst.* **42**(12), 67–69 (1969)

61. Shinskey, F.G.: Process control: as taught versus as practiced. *Ind. Eng. Chem. Res.* **41**(16), 3745–3750 (2002)
62. Silva, G.J., Datta, A., Bhattacharyya, S.P.: New results on the synthesis of pid controllers. *IEEE Trans. Autom. Control* **47**(2), 241–252 (2002)
63. Skogestad, S.: Simple analytic rules for model reduction and PID controller tuning. *J. Process Control* **13**(4), 291–309 (2003)
64. Swanda, A.P., Seborg, D.E.: Controller performance assessment based on setpoint response data. In: *Proc. ACC 1999*, San Diego, CA, pp. 3863–3867 (1999)
65. Taiwo, O.: Comparison of four methods of on-line identification and controller tuning. *IEE Proc., Control Theory Appl.* **140**(5), 323–327 (2006)
66. Tan, W., Liu, J., Chen, T., Marquez, H.J.: Comparison of some well-known PID tuning formulas. *Comput. Chem. Eng.* **30**(9), 1416–1423 (2006)
67. Toscano, R.: A simple robust PI/PID controller design via numerical optimization approach. *J. Process Control* **15**(1), 81–88 (2005)
68. Umland, J.W., Safiuddin, M.: Magnitude and symmetric optimum criterion for the design of linear control systems: what is it and how does it compare to the others? *IEEE Trans. Ind. Appl.* **26**(3), 489–497 (1990)
69. Veronesi, M., Visioli, A.: Performance assessment and retuning of PID controllers. *Ind. Eng. Chem. Res.* **48**(5), 2616–2623 (2009)
70. Vilanova, R.: IMC based robust PID design: tuning guidelines and automatic tuning. *J. Process Control* **18**(1), 61–70 (2008)
71. Vilanova, R., Alfaro, V.M., Arrieta, O., Pedret, C.: Analysis of the claimed robustness for PI/PID robust tuning rules. In: *Proc. 18th Mediterranean Conference on Control & Automation*, Marrakech, Morocco, pp. 658–662 (2010)
72. Visioli, A.: Optimal tuning of PID controllers for integral and unstable processes. *IEE Proc., Control Theory Appl.* **148**(2), 180–184 (2001)
73. Wang, Q.G., Fung, H.W., Zhang, Y.: PID tuning with exact gain and phase margins. *ISA Trans.* **38**(3), 243–249 (1999)
74. Yu, C.C.: *Autotuning of PID Controllers: A Relay Feedback Approach*. Springer, London (2006)
75. Ziegler, J.G., Nichols, N.B.: Optimum settings for automatic controllers. *Trans. Am. Soc. Mech. Eng.* **64**, 759–768 (1942)