Information Sciences 235 (2013) 3–35

Contents lists available at SciVerse ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

From model-based control to data-driven control: Survey, classification and perspective

Zhong-Sheng Hou [a,*], Zhuo Wang [b]

[a] Advanced Control Systems Laboratory of School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing 100044, PR China
[b] Department of Electrical and Computer Engineering, University of Illinois, Chicago, IL 60607-7053, USA

ARTICLE INFO

Article history:
Available online 4 August 2012

Keywords:
Data-driven control
Data-based control
Survey
Classification
Perspective

ABSTRACT

This paper is a brief survey on the existing problems and challenges inherent in model-based control (MBC) theory, and some important issues in the analysis and design of data-driven control (DDC) methods are here reviewed and addressed. The necessity of data-driven control is discussed from the aspects of the history, the present, and the future of control theories and applications. The state of the art of the existing DDC methods and applications are presented with appropriate classifications and insights. The relationship between the MBC method and the DDC method, the differences among different DDC methods, and relevant topics in data-driven optimization and modeling are also highlighted. Finally, the perspective of DDC and associated research topics are briefly explored and discussed.

1. Model based control theory

Since the late 1960s, modern control theory has been fully grown and developed. Its main branches, system identification, adaptive control, robust control, optimal control, variable structure control, and stochastic system theory, have been extensively used in industrial processes, aerospace, traffic systems, and other applications. However, the field of modern control theory still holds many challenging topics from both theoretical aspects and practical perspectives.

1.1. Modeling and identification

The introduction of the parametric state-space model by Kalman in 1960 and together with optimal control gave birth to the modern control theory, which is also called model-based control (MBC) [70,71]. Successful applications abounded, particularly in aerospace, where accurate models were available.

Modern control theory includes control theory for both linear and nonlinear systems. Typical linear control systems design methodologies include zero-pole assignment, LQR design, and robust control. For nonlinear systems, typical controller design methods include *Lyapunov*-based controller designs, backstepping controller design, and feedback linearization, etc. All these controller design methodologies are regarded as typical MBC system design. In applications of MBC theory, the first step is modeling the plant, or identifying the plant model, and then designing the controller based on the plant model obtained using the certainty equivalence principle with the faith that the plant model represents the true system. Therefore, the modeling and identification of the plant is necessary to MBC theory.

* Corresponding author. Tel.: +86 10 51688617.
E-mail address: zhshhou@bjtu.edu.cn (Z.-S. Hou).

0020-0255/$ - see front matter © 2012 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.ins.2012.07.014

Modeling a plant using first principles requires that the parameters be calibrated on-line or off-line using measured data. Identification theory may be used to develop a plant model within a model set that either covers the true system or approximates it in terms of bias and variance error on the identified model. Modeling, whether by the first principles or by identification from data, is an approximation of the true system, and some error is inevitable. Unmodeled dynamics always exist in the modeling process. Consequently, the closed loop control system, designed on MBC approaches which are thought to be unalterable, is inherently less safe and less robust because of these unmodeled dynamics [5–8].

In order to preserve the obvious advantages of MBC design while increasing robustness against model errors, much effort has been expended toward the development of robust control theory. Various ways of describing model errors in the configuration of closed loop systems have been considered. These include additive and multiplicative descriptions and the assumption on *priori* bounds on noise or modeling errors or uncertainties. However, the model uncertainty descriptions upon which robust control design methods have been based are not consistent with the methods delivered by physical mathematical modeling and identification modeling [86]. Modeling by first principles and by identification from data have very little to offer in terms of explicit quantification of errors. The main stumbling block in the application of model-based robust control design techniques is the lack of adequate, practical uncertainty descriptions [40].

It is very natural that, first spending a significant amount of efforts to obtain a very accurate model (including a model uncertainty set) for the unknown system by mechanism modeling or identification techniques, then computing a model based robust controller from this model and its uncertainty set. However, there are both practical and theoretical obstacles for the researchers who want to establish the perfect control theory. First, unmodeled dynamics and the robustness are a pair of inevitable twinborn problems and they cannot be solved simultaneously within the conventional MBC theoretical framework. Second, the more accurate the model is, the more effort or cost must be spent on the design of the control system. Until now, there has been no efficient way of producing an accurate plant model. Accurate modeling can be more difficult than control system design. Furthermore, there is no well-recognized means of addressing certain types of complexity, such as that observed in plants whose parameters vary quickly or whose structures change over time. If the system dynamics is of too high order, we cannot use it as a control system design model. Even if it were used as a model for control system design, this would typically lead to a controller with too high order. High-order controllers are not suitable to use in practice and reduction of model or controller order must be performed. Modeling an accurate high-order model to target high performance for a control system design, then having to perform a controller order reduction or model simplification for a low order controller, seems paradoxical. The last but not least is the persistence of excitation or persistently exciting inputs condition for modeling. Without the persistently exciting inputs, an accurate model cannot be produced. Without an accurate model, most model-based theoretical results of a closed loop control system scheme, such as stability and convergence, cannot be guaranteed as what they are claimed when they are used in practice [6–8,40].

1.2. Model based controller design

The certainty equivalence principle is a fundamental assumption in MBC theory. Model-based controller design may not work well if the plant model does not fall into the assumed model set. For this reason, designing a controller using an inaccurate model could leads to either bad performance or an unstable closed-loop system. Arbitrarily small modeling errors can lead to arbitrarily bad closed-loop performance [121]. For adaptive control, Rohr's counterexample has demonstrated that reported stable adaptive control systems based on some assumptions made about the system model may show certain unexpected behavior in the presence of unmodeled dynamics [108,109]. Rohr's counterexample is a wake-up call for researchers, who began to contemplate robustness issues in adaptive control.

Even when the model is accurate enough, the results of theoretical analysis, such as those covering stability, convergence and the robustness of a closed loop control system, proven by beautifully rigorous mathematical processes, are not always valuable if the additional assumptions made about the system are not correct. The architecture of MBC theory is shown in Fig. 1. This diagram shows that the system model and assumptions are the starting point for controller design, and also the destination of the MBC control system analysis. The key issue is that there exists a gap between the controlled plant and the system model built using assumptions, and this gap seems to cease to exist in controller design and control system analysis.

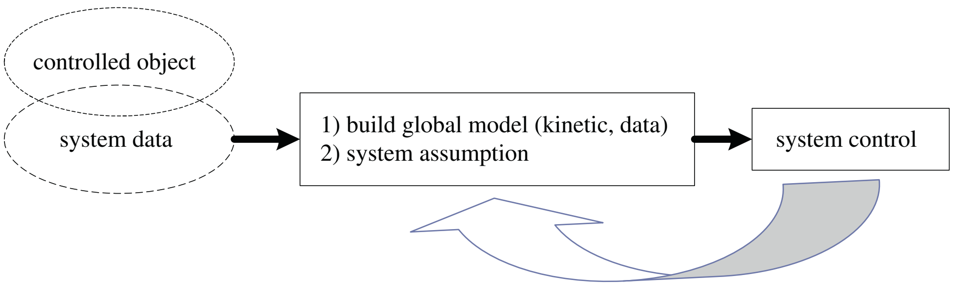

Fig. 1. Architecture of MBC theory.

Taking the most developed branch of MBC theory, adaptive control, as an example, adaptive control methods typically say that under assumptions A, B, C, D, and E, and with use of algorithm F, all signals remain bounded as time goes toward infinity, and then some specific result occurs. All this may be true and the conclusion may be valuable. However, it is not enough to give the user confidence, and the theorem cannot protect the plant with an adaptive controller connected. This is because the stated conclusion does not rule out the possibility that at some time before time goes to infinity, the particular controller connected to the plant will render the plant-controller closed loop unstable. Securing safe adaptive control is far from straightforward under this well-known model-based adaptive control scenario [6–8].

Typical nonlinear control system design methodologies include the *Lyapunov* based method, backstepping method, and feedback linearization. However, all these methods depend on an accurate model of the plant. As stated above, unmodeled dynamics are inevitable in modeling, so these controller design methods would lose their utility if the model is inaccurate. Because system model structure and kinetic equations are included in the controller, their accuracy can greatly influence system performance. The huge gap between MBC theory and practical application is a major obstacle to the use of elegant model-based controllers in practice, and a variety of problems may emerge during practical applications.

We could not draw any conclusion when the model is unavailable or those assumptions did not hold. The MBC control method starts and ends with the model. To some extent, it may be called model theory rather than control theory.

2. Data-driven control theory and related topics

With the development of information science and technology, practical processes such as those relevant to the chemical industry, metallurgy, machinery, electronics, electricity, transportation, and logistics, have undergone significant changes. These industries have production technologies and equipments in a large scale, and production processes have become more complex. Modeling processes using first principles or identification has become more difficult. For this reason, traditional MBC theory has become impractical for control issues in these kinds of enterprises. Furthermore, many industrial processes generate and store huge amounts of process data at every time instant of every day, containing all the valuable state information of process operations and equipments. Using these data, both on-line and off-line, to directly design controllers, predict and assess system states, evaluate performance, make decisions, or even diagnose faults, would be very significant, especially under the lack of accurate process models. For this reason, the establishment and development of data-driven control theory (DDC) are urgent issues both in theory and application.

The term "data-driven" was first proposed in computer science and has only recently entered the vocabulary of the control community. Until now, there have been a few DDC methods, but they are characterized by different names, such as data-driven control, data-based control, modeless control, MFAC (model-free adaptive control), IFT (iterative feedback tuning), VRFT (virtual reference feedback tuning), and ILC (iterative learning control). Strictly speaking, there are some differences between the terms data-driven control and data-based control. Data-driven control hints that the process is a closed loop control and its starting point and destination are both data, while data-based control means the process is an open loop control and only the starting point uses data.

Although the studies on DDC are still at in the embryonic stage, they have attracted a great deal of attentions within the control theory community. The Institute for Mathematics and Its Applications (IMA) of the University of Minnesota held a workshop titled "IMA Hot Topics Workshop: Data-driven Control and Optimization" in 2002, and 49 experts attended the workshop and 12 of them gave talks on this topic. In November 2008, the National Natural Science Foundation of China (NSFC) held a workshop titled "Data-based Control, Decision, Scheduling, and Fault Diagnostics," and 39 experts attended this meeting and 30 of them made speeches on this subject. Following this event, the first national key project of NSFC on DDC theory was granted to the first author of this paper. In June of 2009, a special issue with same title as above was published in ACTA AUTOMATICA SINICA [29]. It included 20 papers on these four subjects. In November 2010, the NSFC and Beijing Jiaotong University jointly held another workshop on this topic titled "International Workshop on Data Based Optimization, Control and Modeling," and 26 experts among the attendees addressed their concerns. The IEEE Transactions on Neural Networks, Information Sciences and IEEE Transactions on Industrial Informatics also launched their CFP for their special issue on this topic in 2010 and 2011, respectively, and the IEEE Transactions on Neural Networks has published the special issue in December of 2011 and the first author of this paper was one of the guest editors [30]. Finally, the Chinese Automation Congresses, held by the Chinese Automation Association in 2009 and 2011, also focused this as a hot topic in one of its six main forums.

2.1. Definition of DDC

There are three literal definitions found by searching on the internet till now. They are as follows:

Definition 1 [60]. Data-driven controls are the control theories and methods in which the controller is designed directly using on-line or off-line I/O data of the controlled system or knowledge from the data processing without using explicit or implicit information of the mathematical model of the controlled process, and whose stability, convergence, and robustness can be guaranteed by rigorous mathematical analysis under certain reasonable assumptions.

Definition 2 [133]. Data-driven control design is the synthesis of a controller using data measured on the actual system to be controlled without explicit use of (non) parametric models of the system to be controlled during adaptation.

Definition 3 [138]. Measured data are used directly to minimize a control criterion. Only one optimization in which the controller parameters are the optimization variables is used to calculate the controller.

From these three definitions, we can draw several conclusions:

First, controller design in DDC methodologies depends only on the measured input–output data of the controlled plant. The architecture of the DDC methodologies is shown in Fig. 2. The features of this system are that the DDC controller design and control system analysis are both performed using only the measurement I/O data of the closed loop control system. The plant model disappears and no longer dominates the process. The measurement I/O data of the closed loop control system are the starting point of the controlled problems and also the end criteria for control system performance.

Second, in Definition 1, the DDC controller design directly uses the plant measurement I/O data, not including any information on the dynamics or structure of the controlled system. In Definition 2, the DDC controller design may include the implicit use of structural information of the controlled plant, and only the adaptive control is considered. In Definition 3, the DDC controller structure is predetermined, and only the method with an off-line controller parameter tuning is included.

Finally, the important issues in practical applications for a control system are the stability, convergence, and robustness. This is also true for DDC methodologies. All these concerns should also be addressed in suitable ways for DDC methods when they are used in practice.

Summarizing these three definitions above, we propose a more general definition of DDC control, which covers all the above mentioned definitions.

Definition 4. Data-driven control includes all control theories and methods in which the controller is designed by directly using on-line or off-line I/O data of the controlled system or knowledge from the data processing but not any explicit information from mathematical model of the controlled process, and whose stability, convergence, and robustness can be guaranteed by rigorous mathematical analysis under certain reasonable assumptions.

Three key points are emphasized in this definition. They are the direct use of the measurement I/O data, data modeling rather than first principles modeling or identified modeling, and the guarantee of the results of theoretical analysis. Simply speaking, it is a kind of methods directly from data to controller input. In other word, let the data speak.

In following discussions, we focus on the DDC methods that fit Definition 4, excepting methods that implicitly use dynamic model and structure information. This is because DDC methods that implicitly use mathematical models of the controlled plant have no essential difference from MBC theories or methods in either design or analysis.

2.2. Control objects of DDC

Control system consists of two main parts, the controlled object and the controller. Real-world controlled plants can be cataloged into the following four classes, shown in Fig. 3.

C1. Those for which accurate mathematical models obtained from the first principles or the identification are available.
C2. Those for which first principles or identification-based mathematical models are roughly accurate with moderate uncertainties.
C3. Those for which first principles or identification-based mathematical models are complicated with too high order and too much nonlinearity, etc.
C4. Those for which first principles or identification-based mathematical models are difficult to establish or unavailable.

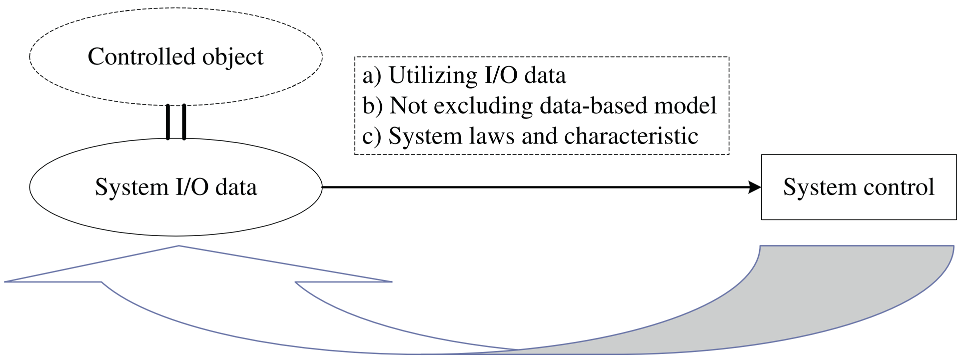

Fig. 2. Architecture of DDC methodologies.

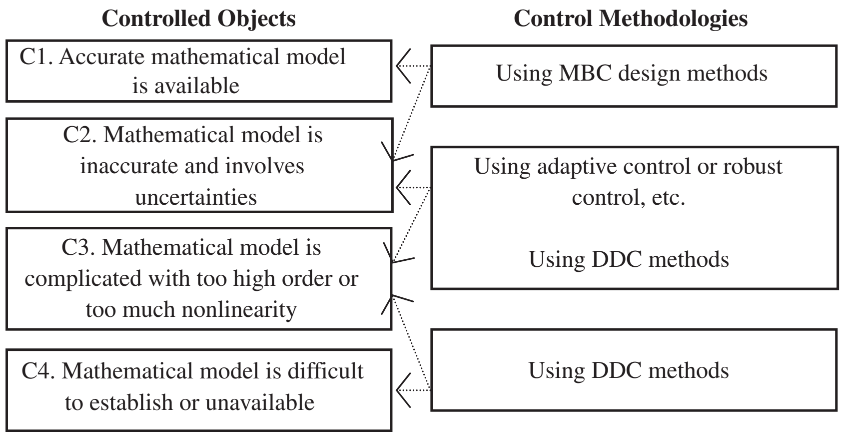

Fig. 3. Controlled objects of DDC.

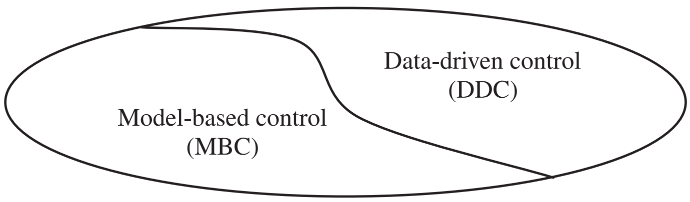

Fig. 4. Perfect control theory.

Generally speaking, classes C1 and C2 have been well addressed by modern control theory, also called MBC theory. For C1, we have many well-studied approaches to deal with both linear and nonlinear systems, such as zero-pole assignment, *Lyapunov* controller design methods, backstepping design methods, and feedback linearization. For C2, both adaptive and robust control have been well developed to focus on issues that occur when uncertainty can be parameterized or the model error bound is not very big and can be assumed to be known. Although many well-developed modern control branches have been established to address these two classes of controlled objects, there are still many open problems for us to study.

For C3, if the model is too complex, consisting of hundreds or thousands of equations and state variables, then it cannot be used for controller design. Very complex class C3 systems can be reclassified as C4. If the first principles or the identified mathematical models are available, accurate, and suitable to controller design, then high order or high levels of nonlinearity must produce controllers with high order and high nonlinearity. Controllers that are too complex could be difficult or costly to use. Faults can be generated too easily. So for this kind of system control problems, the model reduction or controller reduction process is inevitable. Usually, mathematical models, that are too complex, are not suitable for controller design because of the difficulty of controller designing and of control system properties analysis. For C4, there are currently no known methods that can address the relevant control problems efficiently.

Fewer than half of these four classes of controlled plants are well addressed. The other half will be the controlled objects of the DDC because the measurement I/O data is always possible. In other words, if the system model is unavailable or involves large uncertainties, then the DDC method should be considered.

2.3. Status of DDC in control theory

Control theory should include two parts, as shown in Fig. 4. One is MBC theory and the other is DDC theory. The reason why we take this opinion is that, the MBC theory can only solve the problems when reliable mathematics models are available and the uncertainties are constrained within a known moderate bound. In other word, only classes C1 and C2 are studied in the MBC framework. What are the control methods for the classes C3 and C4? The DDC control methods should be the inevitable alternative choice. Based on the observation, the perfect control theory should include all methods capable of dealing with all four classes of controlled objects.

2.4. Fundamental differences between MBC and DDC approaches

MBC control and DDC control are the two parts of control theory, and the ultimate objective should be the same, that is, to design the controller drives the output of the controlled plant to track the desired signal or to satisfy the designed target. The main difference between MBC and DDC is that, one is model-based control system design approach since a reasonable model is available, the other is the data driven control system design approach since there is no reliable mathematics model. Due to this main difference, the DDC has many inherent features:

(1) The controller in DDC approaches does not explicitly include any parts or the whole of the plant model. For this reason, it has overcome the dependence on the plant model for the design of control systems.
(2) The stability and convergence conclusions of DDC approaches do not depend upon the accuracy of the model, excepting the DDC control methods implicitly using the system dynamics and structure information, such as, direct adaptive control, sub-space predictive control, etc., which is the main stumbling block for the applications of the MBC theory.
(3) The most outstanding point of DDC approaches is that the twinborn problem of unmodeled dynamics and robustness in traditional MBC theory do not exist under DDC framework.

The main distinction between MBC and DDC is whether the controller is not designed based on the system model or I/O data only, in other word, whether the system dynamic model is involved in the designing of the controller. If the system model is involved in controller, it is a MBC method; otherwise it is a DDC method. From this point of view, we can conclude that some of the neural-network-based control methods, fuzzy control methods, and many other intelligent control methods are DDC methods, such as, the NN based control methods with NN as a controller directly approximating the inverse of the system. Some of them are not, in which the NN or the fuzzy rule or the knowledge describing the systems act as a system model, and the NN and the fuzzy rule or the knowledge is involved in the controller [20, 127].

2.5. Remarks on DDC

Because DDC is still developing, there are many important topics that must be addressed.

(1) The DDC methods implicitly utilizing system model or structural information, such as direct adaptive control and subspace-identification-based predictive control methods, do not look the same as model-based control methods apparently, but the controller design, stability, and convergence analysis are still the same as in MBC approaches. The conclusions of this kind of control system are still relevant to model accuracy (or structural information, including order and time-delay), and the corresponding traditional robustness problems still exist. For this reason, there is no essential difference between DDC methods that implicitly use these information and MBC methods.
(2) Theoretically speaking, the control problems caused by system time-varying parameters and time-varying model structures are challenging for the MBC methods but not with DDC approaches. There is no meaning whether the system parameters or model structure are time-varying or not in the I/O data level of the system measurement since the controller in DDC approach is designed only using I/O data. Thus, the difficulties in dealing with the time-varying problems of system structure or parameter or delay, which challenges the MBC methods, disappear in the DDC methods.
(3) On the data level, information cannot be clearly classified as linear or nonlinear. An ideal DDC approach should have the ability to deal with the control problems both linear and nonlinear systems uniformly. Iterative learning control, model-free adaptive control, and SPSA-based DDC, shown in Section 3 of *Classification and Brief Survey on the Existing DDC Approaches* for the details, are good examples for this remark.
(4) Robustness in the traditional sense does not exist in DDC approaches excepting the DDC methods that implicitly use dynamic model of controlled plant, and neither the unmodeled dynamics due to only measurement I/O data is involved in the controller design. However, system robustness is a universal concept. A new definition of robustness must be coined for DDC.
(5) It is expected that there is no great distinction between simulation results in a lab and in field applications when the DDC approach is implemented in practice since the DDC method is only depended on the measurement I/O data. Hence the huge gap between control theory and application vanishes.
(6) DDC theory should have an open framework and could cooperate with other control theories and methods. The relationship between DDC and MBC should be complementary or mutual rather than exclusive. DDC and MBC methods can both work in a modularized manner because each method has its own advantages and disadvantages. Different DDC methods such as ILC and PID control, should also benefit each other. General speaking, the more accurate information about the system we used, the better performance of a designed control system we could expected. The promotion of efficient use of existing accurate information by DDC approach is one of the problems that remains to be solved.
(7) DDC is not an omnipotent control method (no such methods exist). Any control method may be proposed for a given class of systems. Certain assumptions must be made before the stability, convergence, and robustness of DDC approaches can be analyzed. However, the assumptions required for DDC would be different from those required for MBC.
(8) The DDC should be considered when any of the following situations occur: (a) The model of the controlled system is unavailable. (b) The uncertainties and varieties of system structure are serious and difficult to express in a unified mathematical model. (c) Modeling the plant is difficult or control performance using MBC methodologies is unacceptable. (d) The mathematical model is too complex for controller design.

(9) From a controller design point of view, the closed loop measurement I/O data includes both on-line and off-line data. The online data are the system I/O data within a finite time window. Different control methods may involve data time windows of different lengths. Adaptive control uses the I/O data within time window length equaling to the system orders. Typical iterative learning controllers use data from the current and previous iterations within a given past time or iteration interval. PID controllers use data from the current instant and two previous instants. On-line data reflects the current system state timely. The control system can capture and adapts the variations if the on-line data fully used. Off-line data is relative to online data. In MBC, the off-line data is used to build dynamic models of the controlled systems. Once the model is built, off-line data is no longer used. However, off-line data contains a great deal of information with respect to system operation, and the potential rules and patterns can be found through processing and mining. If they are used effectively, better performance can be expected.

2.6. Trends in the development of control theory

This section discusses the motivations behind the development of DDC from aspects of history, present and future of control theory and applications.

In the history of system control, control theory has been developed from model-free tuning method, for instance, PID control, to the MBC theory, such as, transfer function model based classical control, and state space model-based modern control, then to knowledge or rule based intelligent control. This developing routine can be imagined as a helix flowing from model free, to model based, to deviation from. What we can image next logically would be the DDC.

From the integrity of control theory, the existing control methods can be divided into three categories: (a) Control methods designed depending on system model, such as aerospace control, optimal control, linear and nonlinear control, large-scale system decomposition and coordination control, and pole placement. (b) Control methods designed partially depending on system model, such as robust control, sliding-mode variable-structure control, adaptive control, fuzzy control, expert systems, neural network control, and intelligent control. (c) Control methods designed depending on system I/O data, such as PID control and MFAC. DDC enhances the integrity of control theory.

From the perspective of control theory research, the problems of unmodeled dynamics and robustness are inevitable in MBC theory. This can cause unsafe controllers and huge gaps between theoretical results and applications, consequently block the healthy growth of MBC theory. Modeling more accurate high-order and complex nonlinear dynamic systems could leads to another paradox. High-order and highly nonlinear controlled systems inherit controllers with high order and high degrees of nonlinearity. These controllers are difficult to design, use, and maintain. Usually, either model reduction or controller reduction is needed to reduce the complexity of the control system. DDC theory may be an alternative way to deal with these paradoxes.

From perspective of practical applications, low-cost, easy-to-install control techniques and automation equipment are a priority for many industrial processes. However, the modeling of a plant requires specific skills and mathematical procedures. Most engineers are not capable of this type of work, so high-level experts or researchers are needed. Taking the batch process as an example, it is impossible to model all batches for all products. For complex systems, it is also impossible to build a global model because of internal complexities and external disturbances. Even modeling a locally accurate model is not easy. Sometimes it is impossible. MBC theory is usually not practical for industrial process. Large amounts of data and scarce knowledge are common problems in complex system control and management. Finally, most control engineers in most fields are unable to deal with complex mathematics and identification theory, which is another obstacle to the application of MBC theory. Practical demands require DDC theory and techniques.

3. Classification and brief survey on the existing DDC approaches

So far, there are over 10 kinds of different DDC methods. Sorted according to the type of data usage, these methods can be summarized as three classes: those based on on-line data; those based on off-line data, and those based on both (hybrid DDC). If sorted by method of controller structure design, they can be divided into two classes: DDC methods with pre-specified fixed controller structures and DDC methods with unknown controller structures. We have briefly surveyed existing DDC methods according to these two observations.

3.1. DDC classification according to the use of the measurement I/O data

3.1.1. On-line data based DDC

3.1.1.1. SPSA-based DDC methods (SPSA). A direct controller approximation method based on SPSA (simultaneous perturbation stochastic approximation) was proposed by Spall in [124]. This method uses only closed-loop measured data rather than a mathematical model of the controlled plant to tune the parameters of the controller [125–129], and the diagram is depicted as Fig. 5.

The SPSA-based control methods assume that the nonlinear dynamics of the controlled plant are unknown. The controller serves as a function approximator whose structure is fixed. The parameters are tunable. The approximator may be a neural network (NN), polynomial, or other type of approximator. For example, if a multilayer feed-forward NN is selected as the

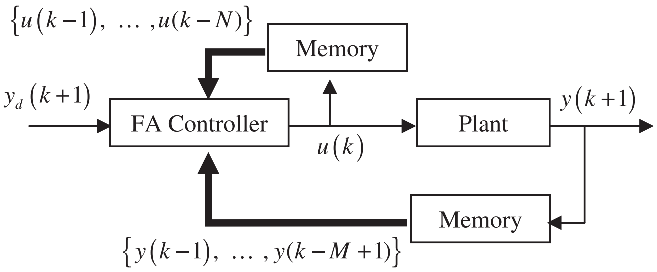

Fig. 5. SPSA-based DDC method.

controller, the number of layers and nodes are determined and then the tunable weighted coefficient connected $\boldsymbol{\theta}$ is the parameter of the controller. The control inputs of the NN are the control signals, system outputs within a fixed time window before the current instant and the one step ahead desired output, i.e., at time instant k, the input of NN are as follows:

$$\begin{aligned} &y(k), y(k-1), \ldots, y(k-M+1), \\ &u(k-1), u(k-2), \ldots, y(k-N), y_d(k+1) \end{aligned} \tag{1}$$

and $u(k)$ is the output of the controller. Where $y(k)$ and $u(k)$ are the output and input of the controlled plant at instant k, $y_d(k+1)$ is the desired output of the controlled plant at instant $k+1$, and M and N are the lengths of the time window of data. The aim of the controller design is, at each time instant, to find an optimal controller parameter $\boldsymbol{\theta}^*$, minimizing the control performance index.

$$J_k(\boldsymbol{\theta}_k) = E[(y(\boldsymbol{\theta}_k, k+1) - y_d(k+1))^2]. \tag{2}$$

To solve this problem, the mathematical model of a controlled plant must be known ahead of time, but here, this information is unknown, so the traditional optimal techniques cannot be used here due to lack of knowledge of $\partial y(\boldsymbol{\theta}_k, k+1)/\partial \boldsymbol{\theta}_k$. This leads to SPSA algorithm is used to address this optimal problem. In this method, recursive formula (3) is used to estimate the sequence $\{\boldsymbol{\theta}_k\}$.

$$\hat{\boldsymbol{\theta}}_k = \hat{\boldsymbol{\theta}}_{k-1} - a_k \hat{\mathbf{g}}_k(\hat{\boldsymbol{\theta}}_{k-1}), \tag{3}$$

where, $\hat{\boldsymbol{\theta}}_k$ is the estimation at current iteration, a_k is a scale parameter, and $\hat{\mathbf{g}}_k(\hat{\boldsymbol{\theta}}_{k-1})$ is estimation of simultaneous perturbation $\mathbf{g}_k(\hat{\boldsymbol{\theta}}_{k-1})$. The lth component of $\hat{\mathbf{g}}_k(\hat{\boldsymbol{\theta}}_{k-1})$ is calculated as follows:

$$\hat{g}_{kl}(\hat{\boldsymbol{\theta}}_{k-1}) = \frac{\widehat{J}_k^{(+)} - \widehat{J}_k^{(-)}}{2c_k\Delta_{kl}}, \tag{4}$$

where, $l = 1, 2, \ldots, L$, and L denote the number of the parameters of the controller, $\widehat{J}_k^{(\pm)}$ is the estimations of $J_k(\hat{\boldsymbol{\theta}}_{k-1} \pm c_k \boldsymbol{\Delta}_k)$, i.e., $\widehat{J}_k^{(\pm)} = \left(y_{k+1}^{(\pm)} - y_d(k+1)\right)^2$, calculated with $y_{k+1}^{(\pm)}$. $y_{k+1}^{(\pm)}$ is the measurement of output when the input is $u_k^{(\pm)}$, $u_k^{(\pm)}$ is the input generated from the controller when the parameter is set as $\boldsymbol{\theta}_k = \hat{\boldsymbol{\theta}}_{k-1} \pm c_k \boldsymbol{\Delta}_k$, and $\boldsymbol{\Delta}_k = (\Delta_{k1}, \Delta_{k2}, \ldots, \Delta_{kL})^T$ is a stochastic vector. Usually Δ_{kl} is with independent bounded symmetric distribution. c_k is a scale coefficient usually be considered a constant or sequence approaching zero. From the description above, only two closed loop experiments are needed in the iteration before the estimation $\hat{\mathbf{g}}_k(\hat{\boldsymbol{\theta}}_{k-1})$ of $\mathbf{g}_k(\hat{\boldsymbol{\theta}}_{k-1})$ can be obtained by using the measurement data. No information regarding the controlled plant is needed in the whole process.

The sufficient condition of the convergence of the SPSA algorithm is given in the literature. If all the conditions hold, and $\boldsymbol{\theta}^*$ exists, then, $(\hat{\boldsymbol{\theta}}_k - \boldsymbol{\theta}^*)$ almost surely approaches zero as k approaches infinity ($\boldsymbol{\theta}_k^*$ approaches $\boldsymbol{\theta}^*$).

In the SPSA-based control algorithm, no assumption is made regarding the controlled plant. In this way, it can deal with the nonlinear controlled plant. However, there is a drawback: The stochastic perturbation to the parameter may lead to wasted product if it is used in practice. The convergence rate is slow and not suitable for the controlled plants whose parameters vary quickly over time. Some improvements in the convergence rate of SPSA algorithm can be found in [126,128,129], and the applications of the SPSA based model free control method in active noise control systems, traffic control systems, and industrial control systems can be found in [123,126].

Similar to IFT, the SPSA-based control algorithm also requires a test signal. Nevertheless, IFT requires two groups of experimental data with length N, and the latter only requires collecting two groups of experiment data with length 1. Both methods are on-line gradient estimate algorithm based DDC methods.

3.1.1.2. Model-free adaptive control (MFAC). Model-free adaptive control was first proposed in 1994 by Hou [52]. The essential idea is that, using an equivalent dynamic linearization data model with a novel concept called pseudo partial derivative at every current operation point to replace the general discrete time nonlinear system, then estimate the pseudo partial derivative on-line solely using the input and output data from the controlled plant, finally design the model-free adaptive control strategy for a class of nonlinear discrete-time systems [53–55,64,65].

The general discrete time SISO nonlinear system can be described as follows:

$$y(k+1) = f(y(k), \ldots, y(k-n_y), u(k), \ldots, u(k-n_u)), \tag{5}$$

where, $y(k)$ and $u(k)$ are the output and input of the controlled plant at instant k, n_y and n_u are the unknown order of output and input, and $f(\cdots)$ is an unknown nonlinear function.

If a system satisfies the generalized *Lipschitz* condition, that is, $|\Delta y(k+1)| \leqslant b|\Delta u(k)|$ or similar conditions for any fixed k and $|\Delta u(k)| \neq 0$, then (5) can be expressed as following three kinds of dynamic linearization data models, and the pseudo partial derivative is uniformly bounded for any fixed k.

(1) Compact-form dynamic linearization data model:

$$y(k+1) = y(k) + \phi(k)\Delta u(k),$$

where $\phi(k)$ is the *pseudo partial derivative* of the controlled system at time instant k.

(2) Partial-form dynamic linearization data model:

$$\begin{aligned} &y(k+1) = y(k) + \boldsymbol{\phi}^T(k)\Delta \boldsymbol{u}(k), \\ &\boldsymbol{\phi}(k) = [\,\phi_1(k) \quad \cdots \quad \phi_L(k)\,]^T, \\ &\Delta \boldsymbol{u}(k) = [\,\Delta u(k) \quad \cdots \quad \Delta u(k-L+1)\,]^T, \end{aligned}$$

where, $\boldsymbol{\phi}(k)$ is *pseudo partial derivative vector* of the controlled plant, and L is the *control input linearization level constant*.

(3) Full-form dynamic linearization data model:

$$\begin{aligned} &y(k+1) = y(k) + \boldsymbol{\phi}^T(k)\Delta \boldsymbol{u}(k), \\ &\boldsymbol{\phi}(k) = \left[\,\phi_1(k) \quad \cdots \quad \phi_{L_u}(k) \quad \phi_{L_u+1}(k) \quad \cdots \quad \phi_{L_y+L_u}(k)\,\right]^T, \\ &\Delta \boldsymbol{u}(k) = \left[\,\Delta u(k) \quad \cdots \quad \Delta u(k-L_u+1) \quad \Delta y(k) \quad \cdots \quad \Delta y(k-L_y+1)\,\right]^T, \end{aligned}$$

where, L_u and L_y are the pseudo order of the system input and output, respectively, and $\boldsymbol{\phi}(k)$ is *pseudo partial derivative vector* of the controlled system.

Compared to other linearization methods for nonlinear function, the proposed dynamic linearization method has the following features:

(1) It does not require a mathematical model, order, or time delay of the controlled plant.
(2) It is an equivalent dynamic linearization data model rather than an approximation model.
(3) It is an extension of finite impulse model of a linear time-invariant system to a nonlinear system.
(4) The dynamic linearization model, having time-varying incremental form with very simple structure and very few parameters, is a virtual data model for the purpose of controller design rather than a first principles model or transfer function model. The introduction of pseudo orders can avoid the high order controller design. Usually, high order controlled plants lead to high order controllers, which increase the computation burden and difficulty of implementation when applied in practice.
(5) For a nonlinear system, the pseudo partial derivative is not unique and is a time-varying parameter, so the dynamic linearization data model is not unique. The differences among these three kinds of dynamic linearization data models are the complexity. In compact-form dynamic linearization data model, all nonlinear properties and estimation error are fused into the scalar pseudo partial derivative. In this way, the dynamic behavior of the pseudo partial derivative may become complicated. If an estimation algorithm fails to estimate the complex dynamic behavior of it, the partial-form dynamic linearization data model or full-form dynamic linearization data model should be selected. This is because all the components involved in a pseudo partial derivative vector share the complex dynamic behavior of the system, and the high-quality estimation can be expected when the same estimation algorithm is used.
(6) The pseudo partial derivative behavior of MFAC may not be sensitive to the variations of the parameter, structure, or delay of the controlled system. However, these are explicit in first principles models and in the transfer function models based control system design, and this problem is hard to handle.
(7) This dynamic linearization method can be easily extended to cases of MISO and MIMO nonlinear systems [52,54,65].
(8) The linearization data model itself is a dynamic linear system in data level. Thus, all the skills and techniques in MBC theory can be borrowed and introduced into the analysis and design of MFAC. MFAC has a series of control system design methods and analysis methods, and has many advantages compared with the other DDC methods.

With help of the dynamic linearization technique, the controller design could be very easy. Take the compact form linearization for example, a nonlinear system is transformed into a linear time-varying data system, and then, the MFAC control scheme based on compact form dynamic linearization can be derived by using the weighted one step-ahead cost function as follows.

$$u(k) = u(k-1) + \frac{\rho_k \hat{\phi}(k)}{\lambda + |\hat{\phi}(k)|^2}(y_d(k+1) - y(k)), \tag{6}$$

$$\hat{\phi}(k) = \hat{\phi}(k-1) + \frac{\eta_k \Delta u(k-1)}{\mu + \Delta u(k-1)^2}(\Delta y(k) - \hat{\phi}(k-1)\Delta u(k-1)), \tag{7}$$

$$\hat{\phi}(k) = \hat{\phi}(1), if |\hat{\phi}(k)| \leqslant \varepsilon, or |\Delta u(k-1)| \leqslant \varepsilon, \tag{8}$$

where, ρ_k and η_k are sequences of step length, λ and μ are weighted factors, and ε is a small positive constant.

According to (6)–(8), controller design is not relevant to the mathematical model or the order of the controlled plant. The scheme can deal with the adaptive control of parameter time-varying and structure time-varying nonlinear system. Because only one on-line parameters need to be tuned in the above simplest control scheme, the burden of computation can be neglected. The pseudo partial derivative $\phi(k)$ varies slowly over time, so any traditional time-varying parameter estimated algorithm can be used. The reset algorithm (8) can strengthen the tracking ability of the estimated algorithm. For the complex nonlinear system with a fast time-varying pseudo partial derivative, if the control performance and the robustness of the model free adaptive control system based on compact-form dynamic linearization is not satisfied, then the other two linearization techniques should be used. As well known, unmodeled dynamics and robustness problems are inevitable in model-based control theory and methodology, while MFAC is based on the I/O data model obtained from the accurate equivalent dynamic linearization of the controlled plant, so the conventional unmodeled dynamics and robustness problems vanish simultaneously [23,24,63].

The proof of stability and convergence of a regulation problem using MFAC based on compact form dynamic linearization and partial form dynamic linearization have been proposed in works [53–55,64,65].

In [35,36], the essential idea of MFAC is introduced into the iterative axis in order to deal with a repeatable control task, so a model-free adaptive iterative learning control is obtained. In [37], the optimal selection of parameter of MFAC was considered. In [55,131,153], the predictive control and predictive functions control of a nonlinear system based on MFAC are discussed. The robust issues of MFAC schemes are considered in [23,24,63].

One of the most outstanding characteristics of MFAC is that it can collaboratively work with other MBC or DDC control methods, as shown in Section 4. So far, the effectiveness of the method has been verified in practical applications [28,34,37,39,61,64,87,88,131,140,153].

The MFAC control method is still developing. There are many open problems in MFAC, such as how to select the length of the PPD vector, how to check the generalized *Lipschitz* condition, and how to prove the stability and convergence of the tracking problems.

3.1.1.3. Unfalsified control (UC) methodology. Unfalsified control was proposed by Safonov in 1995, and it recursively falsifies control parameter sets that fail to satisfy the performance specification [110]. The whole process is designed by the I/O data rather than the mathematic model of controlled plants. UC is a type of switching control method differing from the traditional switching control. UC can falsify the controller, which cannot stabilize the control system before being inserted into the feedback loop, so the transient performance is relatively good. The main elements of UC are as follows: an invertible controller candidate set, cost-detectable performance specifications, and the switching mechanism.

A simple case of UC is depicted in Fig. 6, where P is an unknown controlled plant. The invertible time-invariant controllers $C_1, C_2, \ldots, C_N$ belong to controller set $\mathbf{C}$. At the current instant k, the I/O data of the controlled plant $\{(u(\tau), y(\tau)) | \tau \in [0, k-1]\}$, which is collected within the time interval $[0, k-1]$, is used to evaluate the controller C_j, $j = 1, 2, \ldots, N$, and then the optimal one is selected as the active controller at instant k. It should be noted that the performance of C_j is evaluated before insertion into the closed loop system. With the measured data $u(\tau)$, $y(\tau)$, the fictitious reference signal $\tilde{r}_j(\tau)$ of controller C_j can be expressed as follows:

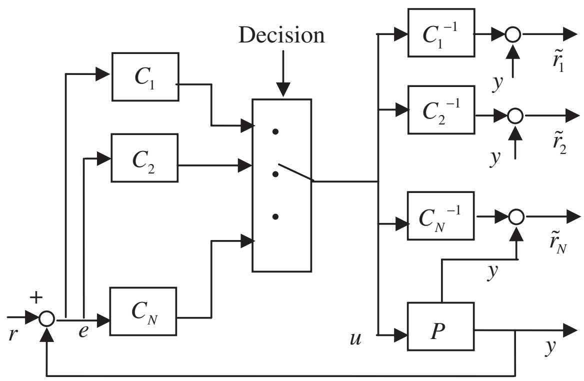

Fig. 6. Simple diagram of an unfalsified control method.

$$\tilde{r}_j(\tau) = C_j^{-1}(u(\tau)) + y(\tau). \tag{9}$$

The controller C_j is evaluated by using the control performance $J(u, y, \tilde{r}_j)$ and data set $\{(u(\tau), y(\tau), \tilde{r}_j(\tau))|\tau \in [0, k-1]\}$. A typical controller performance specification is as follows:

$$J_j(k) = J(u, y, \tilde{r}_j, k) = \max_{\tau\in[0,k]} \frac{\|u(\tau)\|^2 + \|\tilde{r}_j(\tau) - y(\tau)\|^2}{\|\tilde{r}_j(\tau)\|^2 + \alpha}, \quad \alpha > 0. \tag{10}$$

Form (10), using $J_j(k)$, j = 1, 2, ..., N, and set $j^*(k) = \arg\min_{j=1,2,\ldots,N} J_j(k)$, $C_{j^*(k)}$ is chosen as the active controller of the closed loop system at instant k. At each instant, the UC scheme eliminates all $N-1$ controllers except for $C_{j^*(k)}$ and casts the $C_{j^*(k)}$ into the control system.

In UC, in order to select an appropriate replacement for the falsified controller, a switching mechanism is needed. For a finite controller set, the scheme has to evaluate all the controller candidates and select the optimal controller switching into the closed loop system [111]. In literatures [134,141], two switching mechanisms are proposed for infinite parametric controller set. In [134], an ellipsoidal UC was proposed. In this algorithm, the region of controllers that are unfalsified, the unfalsified set, is described by an ellipsoid. Due to the combination of the performance requirement and controller structure, the approximate update of the unfalsified set can be computed analytically, resulting in a computationally cheap algorithm. In [141], a gradient-based parameter iterative selection algorithm was proposed. In this method, the controller candidate set was a fixed structure controller $C(\boldsymbol{\theta})$, where $\boldsymbol{\theta}$ was the tunable parameter. The main idea of this UC algorithm was to determine whether the parameter could satisfy the performance specifications in the direction of the negative gradient. The details are as follows:

First, the fictitious reference signal was calculated according to (9), then $C(\boldsymbol{\theta})$ served as the UC controller if and only if $J(\boldsymbol{\theta}, \tau) \leqslant 0$, $\forall \tau \in [0, t]$, where,

$$J(\boldsymbol{\theta}, \tau) = -\rho(\tau) + \int_0^{\tau} T_{spec}(\tilde{r}(\boldsymbol{\theta}, \zeta), y(\zeta), u(\zeta))d\zeta, \tag{11}$$

$u(\zeta)$, $y(\zeta)$, ($\zeta \in [0, t]$) is the historical data measured and $\tilde{r}(\boldsymbol{\theta}, \zeta)$ is the fictitious reference generated by controller $C(\boldsymbol{\theta})$.

Second, if the current controller, which must be active in the closed loop control system, is falsified, then the parameter is updated toward the direction of $-\nabla J(\boldsymbol{\theta}, t)$ in order to achieve the performance requirements:

$$\frac{d\boldsymbol{\theta}}{dt} = -\gamma \nabla J(\boldsymbol{\theta}, t), \tag{12}$$

where, γ is a pre-designated constant coefficient; $\nabla J(\boldsymbol{\theta}, t)$ is the gradient of $J(\boldsymbol{\theta}, t)$ with respect to $\boldsymbol{\theta}$, and it can be calculated as follows:

$$\nabla J(\boldsymbol{\theta}, t) = \left[\frac{\partial J(\boldsymbol{\theta}, t)}{\partial \theta_1}, \frac{\partial J(\boldsymbol{\theta}, t)}{\partial \theta_2}, \ldots, \frac{\partial JJ(\boldsymbol{\theta}, t)}{\partial \theta_n}\right]^T = \int_0^t \frac{\partial T_{spec}(\tilde{r}(\boldsymbol{\theta}, \zeta), y(\zeta), u(\zeta))}{\partial \tilde{r}} \nabla \tilde{r}(\boldsymbol{\theta}, \zeta) d\zeta \tag{13}$$

where, $\nabla \tilde{r}(\boldsymbol{\theta}, \zeta)$ is the gradient of $\tilde{r}(\boldsymbol{\theta}, \zeta)$ with respect to $\boldsymbol{\theta}$. Because $\tilde{r}(\boldsymbol{\theta}, \zeta) = C^{-1}(\boldsymbol{\theta})u(\zeta) + y(\zeta)$, then the following is true:

$$\begin{aligned} &\nabla \tilde{r}(\boldsymbol{\theta}, \zeta) = -C^{-1}(\boldsymbol{\theta}) \nabla C(\boldsymbol{\theta}) C^{-1}(\boldsymbol{\theta}) u(\zeta), \\ &\nabla C(\boldsymbol{\theta}) = [\tfrac{\partial C(\boldsymbol{\theta})}{\partial \theta_1}, \tfrac{\partial C(\boldsymbol{\theta})}{\partial \theta_2}, \ldots, \tfrac{\partial C(\boldsymbol{\theta})}{\partial \theta_n}]^T. \end{aligned} \tag{14}$$

Finally, the adaptive update law of the parameter is as follows:

$$\frac{d\boldsymbol{\theta}}{dt} = \begin{cases} \gamma \int_0^{\tau} \frac{\partial T_{spec}(\tilde{r}, \zeta)}{\partial \tilde{r}} C^{-1}(\boldsymbol{\theta}) \nabla C(\boldsymbol{\theta}) C^{-1}(\boldsymbol{\theta}) u(\zeta) \times d\zeta, & \text{if } J(\boldsymbol{\theta}, \tau) > 0, \ \forall \tau \in [0, t], \\ 0, \quad \text{if } J(\boldsymbol{\theta}, \tau) \leqslant 0, \ \forall \tau \in [0, t]. \end{cases} \tag{15}$$

Many studies have been published on this topic. In [15], the unfalsified adaptive switching supervisory control was extended to the case of systems whose dynamics are subject to infrequent but large variations. In [16], the input–output stability of the unfalsified adaptive switching control system in a noisy environment was analyzed, and the issue of equivalence among different input–output stabilities was discussed. In [14], the multi-model based method was embedded in the unfalsified adaptive switching control schemes. In [135], the ellipsoidal unfalsified control for the MIMO system is discussed. In [142], a safe adaptive control based on switching among the candidate controllers and the stability of the system are provided if there exists a controller that can stabilize the controller plant in the candidate controller set. In [136], the ellipsoidal unfalsified control was applied to a motion system. In [112], UC was applied to the missile guidance, double-connecting rod robot arms, and industrial processes control. However, some issues still remain: determining the controller candidate set and the invertible controller and determining the performance specifications capable of reflecting the states of various stable and unstable cases.

3.1.2. Off-line data based DDC methods

3.1.2.1. PID control method. Many articles discuss PID control and PID-related methods, and the PID is widely used in practical applications [10–12,81,143]. Until now, 95% of control methods utilized in practical industrial process is PID, although thousands of beautiful and elegant control methods are published every year [117]. PID and its tuning methods, which was proposed by Ziegler and Nichols, may be first DDC method in the world [155]. PID appears in almost all control journals. From PID and similar strategies, we can see that the DDC methods have bright future.

3.1.2.2. Iterative feedback tuning (IFT). IFT was proposed by Hjalmarsson in 1994 [47]. It is a typical data-driven control scheme involving iterative optimization of the parameter of the fixed controller according to an estimated gradient of a control performance criterion. At each iteration, the estimate is constructed from a finite set of data obtained partly from the normal operating condition of the closed-loop system and partly from a special experiment in which the output of the plant is fed back in the reference signal of the closed loop. A closed loop control system is shown in Fig. 7, where $P(z^{-1})$ is a SISO LTI plant, $C(\boldsymbol{\theta},z^{-1})$ is a fixed parameterized LTI controller with parameter vector $\boldsymbol{\theta}$, and the signals r, u and y are the reference, control input, and plant output, respectively. The control performance criterion is defined as follows:

$$J(\boldsymbol{\theta}) = \frac{1}{2N}\sum_{k=1}^{N}(y(\boldsymbol{\theta},k) - y_d(k))^2, \tag{16}$$

where $y(\boldsymbol{\theta},k)$ is the output of closed-loop with controller $C(\boldsymbol{\theta},z^{-1})$, y_d is a user-specified desired output, and N is the number of samples considered. The minimization objective is to find the optimal $\boldsymbol{\theta}^*$ satisfying.

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}}(J(\boldsymbol{\theta})). \tag{17}$$

If the gradient $\partial J/\partial\boldsymbol{\theta}$ is available, then $\boldsymbol{\theta}^*$ can be obtained using the following iterative algorithm:

$$\boldsymbol{\theta}_{i+1} = \boldsymbol{\theta}_i - \gamma_i R_i^{-1}\frac{\partial J(\boldsymbol{\theta}_i)}{\partial\boldsymbol{\theta}}, \tag{18}$$

where γ_i is a positive real scalar step size, and R_i is an appropriate positive definite matrix.

Using (16), the following is produced:

$$\frac{\partial J(\boldsymbol{\theta}_i)}{\partial\boldsymbol{\theta}} = \frac{1}{N}\sum_{k=1}^{N}(y(\boldsymbol{\theta}_i,k) - y_d(k))\frac{\partial y(\boldsymbol{\theta}_i,k)}{\partial\boldsymbol{\theta}}. \tag{19}$$

Because $y(\boldsymbol{\theta}_i,k)$ can be measured in closed-loop, and $y_d(k)$ is known, only $\partial y(\boldsymbol{\theta}_i,k)/\partial\,\boldsymbol{\theta}$cannot be computed when $P(z^{-1})$ is unknown. Estimating $\partial y(\boldsymbol{\theta}_i,k)/\partial\boldsymbol{\theta}$ in IFT is constructed from data collected in the closed loop with the actual controller.

As shown in Fig. 7, $y(\boldsymbol{\theta})$ can be described as follows:

$$y(\boldsymbol{\theta}) = \frac{C(\boldsymbol{\theta},z^{-1})P(z^{-1})}{1 + C(\boldsymbol{\theta},z^{-1})P(z^{-1})}r. \tag{20}$$

Eq. (20) yields the following:

$$\frac{\partial y(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}} = \frac{1}{C(\boldsymbol{\theta},z^{-1})}\frac{\partial C(\boldsymbol{\theta},z^{-1})}{\partial\boldsymbol{\theta}}\left[\frac{C(\boldsymbol{\theta},z^{-1})P(z^{-1})}{1 + C(\boldsymbol{\theta},z^{-1})P(z^{-1})}(r - y(\boldsymbol{\theta}))\right]. \tag{21}$$

The term in the square bracket can be obtained by using the signal $r - y(\boldsymbol{\theta})$ as the reference signal in the closed loop, which serves as the plant output in a new experiment; thus two experiments are involved in IFT algorithm for each iteration. The first experiment, namely normal experiment, collects the corresponding N samples of plant output denoted as $y_1(\boldsymbol{\theta}_i)$ by setting the reference $r = y_d$. The second experiment, called the gradient experiment, collects the corresponding N samples of plant output denoted as $y_2(\boldsymbol{\theta}_i)$ by setting the reference $r = y_d - y_1(\boldsymbol{\theta}_i)$. With these two pairs of data sets, the estimate of $\partial y(\boldsymbol{\theta}_i)/\partial\boldsymbol{\theta}$ is computed as follows:

$$\frac{\partial\hat{y}(\boldsymbol{\theta}_i)}{\partial\boldsymbol{\theta}} = \frac{1}{C(\boldsymbol{\theta}_i,z^{-1})}\frac{\partial C(\boldsymbol{\theta}_i,z^{-1})}{\partial\boldsymbol{\theta}}y_2(\boldsymbol{\theta}_i). \tag{22}$$

Using $\partial\hat{y}(\boldsymbol{\theta}_i,k)/\partial\boldsymbol{\theta},\ k = 1,\ldots,N$ from two experiments, Eq. (19) gives the estimate $\partial\widehat{J}(\boldsymbol{\theta}_i)/\partial\boldsymbol{\theta}$ finally. Then according to (18), the new parameter can be estimated. With some suitable assumptions, the algorithm can be shown to converge to a local minimum of the control performance criterion [49].

Several studies have evaluated the extension of prototype IFT to nonlinear systems [48,118–120]. Ref. [48] has provided a preliminary analysis of IFT for the nonlinear systems. It showed that IFT is workable if the first order Taylor approximation of the nonlinear plant near the trajectories in the first experiment is reasonably accurate. In [118–120], the standard IFT is extended to the case where both the plant and the controller can be nonlinear, and the proposed method requires $n + 1$ or $n + 2$ experiments to compute the derivatives of each iteration, where n is the number of controller parameters. The method

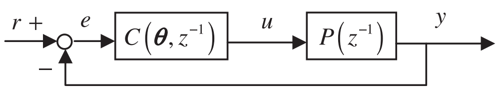

Fig. 7. Closed-loop control system of IFT.

presented in [119,120] is in order to reduce the number of experiments in each iteration to one by describing the nonlinear plant as a linearized time-varying model along the reference trajectory. Other modifications are explored in [51].

IFT is a data-driven controller tuning method, and it does not require a model of the controlled plant in the tuning procedure. However, the shortcomings are also obvious. First, because the gradient experiments are needed at each iteration, the unqualified products and time requirements are unavoidable. IFT is developed using a fixed structure controller, but there is no guideline for the selection of the controller. There is no guarantee of the closed loop stability.

In [49,51], the industrial and laboratory applications of IFT were explored. Ref. [50] extended IFT to MIMO plants. In [41] an algorithm on the step size selection in order to improve the efficiency of the IFT was proposed. In [104], a fuzzy control method combined with IFT is proposed. In [45,46], an optimal prefilter for the input data in IFT was introduced in order to enhance the accuracy of the IFT update. Other modifications and applications can be found in [68] and [79].

In [72], another DDC method, called spectral analysis, is introduced. This technique uses the spectral analysis of closed-loop experimental data to compute the derivatives of cost functions with respect to the controller parameters. This method could be viewed as a frequency-domain version of IFT.

3.1.2.3. Correlation-based tuning (CbT). Correlation-based tuning was proposed by Karimi et al. in 2002 [74]. It is a data-driven iterative controller tuning method. The underlying idea is inspired by the well-known correlation approach in system identification. The controller parameters are tuned iteratively either to decorrelate the closed-loop output error between designed and achieved closed-loop systems with the external reference signal (decorrelation procedure) or to reduce this correlation (correlation reduction). The block diagram of CbT method is shown in Fig. 8, where $P(z^{-1})$ is a SISO LTI plant, $C(\boldsymbol{\theta},z^{-1})$ is a parameterized LTI controller with parameter vector $\boldsymbol{\theta}$, and the signals r, u, y, and v are the reference, control input, plant output, and output disturbance, respectively. Suppose that the controller $C_d(z^{-1})$ is designed using the plant model $P_d(z^{-1})$ such that the closed-loop consisting of $C_d(z^{-1})$ and $P_d(z^{-1})$ equals to reference model $M(z^{-1})$. When $C(\boldsymbol{\theta},z^{-1})$ is applied to the real plant $P(z^{-1})$, the real closed-loop output is as follows:

$$y = \frac{C(\boldsymbol{\theta},z^{-1})P(z^{-1})}{1+C(\boldsymbol{\theta},z^{-1})P(z^{-1})}r + \frac{1}{1+C(\boldsymbol{\theta},z^{-1})P(z^{-1})}v.$$

The desired output is as follows:

$$y_d = M(z^{-1})r = \frac{C_d(z^{-1})P_d(z^{-1})}{1+C_d(z^{-1})P_d(z^{-1})}r.$$

Then the closed-loop output error is as follows:

$$\varepsilon = y - y_d = \frac{C(\boldsymbol{\theta},z^{-1})P(z^{-1}) - C_d(z^{-1})P_d(z^{-1})}{(1+C(\boldsymbol{\theta},z^{-1})P(z^{-1}))(1+C(\boldsymbol{\theta},z^{-1})P_d(z^{-1}))}r + \frac{1}{1+C(\boldsymbol{\theta},z^{-1})P(z^{-1})}v. \tag{23}$$

The closed-loop output error ε contains contributions from the difference between $C(\boldsymbol{\theta},z^{-1})P(z^{-1})$ and $C_d(z^{-1})P_d(z^{-1})$ and the disturbance v. The contribution, originating from the difference between $C(\boldsymbol{\theta}, z^{-1})P(z^{-1})$ and $C_d(z^{-1})P_d(z^{-1})$, is correlated with the reference signal r. In other words, if we tune $\boldsymbol{\theta}$ such that the closed-loop output error ε is completely not correlated with the reference signal r, this means that the difference between $C(\boldsymbol{\theta},z^{-1})P(z^{-1})$ and $C_d(z^{-1})P_d(z^{-1})$ is zero and the perfect reference model tracking is achieved regardless of the presence of the disturbance v. This features the CbT method.

The cross-correlation function is as follows:

$$\xi(\boldsymbol{\theta}) = E\{\hat{\xi}(\boldsymbol{\theta})\}, \tag{24}$$

where $E\{\cdot\}$ denotes the mathematical expectation and $\hat{\xi}(\boldsymbol{\theta})$ is defined as follows:

$$\hat{\xi}(\boldsymbol{\theta}) = \frac{1}{N}\sum_{k=1}^{N}\zeta(k)\varepsilon(\boldsymbol{\theta},k), \tag{25}$$

where $\varepsilon(\boldsymbol{\theta},k)$ is the closed-loop output error when $C(\boldsymbol{\theta},z^{-1})$ is in the loop, $\zeta(k)$ is the instrumental variable correlated with $r(k)$ and independent of $v(k)$, and N is the number of data. If the controller set is large enough to allow for perfect decorrelation of ε and r, then CbT calculates the controller parameters as the roots of the cross-correlation function. This is called the

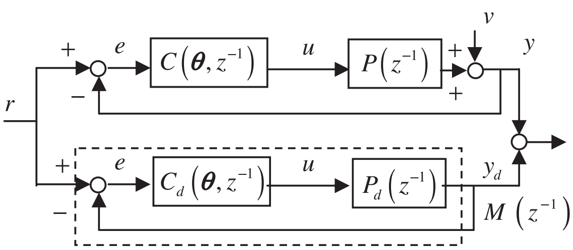

Fig. 8. Block diagram of CbT method.

decorrelation procedure. If the controller that achieves decorrelation does not exist in the controller set, CbT updates the controller parameters by minimizing the cross-correlation function, which is called correlation reduction.

The decorrelation procedure (details of correlation reduction see [93]) is to find the roots of the following equation:

$$\xi(\boldsymbol{\theta}) = 0 \tag{26}$$

$\hat{\xi}(\boldsymbol{\theta})$ can be viewed as the measurement of $\xi(\boldsymbol{\theta})$ with noise. Eq. (26) is of the exact form for which the Robbins–Monro stochastic approximation algorithm is intended. In other words, the solution of (26) can be found using the following algorithm:

$$\boldsymbol{\theta}_{i+1} = \boldsymbol{\theta}_i - \gamma_i \hat{\xi}(\boldsymbol{\theta}_i), \tag{27}$$

where γ_i is a positive scalar step size and $\hat{\xi}(\boldsymbol{\theta})$ can be easily computed from (25) using data collected during the closed-loop experiment with $C(\boldsymbol{\theta}_i, z^{-1})$. To ensure the convergence of the R–M algorithm, the instrumental variable is selected as the estimate of the gradient of plant output with respect to the controller parameters. These parameters can be computed using the model of the plant. The accuracy of the model does not affect the degree of convergence but it does affect the rate of convergence [93].

CbT and IFT are closely related methods, but one should keep in mind that they differ in two very important respects: (a) underlying control objective and (b) means of obtaining gradient estimates (Section 3.2.2 in [93]). Unlike IFT, CbT needs only one experiment per iteration. In [94], CbT was extended to MIMO systems. In [75,92], CbT is applied to suspension system.

3.1.2.4. Virtual reference feedback tuning (VRFT). VRFT was proposed by Guardabassi and Savaresi in 2000 [42]. It is a one-shot direct data-driven method that can be used to select the controller parameter for the LTI system. VRFT formulates the controller tuning problem as a controller parameter identification problem via introducing virtual reference signal.

The block diagram of VRFT is shown in Fig. 9a, where $P(z^{-1})$ is an unknown SISO LTI plant, $C(\boldsymbol{\theta}, z^{-1})$ is a parameterized LTI controller with parameter vector $\boldsymbol{\theta}$, $M(z^{-1})$ is a user-specified reference model, and the signals r, u, and y are the reference, control input, and plant output, respectively. The control objective is the minimization of the following model-reference criterion:

$$J(\boldsymbol{\theta}) = \left\| \frac{C(\boldsymbol{\theta}, z^{-1})P(z^{-1})}{1 + C(\boldsymbol{\theta}, z^{-1})P(z^{-1})} r - M(z^{-1}) r \right\|^2. \tag{28}$$

Because $P(z^{-1})$ is unknown, the minimization of $J(\boldsymbol{\theta})$ cannot be performed. The traditional approach is to identify the model $P_d(z^{-1})$ of $P(z^{-1})$ using a sample I/O data set $\{(u(k), y(k))_{k=1,\ldots,N}\}$ of the plant and then minimize $J(\boldsymbol{\theta})$ by substituting $P_d(z^{-1})$ for $P(z^{-1})$ in (28). However, this renders modeling very difficult and introduces unavoidable modeling error. VRFT avoids the building model procedure. As shown in Fig. 9b, VRFT derives a virtual I/O data set $\{(e^{vir}(k), u^{vir}(k))_{k=1,\ldots,N}\}$ of controller $C(\boldsymbol{\theta}, z^{-1})$ from the sampling I/O data set $\{(u(k), y(k))_{k=1,\ldots,N}\}$ of the plant. Here,

$$e^{vir}(k) = r^{vir}(k) - y(k) = M^{-1}(z^{-1})y(k) - y(k),$$
$$u^{vir}(k) = u(k).$$

where $M^{-1}(z^{-1})$ is the inversion of reference model. Signal r^{vir} is called the virtual reference, where "virtual" indicates that it does not exist in reality and was not in place when data set $\{(u(k), y(k))_{k=1,\ldots,N}\}$ is collected. It only exists in the computer in which it is constructed using relationship $r^{vir} = M^{-1}(z^{-1})y$.

Using the virtual I/O data set $\{(e^{vir}(k), u^{vir}(k))_{k=1,\ldots,N}\}$ of controller $C(\boldsymbol{\theta}, z^{-1})$ to identify the controller parameter $\boldsymbol{\theta}$ means minimizing the following criterion:

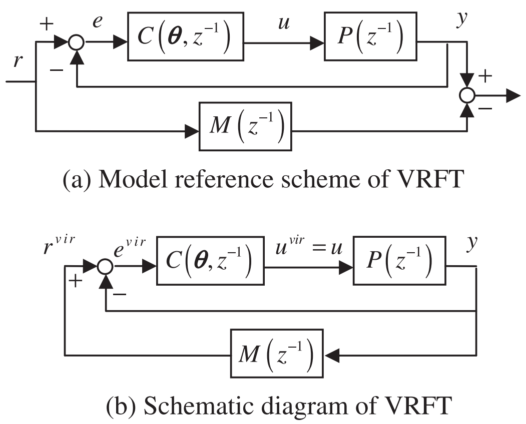

Fig. 9. VRFT method.

$$J_{VRFT}(\boldsymbol{\theta}) = \|C(\boldsymbol{\theta}, z^{-1})e^{vir} - u^{vir}\|^2. \tag{29}$$

The original control objective is to minimize criterion (28). In order to avoid the procedure of building model, VRFT minimizes the criterion (29). The question is whether the two criterions are equivalent. Theorem 1 in [27] shows that $J_{VRFT}(\boldsymbol{\theta})$ shares with $J(\boldsymbol{\theta})$ the same minimizer in the case the optimal solution $\boldsymbol{\theta}^*$ of $J(\boldsymbol{\theta})$ satisfies $J(\boldsymbol{\theta}^*) = 0$. In other words, the controller class is large enough to allow for perfect matching in this case. In other cases, the two criteria do not have the same minimizer. Proposition 1 in [25] shows that selecting the appropriate filter for virtual signals e^{vir} and u^{vir} could ensure the two criterions have the same minimizer. However, designing the perfect filter requires that $P(z^{-1})$ is known. In this way, the selection of controller class determines the effect of controller tuning. If the controller class is large enough to allow for perfect matching, then VRFT offers a significant advantage.

VRFT directly identifies the optimal parameters of the controller using the I/O data set of the controlled plant. However, there are also a number of drawbacks: (a) The tuning procedure is off-line, so the parameters must be re-tuned if the plant structure changes. (b) The selection of controller set determines the effect of controller tuning and there is no guideline for the selection. (c) Not all data sets contain sufficient dynamic information.

Ref. [27] explored the extension of VRFT approach to a nonlinear system. Unlike the linear version, in the nonlinear setup, VRFT is not a one-shot method. In order to produce a suitable controller parameter for a user-specified reference signal r(which in gerenal is different from r^{vir}), VRFT resorts to a multi-pass, iterative procedure. In [97], VRFT was extended to a multi-input multi-output system. In [113], a modification of VRFT was given, which may have alleviated noise-induced correlations and allowed the use of this approach in unstable plants. Ref. [114] showed that VRFT could be formulated in different settings, such as control-relevant identification, closed-loop identification, and controller identification. In [105,106], VRFT was applied to the design of closed-loop controllers for FES-supported systems. In [152], VRFT was used to control a vertical single-link arm. In [26], VRFT has been successful as a control method for a benchmark active suspension system. In [73], it was reported that an adaptive VRFT design method could be used for adaptive PID controller parameter tuning. In [107], VRFT was used as the velocity controller in a self-balancing industrial manual manipulator.

3.1.2.5. Noniterative data-driven model reference control. Noniterative data-driven model reference control is proposed by Van Heusden et al. in [76,137]. This controller tuning approach leads to an identification problem where the input is affected by noise but not the output as in standard identification problems.

Consider the unknown LTI SISO plant $P(z^{-1})$. The objective is to design a linear, fixed controller $C(\boldsymbol{\theta}, z^{-1})$ with parameters $\boldsymbol{\theta}$ such that the closed-loop approximates the reference model $M(z^{-1})$. This can be achieved by minimizing the following model-reference criterion:

$$J(\boldsymbol{\theta}) = \left\| M(z^{-1})r - \frac{C(\boldsymbol{\theta}, z^{-1})P(z^{-1})}{1 + C(\boldsymbol{\theta}, z^{-1})P(z^{-1})} r \right\|^2. \tag{30}$$

where r is the reference signal. Note that the objective is to design a fixed controller and $J(\boldsymbol{\theta}) = 0$ cannot generally be achieved. The model reference criterion (30) is nonconvex with respect to the controller parameters $\boldsymbol{\theta}$. An approximation that is convex for linearly parameterized controllers can be defined using the reference model $M(z^{-1})$ as a illustration next. The notation is shortened by dropping these argument z^{-1} for simplicity. M can be represented as follows:

$$M = \frac{C^* P}{1 + C^* P}, \tag{31}$$

where C^* is the ideal controller, which is defined indirectly by P and M:

$$C^* = \frac{M}{P(1 - M)}. \tag{32}$$

This controller C^* exists if $M \neq 1$. The unknown ideal controller will only be used for analysis. Using (31), (30) can be rewritten as

$$J(\boldsymbol{\theta}) = \left\| \frac{C^* P - C(\boldsymbol{\theta})P}{(1 + C^* P)(1 + C(\boldsymbol{\theta})P)} r \right\|^2. \tag{33}$$

Replacing $(1 + C(\boldsymbol{\theta})P)$ with $(1 + C^*P)$ leads to the following approximation of (33):

$$\widehat{J}(\boldsymbol{\theta}) = \left\| \frac{C^* P - C(\boldsymbol{\theta})P}{(1 + C^* P)^2} r \right\|^2 = \|(1 - M)[M - C(\boldsymbol{\theta})(1 - M)P]r\|^2 = \|(1 - M)Mr - C(\boldsymbol{\theta})(1 - M)Pr\|^2. \tag{34}$$

If the controller is linearly parameterized, then $\widehat{J}(\boldsymbol{\theta})$ is convex to the controller parameters $\boldsymbol{\theta}$.

Considering the case where there is measurement noise on the plant output, namely $y(k) = P(z^{-1})u(k) + \upsilon(k)$, where $u(k)$ is the plant input, $\upsilon(k)$ is the measurement noise and $P(z^{-1})$ is stable, the optimal solution of criterion (34) can be found by minimizing the norm of the following error

$$\varepsilon_c(\boldsymbol{\theta}) = M(1 - M)r - C(\boldsymbol{\theta})(1 - M)^2 y = M(1 - M)r - C(\boldsymbol{\theta})(1 - M)^2 Pr - C(\boldsymbol{\theta})(1 - M)^2 \upsilon. \tag{35}$$

The diagram of (35) is shown in Fig. 10a. This diagram can be re-diagramed as Fig. 10b in order to clearly show the nature of the identification problem. In Fig. 10b, the unknown signals are $y_c^*(k)$, $v(k)$ and $\tilde{y}_c(k)$. The known signals are $r(k)$, $y_c(k) = (1-M)^2 y(k)$, and $s(k)$. They are given by the following:

$$s(k) = (1-M)^2 PC^* r(k) = M(1-M)r(k).$$

The controller parameter tuning problem has become a parameter identification problem. Here the plant to be identified is C^* and the model to be identified is $C(\boldsymbol{\theta})$. The main difference between the controller tuning and the standard identification is that the input is affected by noise in the former ($y_c^*(k)$ is affected by the noise $\tilde{y}_c(k)$) while the output is affected by noise in the latter. The correlation approach is applied to address the affects of noise on input (see [138]).

In the scheme of Fig. 10, $P(z^{-1})$ is stable is assumed. For unstable $P(z^{-1})$, an initial stabilizing controller is needed to perform the experiment [138].

The model reference controller tuning method has been extended to a constrained case that ensures closed-loop stability. This constraint is derived from stability conditions based on the small-gain theorem.

Like VRFT, this approach converts the controller tuning problem into an off-line identification problem. The selection of controller set and whether the data set contains sufficient dynamic information become the main concerns.

3.1.2.6. Subspace approach. In literatures, the subspace approach [66,77,98], the data space approach [38,69,103] and the data-driven simulation approach [89–91] have been shown to share the idea that system dynamics are represented as a subspace of a finite-dimensional vector space, which consists of the time series data of input/state/output or input/output. The subspace approach is developed using the input/state/output representation, and the other two involve input/output representation. The cornerstone of these methods is that the basis of the finite-dimensional vector space, called the dynamic matrix, involves all dynamic information from the LIT system. The different subspace identification techniques available in the literature also differ in the manner in which the basis of the state space is estimated. The numerical tools used in the estimation of this basis include singular value decomposition [95,98], QR-decomposition [139], and canonical variable analysis [83,116]. Some subspace identification methods also differ in how the disturbances are characterized.

Here, to simplify the description, we introduce only the subspace predictive control approach and do not consider disturbances. In this method, the dynamic matrix is obtained using the Moore–Penrose pseudo-inverse algorithm. Then it serves as the predictor of the controlled plant. With this predictor, the so called data-driven MPC is established. The main idea underlying the data-driven MPC control is briefly reviewed as follows.

The controlled plant of the data-driven MPC is a LTI noise-free object that can be described by the following:

$$x(k+1) = Ax(k) + Bu(k),$$
$$y(k) = Cx(k) + Du(k),$$

where A, B, C, and D are time-invariant matrices with appropriate dimensions. Then the output equations can be expressed in a lifting form as follows:

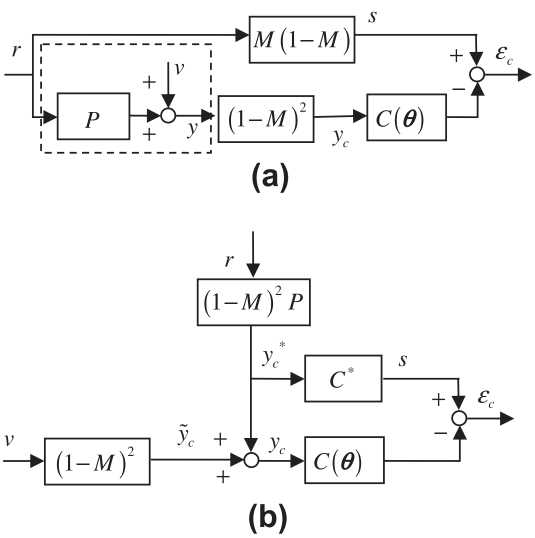

Fig. 10. Noniterative data-driven model reference control.

$$\begin{bmatrix} y(k) \\ y(k+1) \\ \vdots \\ y(k+i) \end{bmatrix} = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^i \end{bmatrix} x(k) + \begin{bmatrix} D & 0 & 0 & 0 \\ CB & D & 0 & 0 \\ \vdots & \cdots & \ddots & 0 \\ C^{i-2}B & C^{i-1}B & \cdots & D \end{bmatrix} u(k). \tag{36}$$

In order to determine the Hankel matrix of the system, using the I/O data and the Hankel matrix $H_{i\times j}$ of a signal $w(k)$

$$H_{i,j}(w(k)) := \begin{bmatrix} w(k) & w(k+1) & \cdots & w(k+j-1) \\ w(k+1) & w(k+2) & \cdots & w(j+1) \\ w(k+2) & w(k+3) & \cdots & w(j+1) \\ \vdots & \vdots & & \vdots \\ w(k+i-1) & w(k+i) & \cdots & w(k+i+j-2) \end{bmatrix}$$

the data matrices can be constructed as follows:

$$U_p = H_{i,j}(u(1)), \quad U_f = H_{i,j}(u(i)), \quad Y_p = H_{i,j}(y(1)), \quad Y_f = H_{i,j}(y(i))$$

Using (36), an equivalent data model, the predictor of the plant model can be described using (37) [66].

$$\widehat{Y}_f = L_w W_p + L_u U_f, \tag{37}$$

where $\widehat{Y}_f$ is the output of the data-driven predictor model, L_w and L_u are dynamic matrixes, and $W_p = \begin{bmatrix} Y_p^T & U_p^T \end{bmatrix}^T$.

If the dynamic matrixes are available, a data-driven MPC can be derived easily. Because of the Moore–Penrose pseudo-inverse and (37), the dynamic matrixes can be calculated using (38):

$$(L_w \ L_u) = Y_f \begin{pmatrix} W_p \\ U_f \end{pmatrix}^{+} = Y_f \left(W_p^T \ U_f^T \right) \left(\begin{pmatrix} W_p \\ U_f \end{pmatrix} (W_p \ U_f) \right)^{-1}. \tag{38}$$

In order to determine the predictive controller, the following a quadratic performance index of MPC can be introduced:

$$J = \sum_{i=1}^{N_y} \|r_{k+i} - \hat{y}_{k+i}\|_Q^2 + \sum_{i=1}^{N_u} \|u_{k+i}\|_R^2. \tag{39}$$

Optimizing (39) gives the MPC controller as (40).

$$u_f = \arg\min_{u_f} \left\{ (r_f - \hat{y}_f)^T Q (r_f - \hat{y}_f) + u_f^T R u_f \right\}. \tag{40}$$

Setting $Q = R = \mathbf{I}$, and substituting (37) into (40), a simple data-driven subspace predictive controller is obtained.

$$u_f = \left(\lambda I + L_u^T L_u \right)^{-1} L_u^T (r_f - L_w w_p). \tag{41}$$

This procedure only involves the projection step of subspace methods. No explicit information from the model has been included in this predictive controller. The model structure is implicitly involved in the controller (41). Theoretically speaking, the persistent excitation condition is another implicit assumption because the inverse of the matrix is included in the controller.

3.1.2.7. Approximate dynamic programming (ADP). Approximate dynamic programming has been proposed in [147,148] as a solution to optimal control problems forward-in-time. ADP combines reinforcement learning using adaptive critic structures with dynamic programming. ADP includes four main schemes [148]: heuristic dynamic programming, dual heuristic dynamic programming, action-dependent heuristic dynamic programming, i.e., Q-learning [144–146], and action-dependent dual heuristic dynamic programming. Here, Q-learning is introduced in detail because this method does not require knowledge of the plant model.

Q-learning was originally proposed as a solution to the discrete Markov decision processes (MDPs) where the number of state and action pairs is finite and the MDP model is not available by Watkins and Dayan in [144,145].

Consider the following deterministic Markov process:

$$x(k+1) = f(x(k), u(k)), \tag{42}$$

where the state $x(k) \in \mathbf{S}$, $\forall k \in \mathbf{N}$, $\mathbf{S}$ is a finite set of states, the action $u(k) \in \mathbf{A}$, $\forall k \in \mathbf{N}$, $\mathbf{A}$ is a finite set of actions, and $f(\cdot)$ is an unknown function. The goal is to find the optimal policy π^* that will minimize the following cost function:

$$J(x(0), \pi) = \sum_{t=0}^{\infty} \gamma^t r(x(t), u(t)) \tag{43}$$

where $\gamma \in [0,1]$ is a discount factor, $r(\cdot)$ is a single-stage cost function, and $u(t) = \pi(x(t))$, $t = 0, \ldots, \infty$.

If the $f(\cdot)$ is known, dynamic programming (DP) is a general approach for solving the above optimization problems. The objective of DP is to obtain a so-called cost-to-go function, defined as follows:

$$J^*(x(k)) = \min_{\pi}\left(\sum_{t=k}^{\infty}\gamma^t r(x(t), \pi(x(t)))\right) = \sum_{t=k}^{\infty}\gamma^t r(x(t), \pi^*(x(t))), \tag{44}$$

where π^* is the optimal policy. Note that $J^*(x(0))$ is the optimal solution to (43). Eq. (44) can be rewritten as follows:

$$J^*(x(k)) = \min_{u(k)}(r(x(k), u(k)) + \gamma J^*(x(k+1))). \tag{45}$$

The above equation is the well-known Bellman equation. Once $f(\cdot)$ and $J^*(\cdot)$ have been obtained, one can solve the following single-stage optimal decision problem on-line:

$$\pi^*(x(k)) = \arg\min_{u(k)}(r(x(k), u(k)) + \gamma J^*(f(x(k), u(k)))). \tag{46}$$

This is equivalent to the above infinite-horizon problem. The off-line iterative algorithms of obtaining the cost-to-go function $J^*(\cdot)$ for finite states and actions have been described previously [17].

In cases where $f(\cdot)$ is unknown, obtaining $J^*(\cdot)$ does not facilitate selection of optimal actions because (45) cannot be solved. To overcome the obstacle, a new cost-to-go function, the Q-function, can be defined as follows:

$$Q(x(k), u(k)) = r(x(k), u(k)) + \gamma J^*(f(x(k), u(k))). \tag{47}$$

Note that $Q(x(k),u(k))$ is exactly the quantity that is minimized in (46) so that the optimal action $\pi^*(x(k))$ in state $x(k)$ can be selected. Therefore, we can rewrite (46) in terms of $Q(x(k),u(k))$ as follows:

$$\pi^*(x(k)) = \arg\min_{u(k)}(Q(x(k), u(k))). \tag{48}$$

This shows that if $Q(\cdot)$ is obtained instead of $J^*(\cdot)$, it will be allow the user to select optimal actions even when no knowledge of $f(\cdot)$ is available. Note that

$$J^*(x(k)) = \min_{u(k)}(Q(x(k), u(k))),$$

which allows us to rewrite (47) as

$$Q(x(k), u(k)) = r(x(k), u(k)) + \gamma\min_{u'}Q(f(x(k), u(k)), u'). \tag{49}$$

This recursive definition of $Q(\cdot)$ provides the basis for off-line algorithms that iteratively approximate $Q(\cdot)$ as follows:

$$\widehat{Q}_{i+1}(x(k), u(k)) = (1-\alpha)\widehat{Q}_i(x(k), u(k)) + \alpha(r(x(k), u(k)) + \gamma\min_{u'}\widehat{Q}_i(x(k+1), u')), \tag{50}$$

where $\widehat{Q}_i(\cdot)$ is the approximation of $Q(\cdot)$ at the i-th iteration and α is a learning rate parameter between 0 and 1. Using this algorithm $\widehat{Q}_i(\cdot)$ converges to the actual $Q(\cdot)$, provided actions are chosen so that every state-action pair is visited infinitely [144].

However, the conventional Q-learning methods are not well suited to process control problems because of the continuous nature of typical state and action spaces. This is because different control policies and randomization do not ensure multiple visits to the same exact states. Some modified Q-learning algorithms are better suited to process control. These can be found in published literature. The main idea of [122] was to use an existing approximated Q-function to train neighboring Q-functions and to use a hyper-elliptic hull to prevent extrapolation. Ref. [84] employs local averages for the approximation of the Q-function. Both considered general nonlinear systems, but neither evaluated convergence of the algorithms.

For process control problems in linear systems, some results can be found in [4,21,78]. In [21], the Q function is expressed by a parametric linear quadratic function. The recursive least squares technique guarantees the estimate converges to the true parameters as persistent excitation condition is satisfied. In [4], a Q-learning model-free approach is proposed to solve the zero-sum game forward in time. It had been shown that the critic networks converge to the game value function and the action networks converge to the Nash equilibrium of the game. The main drawback of this method is that it requires a great deal of computing power. To overcome this, Kim et al. derived an iterative solution algorithm using linear matrix inequalities (LMI) and policy iteration for H_∞ control design [78]. Under the condition that probing noise is sufficiently rich, the parametric matrix can be estimated, and the optimal control policy can be guaranteed.

Because it is model-free, Q-learning has been used in many practical applications. Weissensteiner used it to derive optimal consumption and investment strategies [146]. Park et al. realized the multi-agent cooperation for robot soccer based on a modularized Q-learning [102]. Lim et al. used Q-learning to design guide-path networks for automated guided vehicles [85].

3.1.3. On-line/off-line data based hybrid DDC methods

3.1.3.1. Iterative learning control (ILC). Iterative learning control was first proposed by Uchiyama in Japanese in 1978 [132], which did not get much attentions. After one critical report [9] was published in 1984, ILC was extensively studied and

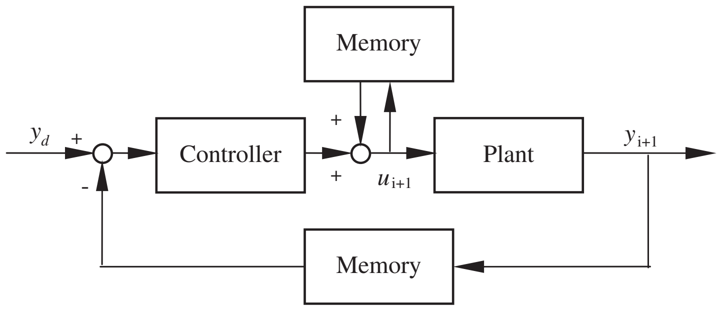

Fig. 11. Block diagram of ILC system.

significant progress was made in both theory and application in many fields. For a system that repeats the same task in a finite interval, ILC is an ideal technique to learn from the repetitive dynamics to achieve better control performance. ILC has a very simple controller structure and requires little prior knowledge of the system. It can guarantee learning error convergence as the number of iterations approaches infinity. Several previous studies provided a comprehensive and systematic summary of the recent ILC research [33,96,130,149,150]. The contraction mapping method forms the basis of most ILC theory [31,82,149].

A block diagram of an ILC system is shown in Fig. 11. Two memory components are used to record the control signal and output signal of the preceding trials. Let us consider the following dynamic system:

$$\begin{aligned} \boldsymbol{x}_i(k+1) &= \boldsymbol{f}(x_i(k), \boldsymbol{u}_i(k), k), \\ \boldsymbol{y}_i(k) &= \boldsymbol{g}(x_i(k), \boldsymbol{u}_i(k), k), \end{aligned} \tag{51}$$

where $\boldsymbol{f}$ and $\boldsymbol{g}$ are global *Lipschitz* continuous functions of the arguments $\boldsymbol{x}_i$ and $\boldsymbol{u}_i$; $x_i(k) \in \mathbf{R}^n$, $\boldsymbol{y}_i(k) \in \mathbf{R}^m$, and $\boldsymbol{u}_i(k) \in \mathbf{R}^r$ are the state, plant output, and control input at instant k, respectively; $k \in \{0, 1, \ldots, T\}$ denotes the specific point in time; and $i \in \{0, 1, 2, \ldots\}$ denotes the number of iterations.

The control task is to drive the output $y_i(k)$ to track the desired output $y_d(k)$ on a fixed interval $k \in [0, T]$ for any k as the iteration i goes to infinity. In other word, we expect tracking error $\boldsymbol{e}_i(k) = \boldsymbol{y}_d(k) - \boldsymbol{y}_i(k)$, $\forall k \in [0, T]$ uniformly converge to zero when iteration $i \to \infty$.

The general ILC controller designing block diagram is shown in Fig. 12. It is clear that the control input $\boldsymbol{u}_i(k)$, at instant k of the ith iteration, can be designed using the all control inputs before time instant k at the ith iteration, the control inputs at all instants of N iterations before the ith iteration, the tracking errors as instant k of the ith iteration and before, and the tracking error at all instants of N iterations before the ith iteration. The most general iteration learning law can be expressed as follows:

$$\boldsymbol{u}_i(k) = h(\boldsymbol{u}_i(< k), \boldsymbol{u}_{i-1}(\cdot), \ldots, \boldsymbol{u}_{i-N}(\cdot), \boldsymbol{e}_i(\leqslant k), \boldsymbol{e}_{i-1}(\cdot), \ldots, \boldsymbol{e}_{i-N}(\cdot)). \tag{52}$$

Apparently, P-type learning law, D-type learning law, PID-type learning law, high-order learning law, robust learning law, optimal learning law, and feedback–feedforward learning law are the special cases of (52).

Considering the P-type law $\boldsymbol{u}_i(k) = \boldsymbol{u}_{i-1}(k) + L(k)\boldsymbol{e}_{i-1}(k)$, its convergence condition is $|1 - L \cdot \partial \mathbf{g}/\partial \boldsymbol{u}| < 1$. This means that if plant (51) is global *Lipschitz* and the boundedness of $\partial \mathbf{g}/\partial \boldsymbol{u}$ is known, then the learning gain L can be properly chosen to ensure the convergence conditions. No other information about the plant is needed. Convergence conditions of the other kinds of learning laws are similar.

The features of ILC can be summarized as follows: (a) ILC aims at output tracking control without any knowledge of the system state dynamics. (b) It has a very simple structure and is an integrator along the iteration axis; (c) it is a memory-based learning process. (d) It requires very little system knowledge, so it is a data-driven model-free method. (e) The identical initialization conditions play an important role in the learning process. (f) The target trajectory $\boldsymbol{y}_d(k)$ must be identical for all iterations.

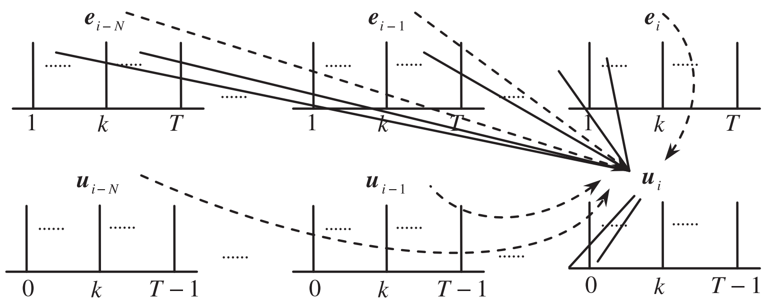

Fig. 12. Schematic design diagram of ILC.

In addition, ILC has been widely applied in many fields [3,59]. ILC uses data in a more abundant and systematic way than other DDC approaches. ILC uses both on-line and off-line. ILC does not use data to tune controller parameters but to directly determine the optimal control input signal.

3.1.3.2. Lazy learning (LL). LL algorithms are kinds of supervised machine learning algorithm. Schaal and Atkeson first applied lazy learning algorithms to control problems in 1994 [115]. Like other supervised machine learning algorithms, the goal of LL algorithms is to determine the relationship between input and output from a collection of input and output data called the training set.

Here, we just introduce a simple LL algorithm (for more elaborate LL algorithms see previous studies [2,13,18,19]). Let us consider an unknown nonlinear function $y = f(\boldsymbol{\phi})$, where f:$\mathbf{R}^n \to \mathbf{R}$, $\boldsymbol{\phi} \in \mathbf{R}^n$, and $y \in \mathbf{R}$ with a collection of input/output values, $\{(\boldsymbol{\phi}_i, y_i)_{i=1,\ldots,N}\}$ called training data set. Now, we want to estimate the output $y_q \in \mathbf{R}$ of the query point $\boldsymbol{\phi}_q \in \mathbf{R}^n$. In order to estimate $\hat{y}_q$, there are three steps as following:

Step 1. *Local model generation using local weighted linear regression*. The local model is the linear function $y = [\boldsymbol{\phi}^T,1] \cdot \boldsymbol{\theta}$, where $\boldsymbol{\theta} \in \mathbf{R}^{n+1}$ is the parameter vector. For given h, local weighted linear regression is used to find the optimal solution $\boldsymbol{\theta}^*(\boldsymbol{h})$ of the following criterion:

$$J(\boldsymbol{\theta}, h) = \sum_{i=1}^{N} \left\{ (y_i - [\boldsymbol{\phi}_i^T, 1] \cdot \boldsymbol{\theta})^2 \cdot K\left(\frac{D(\boldsymbol{\phi}_i, \boldsymbol{\phi}_q)}{h}\right) \right\}, \tag{53}$$

where $D(\boldsymbol{\phi}_i, \boldsymbol{\phi}_q)$ is the distance function (e.g., Euclidean distance between $\boldsymbol{\phi}_i$ and $\boldsymbol{\phi}_q$), and h is the bandwidth of the weighting function $K(\cdot)$, selected as follows:

$$K(x) = \begin{cases} 1, & x \leqslant 1, \\ 0, & x > 1. \end{cases}$$

Selecting M and the different values h, we obtain a set of local model candidates, $\{(\boldsymbol{\theta}^*(h_i))_{i=1,\ldots,M}\}$.

Step 2. *Local model validation*. Criterion (53) is used to validate each of the candidate sets $\{(\boldsymbol{\theta}^*(h_i))_{i=1,\ldots,M}\}$. The validation is biased because it uses the same data set $\{(\boldsymbol{\phi}_i, y_i)_{i=1,\ldots,N}\}$ as the identification. More effective validations are given in previous studies [13,18,19].

Step 3. *Local model selection and function output estimation*. The optimal local model is as follows.

$$\boldsymbol{\theta}^*(h^*) = \arg \min_{\boldsymbol{\theta} \in \{(\boldsymbol{\theta}^*(h_i))_{i=1,\ldots,M}\}} J(\boldsymbol{\theta}, h).$$

Then the estimation of the output of the query point $\boldsymbol{\phi}_q$ is

$$\hat{y}_q = [\boldsymbol{\phi}_q^T, 1] \cdot \boldsymbol{\theta}^*(h^*).$$

After determining $\hat{y}_q$, the LL algorithm discards the optimal local model $y = [\boldsymbol{\phi}^T,1] \cdot \boldsymbol{\theta}^*(h^*)$. For a new query point, the above three steps must be repeated. LL algorithms are estimations of function output values but not of functions. Although the local model complexity is lower than global model complexity, LL algorithms must to build a local model for each query point. In this way, computational costs of LL algorithms are high. In order to reduce these computational costs, the local models selected are usually as simple as possible and the linear model is usually the most widely used. The local model just describes the mapping relation near the query point. In this way, though the local model is as simple as a linear function, LL algorithms can provide highly accurate estimates when applied to estimate the output of complex nonlinear functions.

When the local model of LL algorithms is selected as a linear model, nonlinear systems become easier to handle. LL control is a divide-and-conquer control method [19]. Its main idea is that first, a local linear dynamic model of each time instant is built by LL algorithms. Then, a local controller at each time instant is designed according to the local linear dynamic model. Here we introduce the LL self-tuning control [18]. Let us consider the following SISO nonlinear plant

$$y(k+1) = f(y(k), \ldots, y(k-n_y), u(k), \ldots, u(k-n_u)), \tag{54}$$

where $y(k) \in \mathbf{R}$ is the output at instant k, $u(k) \in \mathbf{R}$ is the control input at instant k, n_y and n_u are the known order of the plant, and $f : \mathbf{R}^{n_y+n_u} \to \mathbf{R}$ is an unknown nonlinear function. Using query point

$$\boldsymbol{\phi}_q(k) = [y(k), \ldots, y(k-n_y), u(k), \ldots, u(k-n_u)]^T \tag{55}$$

and the LL algorithm, the local linear dynamic model can be constructed as follows

$$y(k+1) = [y(k), \ldots, y(k-n_y), u(k), \ldots, u(k-n_u), 1] \cdot \boldsymbol{\theta}^*(h^*) \tag{56}$$

for (54) at instant k, where $\boldsymbol{\theta}^*(h^*) \in \mathbf{R}^{n_y+n_u+1}$. The element $u(k)$ of both $\boldsymbol{\phi}_i$ and $\boldsymbol{\phi}_q$ is ignored when computing the distance function $D(\boldsymbol{\phi}_i, \boldsymbol{\phi}_q)$. This is because $u(k)$ is not available for $\boldsymbol{\phi}_q$ which is the expected outcome of the procedure. After getting the local linear dynamic model (56), we can use minimum-variance or pole placement controller design approach to design the local controller. This controller generates the control input $u(k)$ at instant k.

LL control, using the historical data set of the plant, builds a local linear dynamic model for the nonlinear plant at every time instant, and then instantaneously designs a local controller based on the available local linear model. Because the historical data set is constantly updated, LL control can be considered an intrinsically adaptive method. However, its computational cost is high, and there is also lack of the theoretical analysis of the stability. Ref. [18] combined the LL algorithms with conventional linear control techniques (e.g., minimum variance, pole placement, optimal control). In [13,115], LL based control method was applied to robot control. In [80], a new LL control approach is directly used to compute the control input without help of the local model of the plant. In [99], the LL algorithm was used to tune the PID controller parameter. In [100], an LL based algorithm was proposed to address the management of large datasets and search for the relevant neighbors in order to improve the computational efficiency.

In literatures, there are several methods similar to LL, such as just-in-time learning (JITL) [44], instance-based learning [1], local weighted model [13], and model-on-demand [22,67]. In addition, some researchers propose another approach based on Taylor series expansion at the operating point and neighboring data query aiming at to more efficient utilization of the I/O database. Many details remain to be studied [43,101,154].

3.2. DDC classification by controller structure design

We have classified DDC methods according to data usage in the controller design. In this subsection, we will use another criterion, whether the structure of the controller is known or not, to classify DDC approaches. With this criterion, DDC methods can be classified into two types.

3.2.1. DDC methods with fixed controller structures

This kind of DDC method involves controller design that depends only on plant I/O measurements with a pre-specified fixed controller structure. Controller parameters are obtained from some optimization procedures, such as batched or recursive algorithms. Here the controller design problem is transformed into the controller parameter identification with the help of the assumption that the controller structure is known prior and linear in controller parameters. PID, IFT, VRFT, UC, SPSA-based control, CbT, and ADP are typical of this kind of method. No information regarding the plant model or dynamics is involved. The main issue in this kind of methods is how to determine controller structure for a given controlled plant. Obtaining good controller structure with unknown parameters, especially for general nonlinear systems, is quite difficult. It can be as hard as modeling a plant. Another issue with this kind of DDC method is the lack of the stability and analysis methodologies.

3.2.2. DDC methods with unknown controller structures

3.2.2.1. Apparent DDC methods. This kind of DDC method means that the controller design is depended only on measured plant I/O data apparently, and in essence, the plant model structure and the dynamics are implicitly involved in controller design. Their control system design and methods of theoretical analysis are similar to those of MBC designs. However, they are also of significance for this kind of control system because they are more robust when they are used in practice. The direct adaptive control and subspace predictive control methods are typical of this [66].

3.2.2.2. Model-free DDC methods. This kind of DDC method implies that the controller is designed directly and merely using the measured plant I/O data, without explicitly or implicitly using model information. This is the ideal type of DDC method. The outstanding features of this kind of DDC methods is that it has a systematic controller design framework and systematic means of analyzing stability. ILC and MFAC are typical of this type of DDC method. The main difference between this kind of DDC method and others is that the effectiveness or rationality of the controller structure or controller designing is theoretically guaranteed using rigorous mathematics. This strategy can deal with the system control problems using a uniform way both for linear and nonlinear systems.

4. Relationships between MBC and DDC approaches and among DDC approaches

Each control method, whether it is a MBC or DDC method, has its own advantages and disadvantages in practice. MBC methods have a strong ability to control plants when accurate models are available and they also have systematic design and analysis tools. DDC methods have a better performance when the plant models are not available but they lack systematic designing procedures and means of analysis. In this subsection, we introduce some ways of designing a complementary, modularized control system incorporating either MBC and DDC methods or more than one DDC methods.

4.1. Relationship between MBC and DDC

The relationships between using the other MBC and DDC methods can be understood by viewing adaptive control and MFAC as an example [62].

4.1.1. MFAC-based estimated-type modularized controller design

Suppose a controlled system as follows:

$$y(k+1) = f(y(k), \ldots, y(k-n_y), u(k), \ldots, u(k-n_u)),$$

where $y(k)$ and $u(k)$ are the output and input of the controlled plant at instant k, n_y and n_u are the unknown orders of output and input, and $f(\cdots)$ is a general nonlinear function.

A control engineer usually builds a low order linear model, then designs a controller for the controlled nonlinear system based on this simplified model. If the first or second order model is used, such as, the first order model

$$y(k+1) = [y(k), u(k)]\begin{bmatrix}\theta_1(k)\\ \theta_2(k)\end{bmatrix} = \boldsymbol{\phi}^T(k)\boldsymbol{\theta}(k),$$

or the second order model

$$y(k+1) = [y(k), y(k-1), u(k), u(k-1)]\begin{bmatrix}\theta_1(k)\\ \theta_2(k)\\ \theta_3(k)\\ \theta_4(k)\end{bmatrix} = \boldsymbol{\phi}^T(k)\boldsymbol{\theta}(k).$$

Then a control engineer can design the corresponding controllers as follows: for the first order model

$$u(k) = \frac{1}{\hat{\theta}_2(k)}(y^*(k+1) - \hat{\theta}_1(k)y(k)),$$

and for the second order model

$$u(k) = \frac{1}{\hat{\theta}_3(k)}(y^*(k+1) - \hat{\theta}_1(k)y(k) - \hat{\theta}_2(k)y(k-1) - \hat{\theta}_4(k)u(k-1)). \tag{57}$$

where $\boldsymbol{\theta}(k)$ is a parameter vector of the system model, and $\hat{\boldsymbol{\theta}}(k)$ is the estimated value of $\boldsymbol{\theta}(k)$.

From the design process shown above, we cannot expect the adaptive control system to give a good performance due to the existing unmodeled dynamics NL.

$$y(k+1) = [y(k), u(k)]\begin{bmatrix}\theta_1(k)\\ \theta_2(k)\end{bmatrix} + NL = \boldsymbol{\phi}^T(k)\boldsymbol{\theta}(k) + NL,$$

$$y(k+1) = [y(k), y(k-1), u(k), u(k-1)]\begin{bmatrix}\theta_1(k)\\ \theta_2(k)\\ \theta_3(k)\\ \theta_4(k)\end{bmatrix} + NL = \boldsymbol{\phi}^T(k)\boldsymbol{\theta}(k) + NL.$$

If we could find a method of estimating the unmodeled dynamics NL and design a compensation algorithm that would take this unmodeled dynamics into the controller, then we could expect that this modularized control system would lead to much better improvement.

Based on the MFAC dynamic linearization technique discussed in Section 3, the estimation of modularized unmodeled dynamics can be designed as follows. The block diagram is shown in Fig. 13.

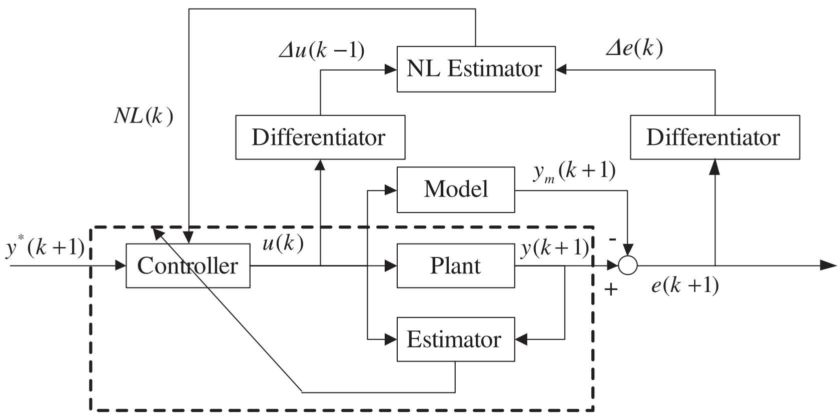

Fig. 13. MFAC-based estimated-type modularized controller design.

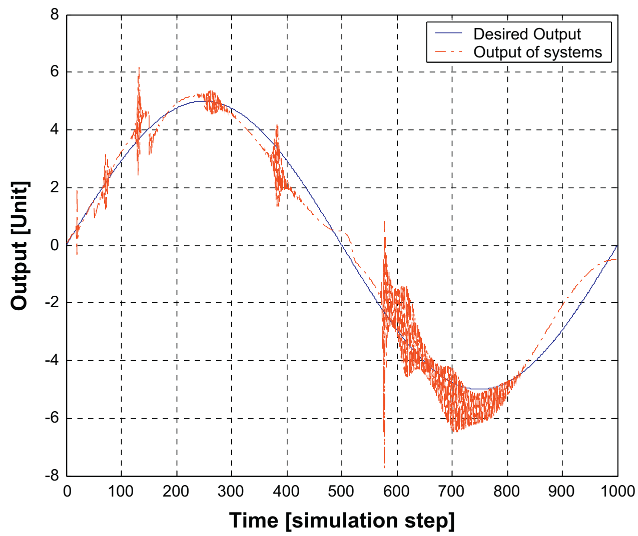

Fig. 14. Performance of adaptive control based on second order model.

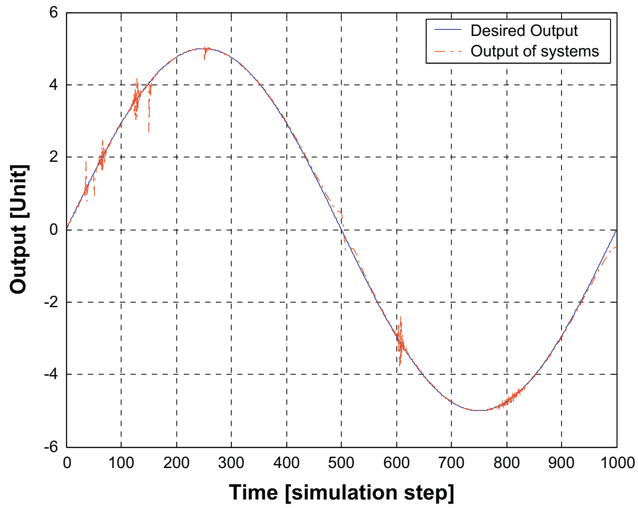

Fig. 15. Performance of MFAC-based estimated-type modularized controller design.

The detailed estimation algorithm for the NL is as follows:

$$\begin{aligned}&\hat{\phi}(k)=\hat{\phi}(k-1)+\tfrac{\eta\Delta u(k-1)}{\mu+\Delta u(k-1)^2}\times(\Delta e(k)-\hat{\phi}(k-1)\Delta u(k-1)),\\&N\hat{L}(k)=e(k-1)+\hat{\phi}(k-1)\Delta u(k-1).\end{aligned}\tag{58}$$

This estimation algorithm can be used to foster a complementary modularized control system. The adaptive control law with an unmodeled dynamic compensation algorithm for the first order model is as follows:

$$u(k)=\frac{1}{\hat{\theta}_2(k)}(y^*(k+1)-\hat{\theta}_1(k)y(k)-N\hat{L}),$$

Similar to the first-order model case, the adaptive control law with unmodeled dynamic compensation algorithm for the second order model is as follows:

$$u(k)=\frac{1}{\hat{\theta}_3(k)}(y^*(k+1)-\hat{\theta}_1(k)y(k)-\hat{\theta}_2(k)y(k-1)-\hat{\theta}_4(k)u(k-1)-N\widehat{L}).\tag{59}$$

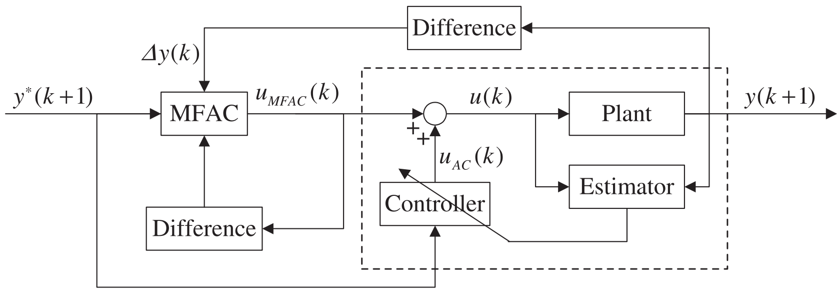

Fig. 16. Diagram of MFAC-based embedded-type modularized controller design.

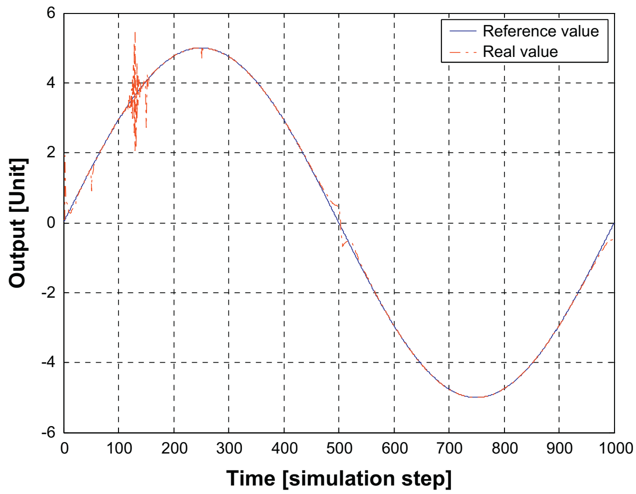

Fig. 17. Control performance of MFAC based embedded-type modularized controller design.

The following example illustrates the performance of a MFAC-based estimated-type modularized controller. The controlled plant is described as follows:

$$y(t+1) = \frac{y(t)}{1+y(t)^2} + a(t)u(t)^3 + 0.2y(t-1),$$
$$a(t) = 0.1 + 0.1^{round(t/100)}, \quad t = 1, 2, \ldots, 1000.$$

The control performance using adaptive controller (57) is shown in Fig. 14 and the results of using an MFAC-based estimated-type modularized controller (58) and (59) are shown in Fig. 15.

4.1.2. MFAC-based embedded-type modularized controller design

The plant controlled by an adaptive control can be regarded as a new augmented controlled plant. In this way, we can use MFAC to control this new augmented system because the MFAC method does not require any system information. This can be realized in the following inter-loop and out-loop configurations shown in Fig. 16. The actual control input is calculated as follows:

$$u(k) = u_{existing}(k) + u_{mfac}(k),$$

where $u_{mfac}(k)$ and $u_{existing}(k)$ are the control input signals at time instant k for the MFAC controller and the existing feedback controller, respectively.

From Fig. 16, we can see that the external loop (MFAC) and the existing internal loop (adaptive control) can work independently or jointly in a complementarily manner.

Simulation results using this MFAC-based embedded-type modularized controller design for the same example as subsection 4.1.1 is shown in Fig. 17.

4.1.3. Switching mechanism

The switching connected configuration, shown in Fig. 18, is another strategy for incorporating DDC and MBC methods. This configuration is based on the philosophy that the MBC or DDC control method each has itself merit to each other,

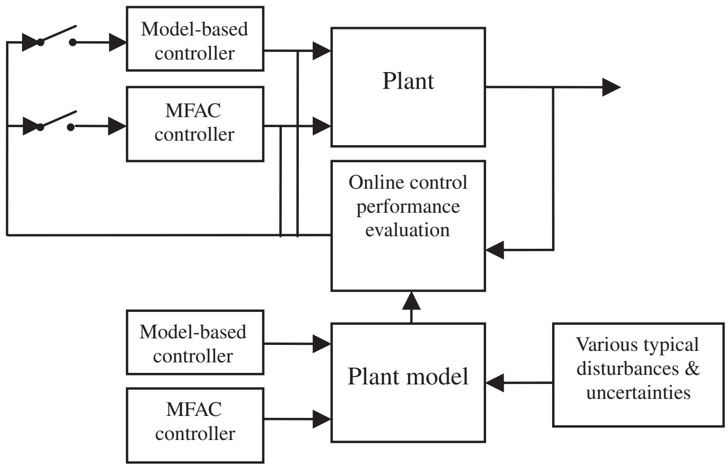

Fig. 18. Switched mechanism between MBC and DDC methods.

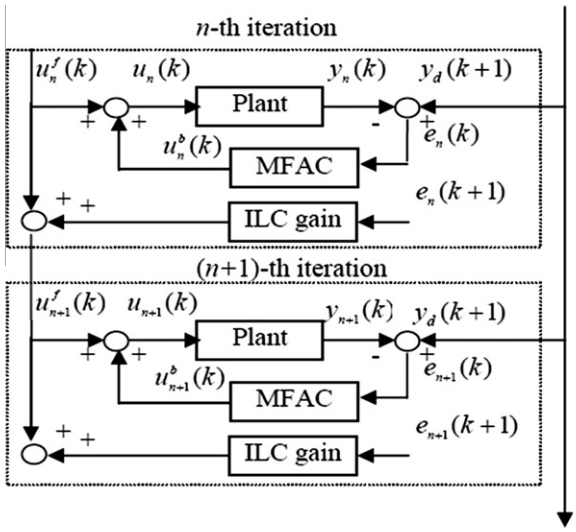

Fig. 19. Modularized controller design of MFAC and ILC.

and the dynamics of the controlled plant may change during the operation of the controlled process. Therefore, which control system takes on the dominant role will depend on the control performance, as established by an online control performance evaluation criterion. Realizing this mechanism in practice would be of great significance.

4.1.4. Other mechanisms

In philosophy, one recognizes that, the more accurate knowledge on information about the controlled plant we have, the better control strategy we can design. Thus, how to utilize the information available about the controlled plant in the DDC controller design is another issue that needs to be addressed. This topic is discussed in Section 5.

4.2. Relationship among different DDC approaches

4.2.1. Modularized controller design

In this subsection, we will use two typical DDC methods, ILC and MFAC, as examples for discussion on the modularized controller design. If a controlled plant is operated in a repeatable manner, as the robot arms in an assembly line and batch-to-batch running processes, that is, the controlled process has a feature of repeatability, then we can utilize this characteristic to design a better controller to execute the control task using following configuration shown in Fig. 19.

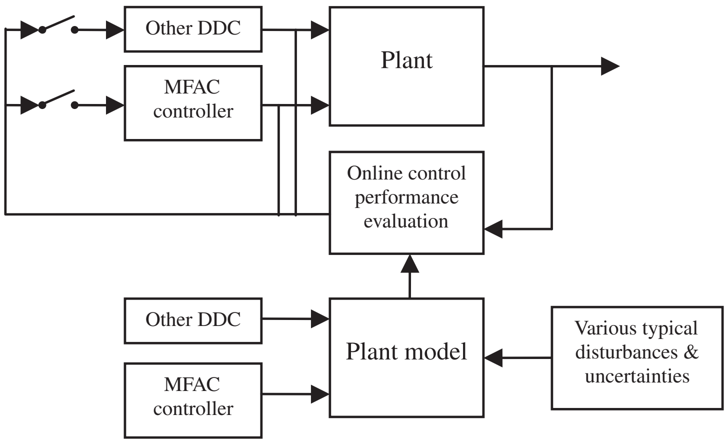

Fig. 20. Switched mechanism among different DDC methods.

This configuration can be written in a mathematical form as follows [61]

$$\begin{aligned}
&u_n(k) = u_n^f(k) + u_n^b(k),\\
&u_n^f(k) = u_{n-1}^f(k) + \beta e_{n-1}(k+1),\\
&u_n^b(k) = u_n^b(k-1) + \frac{\rho\hat{\phi}_n(k)}{\lambda + |\hat{\phi}_n(k)|^2} \times [y_d(k+1) - y_n(k)],\\
&\hat{\phi}_n(k) = \hat{\phi}_n(k-1) + \frac{\eta\Delta u_n^b(k-1)}{\mu + |u_n^b(k-1)|^2} \times \left[\Delta y_n(k) - \hat{\phi}_n(k-1)\Delta u_n^b(k-1)\right],\\
&\hat{\phi}_n(k) = \hat{\phi}_n(1), \quad \text{if } \hat{\phi}_n(k) \leqslant \varepsilon \quad \text{or } |\Delta u_n^b(k-1)| \leqslant \varepsilon.
\end{aligned}$$

The stability of this kind of modularized control system has been analyzed in [56–59,61]. Other kinds of strategies regarding modularized control system design can also be found in the other literature.

4.2.2. Switching mechanism

There is no universal controller, and each DDC strategy has merits and drawbacks. When selecting a control method for a given task, we can design a switching mechanism using different DDC methods. This is reminiscent of the relationship between the MBC and DDC methods described above, the diagram is shown in Fig. 20.

5. Data-driven optimization and modeling

5.1. Data-driven optimization

For any DDC approach, the partial derivative or gradient information of system output with respect to control input is crucial to the controller design. The way to calculate

$$\frac{\partial y(k+1)}{\partial u(k)}$$

creates labels for the control methods. Stochastic approximation leads to the SPSA-based model-free control. Using the projection or least squares algorithm gives the MFAC method. Using iterative optimization can yield IFT, and using batch optimization on the collection of I/O data pairs makes the VRFT. Even for the quite different ILC method in apparentness, different designing methods of this information lead to different labels for the ILC control laws. For MBC methods, this information is available because the plant model is known. In this situation, the main issue in controller design for MBC methods in essence becomes an optimization problem in some sense. For DDC methods, obtaining this information is crucial because the plant model is not known, instead, large amounts of measured I/O data from the controlled plant available. How to obtain this information using these huge data to design the DDC controller becomes the vital difficulty for the DDC methods. Thus, the data driven optimization theory and methods are the fundamental mathematics basis for the DDC.

Table 1
Features of existing typical data-driven control methods.

Title	SPSA	MFAC	UC	PID	IFT	VRFT	ILC	LL
Initiator	• J.C. Spall • 1993 • America	• Z.S. Hou • 1994 • China	• M.G. Safonov • 1995 • America	• J.G. Ziegler • 1942 • America	• H. Hjalmarsson • 1994 • Sweden	• G.O. Guardabassi • 2000 • Italy	• M. Uchiyama • 1978 • Japan	• S. Schaal • 1994 • America
Plant	• Nonlinear	• Nonlinear	• Nonlinear	• LTI	• LTI	• LTI	• Nonlinear	• Nonlinear
Assumption	• Structure of NN is known • Global *Lipschitz* condition	• Generalized *Lipschitz* condition	• Controller set are required • Performance specification • Controller is Inverse	• Unknown	• Controller structure is fixed • The closed system is stable	• Controller structure is fixed • Reference model is inverse, not equal to 1	• Identical initial condition • Global *Lipschitz* condition	• Unknown
Feature	• Estimate gradient using SPSA • Parameter identification in essential • Adaptive control • Heavy computation • Slow convergence • Lack of stability analysis	• Dynamic linearization at each operating point • Gradient estimation • Adaptive control both for parameter and system structure • Low computation • Modularizable	• Switch control in essential • Adaptive control • Controller structure is fixed.	• Controller structure is fixed • Nonadaptation • Low computation • Lack of stability analysis	• Gradient estimation using trial signal. • Parameter identification in essential. • Nonadaptation • Heavy computation • Difficult to modularized design • Lack of stability analysis	• Parameter identification in essential. • Nonadaptation • Heavy computation • Difficult to modularized design • Lack of stability analysis	• Integrator in iteration domain • Low computation. • Theory is almost perfect • Modularized design • 2D system	• Linearization at every operating point • Adaptation • Low computation • Lack of stability analysis
Data	• Online data	• Online data	• Offline data and online data	• Offline data	• Offline data	• Offline data	• Offline data and online data	• Offline data and online data
Trial signal	• Twice trials at every iteration	• None	• None	• None	• Twice trials at every iteration • Collection two group data with length *N*	• None	• None	• None
Frame	• None	• Systematic dynamic linearization methods for SISO, MISO and MIMO systems • Systematic controller design methods • Systematic results of convergence and stability	• Systematic controller design methods • Systematic results of convergence and stability	• None	• Extension to MIMO system • Extension to nonlinear systems • Partial results of convergence and stability	• Extension to MIMO system • Extension to nonlinear systems	• Systematic methods for SISO, MISO and MIMO systems • Systematic controller design methods • Systematic results of convergence and stability	• None
Application	• Wastewater treatment • Traffic signal control	• Freeway traffic control • Moulding control • Motor control • Industrial process • Economic prediction • Welding control • Etc.	• Missile • Robot manipulator • Industrial process • Etc.	• 95% industrial processes	• Motor • Robot • Beam and ball • Magnetic levitation system • Temperature system • Etc.	• Robot manipulator • Motion control	• Refer to [3]	• Robot control

5.2. Data-driven modeling

MBC theory is based on the assumption that the plant model is known. However, accurate plant models are often unavailable. Thus, the causal model-based feedback controller,

$$u(t) = c(e(t+1), e(t), \ldots, e(t-n_y), u(t-1), \ldots, u(t-n_u), \boldsymbol{\rho}),$$

may lead to unsatisfactory control performance due to the inaccuracy of the plant model because the tracking error $e(t+1)$ is calculated using the inaccurate plant model, and the controller parameter $\boldsymbol{\rho}$ is also obtained by using optimization algorithms on this model.

Under the assumption that the controller structure is known and fixed, like that of IFT, VRFT, and UC, the question of how to determine the predicted $y(t+1)$ in tracking error $e(t+1)$ for controller implementation is an obstacle for the DDC methods. In this case, one remedy or an intuitive strategy is that modeling a data model for the plant, which is just a relationship among the data without any physical meaning, like equivalent dynamic linearization data models in MFAC. With this equivalent data model, the output of the plant can be predicted, the tracking error $e(t+1)$ can be determined, then the controller can be implemented. It is noted that the DDC controller designing with the help of data-driven modeling to the closed-loop plant will remove the influence of the unmodeled dynamics since the controller is independent to the plant model. Without the assumption that the controller structure is known, as in MFAC, ILC, or SPSA, the reasonability of an ideal controller structure based on the data driven model, which indicate that the controller is capable of controlling the plant to track a given trajectory with some tunable controller parameters, are guaranteed by rigorous mathematical theory.

As long as the controller structure is independent of the plant dynamics model and the reasonability of an ideal controller structure is guaranteed by rigorous mathematical theory, and the controller is not designed according to a given plant model, then conventional unmodeled dynamics and traditional robustness do not exist. This indicates that the control system may be safer and more reliable than the model based methods. Whether the controller is designed according to a given plant model is the key difference between the MBC and DDC methods.

Given that the controller structure is independent of the plant model, it is necessary to predict system's real one-step-ahead output for the controller parameter tuning with some optimization algorithm. For the system output prediction, theoretically speaking, any existing prediction methods can serve as the predictor. These methods include the ones of data-based, dynamics model based, rule-based model, neural networks based, and so on. Thus, data-based modeling is of great significance for the healthy development of DDC theories.

6. Conclusion and perspective

The definition, classification, relevant topics, and the state-of-the-art of the existing algorithms of DDC methods are briefly surveyed and discussed. The differences and relationships between the MBC and DDC methods, among different DDC methods are also presented with appropriate insightful comments. After some short conclusions are listed in next, a few possible prospective research topics will be highlighted in follows.

(1) Theoretically speaking, ILC, SPSA, UC, and MFAC methods are designed for the control problems of the nonlinear systems, and IFT and VRFT are proposed for linear systems although they can be extended to the nonlinear systems.
(2) SPSA, MFAC, UC, and LL are adaptive, but other methods are nonadaptive. However, the adaptation ability of SPSA may be affected by variations of the plant structure or parameters.
(3) SPSA, IFT, VRFT, and UC (elliptical and gradient-based UC) are controller parameter identification approaches. VRFT is a one-shot direct identification method and the others are iteration identification methods.
(4) Both MFAC and LL are based on dynamic linearization. However, MFAC has a systematic dynamic linearization framework, and a series of controller designing strategies with compact-mapping-like stability analysis for SISO and MIMO nonlinear systems.
(5) Except for PID, ILC and VRFT, the other DDC methods need to estimate the gradient using I/O measurement data. SPSA, IFT and gradient based UC need to calculate the gradient of cost function with respect to controller parameter off-line, and dynamic linearization based MFAC and LL, however, need to estimate the gradient of the output change with respect to the control input change on-line at each time instant.
(6) SPSA, UC, and MFAC use the online measured I/O data. PID, IFT, and VRFT use offline I/O measured data. ILC and LL use both. It is worth noting that ILC uses the on-line and off-line I/O measurement data systematically, but other DDC methods do not. Another outstanding difference is that ILC directly approximates the control input signal rather than tunes controller parameters for asymptotic tracking of the output trajectory.
(7) ILC has the perfect systemic frame both for controller design and performance analysis. MFAC has features similar to those of ILC.

Detailed comparisons of DDC methods are shown in Table 1.

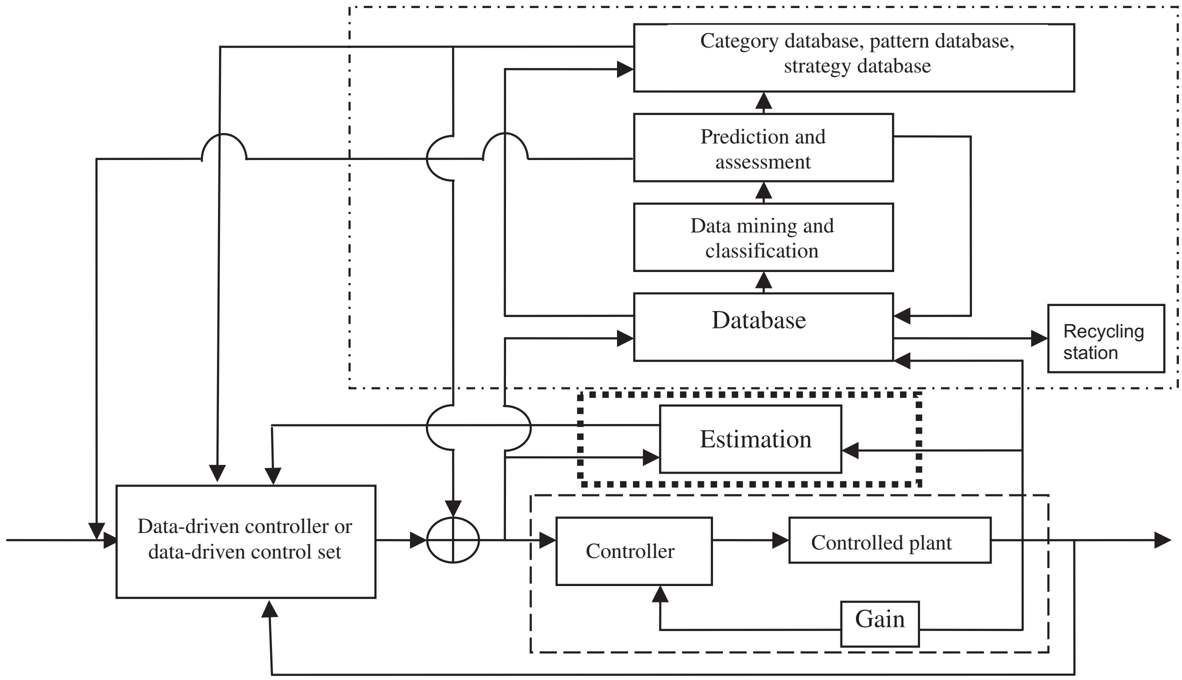

Fig. 21. Promising designing block diagram of DDC system.

Almost all DDC methods are designed using controller parameter tuning approaches except for ILC. Some of them involve on-line tuning, such as MFAC, UC, and SPSA. Some involve off-line tuning. The key point of the DDC methods is that the controller structure does not depend on the plant model. Some of these methods assume the controller structure a priori. MFAC and LL go one step further in that the controller structure of MFAC and LL is based on the theoretically supported dynamic linearization data model. This leads to a question of how to determine the controller structure. Sometimes, the difficulty in determining controller structure for a given plant is equivalent to creating an accurate plant model. Moreover, the problem of parameter tuning is an optimization issue in mathematics, and the optimization in DDC controller design is quite different from the traditional optimization because the system model in DDC controller design is unknown. From this point of view, MFAC, SPSA, and IFT have developed a technique to calculate or estimate the gradient information when the objective function is unknown. MFAC and IFT use a deterministic approach, while SPSA uses a stochastic approximation approach.

Although there exist a series of DDC methods in literature, the DDC theory is still in its embryonic stage. The perspective of DDC theory and associated promising research topics are briefly discussed as follows:

(1) The theoretical framework of DDC methods needs to be established. All DDC methods target to address the same control problem, that is, how to design a controller only using the measurement I/O data of the controlled plant to drive the plant output to track an expected output signal when the plant model is not available. However, these methods were developed independently. It may be possible to establish one or a few fundamental uniform architectures or frameworks for all these DDC methods. Promising architectures and frameworks for DDC theory may come from the identification of controller parameters, data-driven optimization, or dynamic linearization.
(2) For any control theory, the theoretical results, typical approaches and tools of analysis are fundamental to the growth of the discipline, so is DDC theory. Approaches and tools of analysis of the stability and convergence, and the stability conclusion itself, are the most important issues in DDC theory. Because DDC theory requires only I/O measurement data, approaches and tools of analysis should be model independent. The data-driven optimization theory and certain *Lyapunov*-like data energy techniques may be of more concern. Developing new typical analysis method, like the *Lyapunov* method in MBC system design is of great significance, and the new DDC analytical methods should be different in essence from those of traditional model-based control. In this respect, the stability analysis methods of ILC and MFAC may serve as examples for the other DDC methods.
(3) Highly efficient data processing methods and their applications in DDC may be promising [32,151]. There are many off-line data processing methods, such as data mining algorithms, feature extraction algorithms, pattern recognition algorithms, machine learning algorithms, and statistical analysis algorithms. These could be adapted for use in DDC research, and the existing hardware techniques are capable of supplying the computational ability to realize these off-line algorithms online. As the on-line and off-line data contains a great deal of valuable information regarding

system operations and the system patterns, thus, finding a way to use that information and those patterns to design a powerful DDC controller would be of great significance. Using the knowledge abstracted from both off-line and on-line data for controller design poses a significant challenge. One possible configuration for such a DDC method is shown in Fig. 21.

(4) The robustness issues in DDC theory. In MBC theory, robustness refers to the ability of a control system to deal with uncertainties or unmodeled dynamics. However, there is no unmodeled dynamics in DDC method. Hence, a new definition of robustness must be created for DDC. In practice, the data may be contaminated by external disturbances or lost due to a failure of the sensors, actuators, or network. For this reason, we believe that the study of the robustness of DDC should focus on the influences of data noise and data dropouts.

(5) The system operation assessment and prediction based on data are the other important issues. In the model-based method, assessment and prediction are fulfilled based on model information. The unmodeled dynamics may result in the fallibility of assessment and prediction as well as of stability. Thus designing reliable DDC controllers in practice would be of great significance to DDC theory. In fact, data-driven methodologies may include various technologies, which can use the data directly for the implementation of various desired system functions, such as, data-driven decision making, data-driven prediction, data-driven performance assessment, and data-driven fault diagnosis, etc.

(6) DDC and MBC methods have advantages and disadvantages. The development of complementary configurations involving both merits needs to be studied further. The novel modularized controller design and other kinds of complimentary strategies for DDC and MBC should also be addressed.

(7) Data-driven optimization theory and methods, and data-driven modeling are two fundamental theoretical bases for the DDC theory. However, little work has been done on these two topics, especially the data-driven optimization theory. More emphasis should be put on these two research topics.

(8) Applications of DDC in typical industrial processes are also significant studies.

Acknowledgments

This work was supported by the National Science Foundation of China under Grants 60834001 and 61120106009. The first author of this paper would like to thank the National Science Foundation of China for their invitation to present talks on this topic in seminars held in November 2008 and November 2010. These two talks form the substance of this survey paper.

References

[1] D.W. Aha, D. Kibler, M. Albert, Instance-based learning algorithms, Machine Learning 6 (1) (1991) 37–66.
[2] D.W. Aha, Editorial: lazy learning, Artificial Intelligence Review 11 (1–5) (1997) 1–6.
[3] H.S. Ahn, Y.Q. Chen, K.L. Moore, Iterative learning control: brief survey and categorization, IEEE Transactions on Systems, Man, and Cybernetics – Part C: Applications and Reviews 37 (6) (2007) 1099–1121.
[4] A. Al-Tamimi, F.L. Lewis, M. Abu-Khalaf, Model-free Q-learning designs for linear discrete-time zero-sum games with application to H-infinity control, Automatica 43 (3) (2007) 473–481.
[5] P. Albertos, A. Sala, Iterative Identification and Control, Springer-Verlag, London, UK, 2002.
[6] B.D.O. Anderson, Failures of adaptive control theory and their resolution, Communications in Information and Systems 5 (1) (2005) 1–20.
[7] B.D.O. Anderson, A. Dehghani, Historical, generic and current challenges of adaptive control, in: Proc. of Third IFAC Workshop on Periodic Control Systems, Anichkov Palace, Russia, 2007.
[8] B.D.O. Anderson, A. Dehghani, Challenges of adaptive control-past, permanent and future, Annual Reviews in Control 32 (2008) 123–135.
[9] S. Arimoto, S. Kawamura, F. Miyazaki, Bettering operation of robots by learning, Journal of Robotic Systems 1 (2) (1984) 123–140.
[10] K.J. Astrom, T. Hagglund, A. Wallenborg, Automatic Tuning of PID Controllers, Instrument Society of America, North Carolina, 1988.
[11] K.J. Astrom, T. Hagglund, C.C. Hang, W.K. Ho, Automatic tuning and adaptation for PID controllers – a survey, Control Engineering Practice 1 (4) (1993) 699–714.
[12] K.J. Astrom, T. Hagglund, PID Controllers: Theory Design and Tuning, second ed., Instrument Society of America, North Carolina, 1995.
[13] C.G. Atkeson, A.W. Moore, S. Schaal, Locally weighted learning for control, Artificial Intelligence Review 11 (1–5) (1997) 75–113.
[14] S. Baldi, G. Battistelli, E. Mosca, P. Tesi, Multi-model unfalsified adaptive switching supervisory control, Automatica 46 (2) (2010) 249–259.
[15] G. Battistelli, J. Hespanha, E. Mosca, P. Tesi, Unfalsified adaptive switching supervisory control of time varying systems, in: Proc. of the 48th IEEE Conference on Decision and Control and Held Jointly with the 28th Chinese Control Conference, Shanghai, China, 2009, pp. 805–810.
[16] G. Battistelli, E. Mosca, M.G. Safonov, P. Tesi, Stability of unfalsified adaptive switching control in noisy environments, IEEE Transactions on Automatic Control 55 (10) (2010) 2424–2429.
[17] D.P. Bertsekas, J.N. Tsitsiklis, Neuro-Dynamic Programming, Athena Scientific, MA, 1996.
[18] G. Bontempi, M. Birattari, H. Bersini, Lazy learning for modeling and control design, International Journal of Control 72 (7/8) (1999) 643–658.
[19] G. Bontempi, M. Birattari, From linearization to lazy learning: a survey of divide-and-conquer techniques for nonlinear control, International Journal of Computational Cognition 3 (1) (2005) 56–73.
[20] A. Boulkroune, M. M' Saad, H. Chekireb, Design of a fuzzy adaptive controller for MIMO nonlinear time-delay systems with unknown actuator nonlinearities and unknown control direction, Information Sciences 180 (2010) 5041–5059.
[21] S.J. Bradtke, B.E. Ydstie, A.G. Barto, Adaptive linear quadratic control using policy iteration, in: Proc. of the 1994 American Control Conference, Baltimore, USA, 1994, pp. 3475–3479.
[22] M.W. Braun, D.E. Rivera, A. Stenman, A model-on-demand identification methodology for nonlinear process systems, International Journal of Control 74 (18) (2001) 1708–1717.
[23] X.H. Bu, Z.S. Hou, The robust stability of model free adaptive control with data dropouts, in: Proc. of the 8th IEEE International Conference on Control and Automation (ICCA), Xiamen, China, 2010, pp. 1606–1611.
[24] X.H. Bu, Z.S. Hou, S.T. Jin, A statistical analysis of model free adaptive control with measurement disturbance, in: Proc. of the 29th Chinese Control Conference (CCC), Beijing, China, 2010, pp. 2175–2181.

[25] M.C. Campi, A. Lecchini, S.M. Savaresi, Virtual reference feedback tuning: a direct method for the design of feedback controllers, Automatica 38 (8) (2002) 1337–1346.
[26] M.C. Campi, A. Lecchini, S.M. Savaresi, An application of the virtual reference feedback tuning (VRFT) method to a benchmark active suspension system, European Journal of Control 9 (2003) 66–76.
[27] M.C. Campi, S.M. Savaresi, Direct nonlinear control design: the virtual reference feedback tuning (VRFT) approach, IEEE Transactions on Automatic Control 51 (1) (2006) 14–27.
[28] Y. Cang, B. Gao, K.Y. Gu, A model-free adaptive control to a blood pump based on heart rate, ASAIO Journal 57 (4) (2011) 262–267.
[29] T.Y. Chai, D.R. Liu (Eds.), Special Issue on Data-driven Based Control, Decision, Scheduling and Fault Diagnosis, Acta Automatica Sinica 35 (6) (2009).
[30] T.Y. Chai, Z.S. Hou, F.L. Fewis, A. Hussain (Eds.), Special Section on Data-Based Control, Modeling, and Optimization, IEEE Transactions on Neural Networks 22 (12) (2011).
[31] C.J. Chen, A discrete iterative learning control for a class of nonlinear time-varying systems, IEEE Transactions on Automatic Control 43 (5) (1998) 748–752.
[32] L. Chen, L.J. Zou, L. Tu, A clustering algorithm for multiple data streams based on spectral component similarity, Information Sciences 183 (2012) 35–47.
[33] Y.Q. Chen, Iterative Learning Control: Convergence, Robustness and Applications, Springer-Verlag, New York, 1999.
[34] Q.M. Cheng, Y.M. Cheng, M.M. Wang, Y.F. Wang, Simulation study on model-free adaptive control based on grey prediction in ball mill load control, Chinese Journal of Scientific Instrument 32 (1) (2011) 87–92.
[35] R.H. Chi, Z.S. Hou, Dual-stage optimal iterative learning control for nonlinear non-affine discrete-time systems, Acta Automatica Sinica 33 (10) (2007) 1061–1065.
[36] R.H. Chi, Z.S. Hou, Model-free periodic adaptive control for a class of SISO nonlinear discrete-time systems, in: Proc. of the 8th IEEE International Conference on Control and Automation (ICCA), Xiamen, China, 2010, pp. 1623–1628.
[37] L.S. Coelho, A.A.R. Coelho, Model-free adaptive control optimization using a chaotic particle swarm approach, Chaos, Solitons and Fractals 41 (4) (2009) 2001–2009.
[38] Y. Fujisaki, Y. Duan, M. Ikcda, System representation and optimal control in input-output data space, in: Proc. of the 10th IFAC Symposium on Large Scale Systems, Osaka, Japan, 2004, pp. 197–202.
[39] B. Gao, K.Y. Gu, Y. Zeng, Y. Chang, An anti-suction control for an intra-aorta pump using blood assistant index: a numerical simulation, Artificial Organs (2011).
[40] M. Gevers, Modelling, identification and control, in: P. Albertos, A. Sala (Eds.), Iterative Identification and Control Design, Springer-Verlag, 2002, pp. 3–16.
[41] A.E. Graham, A.J. Young, S.Q. Xie, Rapid tuning of controllers by IFT for profile cutting machines, Mechatronics 17 (2–3) (2007) 121–128.
[42] G.O. Guardabassi, S.M. Savaresi, Virtual reference direct design method: an off-line approach to data-based control system design, IEEE Transactions on Automatic Control 45 (5) (2000) 954–959.
[43] W.H. Gui, Data-driven based nonferrous metallurgical process optimal control research and application, in: Technical Report of the 33rd Shuangqing Forum of National Natural Science Foundation of China on Data-driven Based Control, Decision, Scheduling and Fault Diagnosis, Beijing, China, 2008.
[44] G. Gybenko, Just-in-time learning and estimation, in: S. Bittanti, G. Picci (Eds.), Identification, Adaptation, Learning: The Science of Learning Models From Data, Springer, New York, 1996, pp. 423–434.
[45] R. Hildebrand, A. Lecchini, G. Solari, M. Gevers, Prefiltering in iterative feedback tuning: optimization of the prefilter for accuracy, IEEE Transactions on Automatic Control 49 (10) (2004) 1801–1806.
[46] R. Hildebrand, A. Lecchini, G. Solari, M. Gevers, Optimal prefiltering in iterative feedback tuning, IEEE Transactions on Automatic Control 50 (8) (2005) 1196–1200.
[47] H. Hjalmarsson, S. Gunnarsson, M. Gevers, A convergent iterative restricted complexity control design scheme, in: Proc. of the 33rd IEEE Conference on Decision and Control, Orlando, USA, 1994, pp. 1735–1740.
[48] H. Hjalmarsson, Control of nonlinear systems using iterative feedback tuning, in: Proc. of the 1998 IEEE American Control Conference, Philadephia, USA, 1998, pp. 2083–2087.
[49] H. Hjalmarsson, M. Gevers, S. Gunnarsson, O. Lequin, Iterative feedback tuning: theory and applications, IEEE Control Systems Magazine 18 (4) (1998) 26–41.
[50] H. Hjalmarsson, Efficient tuning of linear multivariable controllers using iterative feedback tuning, International Journal of Adaptive Control and Signal Processing 13 (7) (1999) 553–572.
[51] H. Hjalmarsson, Iterative feedback tuning-an overview, International Journal of Adaptive Control and Signal Processing 16 (5) (2002) 373–395.
[52] Z.S. Hou, The Parameter Identification, Adaptive Control and Model Free Learning Adaptive Control for Nonlinear Systems, PhD dissertation, Northeastern University, Shengyang, China, 1994.
[53] Z.S. Hou, W.H. Huang, The model-free learning adaptive control of a class of SISO nonlinear systems, in: Proc. of the 1997 IEEE American Control Conference, Albuquerque, USA, 1997, pp. 343–344.
[54] Z.S. Hou, C.W. Han, W.H. Huang, The model free learning adaptive control of a class of MISO nonlinear discrete time systems, in: Proc. of the 1998 IFAC Low Cost Automation, Shanghai, China, 1998, pp. 227–232.
[55] Z.S. Hou, Nonparametric Models and Its Adaptive Control Theory, Science Press, Beijing, 1999.
[56] Z.S. Hou, On model-free adaptive control: the state of the art and perspective, Control Theory and Applications 23 (4) (2006) 586–592.
[57] Z.S. Hou, J.X. Xu, A new feedback–feedforward configuration for the iterative learning control of a class of discrete-time systems, Acta Automatica Sinica 33 (3) (2007) 323–326.
[58] Z.S. Hou, J.X. Xu, H.W. Zhong, Freeway traffic control using iterative learning control based ramp metering and speed signaling, IEEE Transactions on Vehicular Technology 56 (2) (2007) 466–477.
[59] Z.S. Hou, J.X. Xu, J.W. Yan, An iterative learning approach for density control of freeway traffic flow via ramp metering, Transportation Research Part C 16 (1) (2008) 71–97.
[60] Z.S. Hou, J.X. Xu, On data-driven control theory: the state of the art and perspective, Acta Automatica Sinica 35 (6) (2009) 650–667.
[61] Z.S. Hou, J.W. Yan, Model free adaptive control based freeway ramp metering with feed-forward iterative learning controller, Acta Automatica Sinica 35 (5) (2009) 588–595.
[62] Z.S. Hou, J.W. Yan, Convergence analysis of learning-enhanced PID control system, Control Theory & Applications 27 (6) (2010) 761–768.
[63] Z.S. Hou, X.H. Bu, Model free adaptive control with data dropouts, Expert Systems with Applications 38 (8) (2011) 10709–10717.
[64] Z.S. Hou, S.T. Jin, A novel data-driven control approach for a class of discrete-time nonlinear systems, IEEE Transactions on Control Systems Technology 19 (6) (2011) 1549–1558.
[65] Z.S. Hou, S.T. Jin, Data driven model-free adaptive control for a class of MIMO nonlinear discrete-time systems, IEEE Transactions on Neural Networks 22 (12) (2011) 2173–2188. Special Issue on Data-based control, modeling, and optimization.
[66] B. Huang, R. Kadali, Dynamic Modeling, Predictive Control and Performance Monitoring: A Data-Driven Subspace Approach, Springer, London, 2008.
[67] S. Hur, M. Park, H. Rhee, Design and application of model-on-demand predictive controller to a semibatch copolymerization reactor, Industrial & Engineering Chemistry Research 42 (4) (2003) 847–859.
[68] J.K. Huusom, N.K. Poulsen, S.B. Jørgensen, Improving convergence of iterative feedback tuning, Journal of Process Control 19 (4) (2009) 570–578.
[69] M. Ikeda, Y. Fujisaki, N. Hayashi, A model-less algorithm for tracking control based on input–output data, Nonlinear Analysis 47 (3) (2001) 1953–1960.
[70] R.E. Kalman, A new approach to linear filtering and prediction problems, transactions ASME, series D, Journal of Basic Engineering 82 (1960) 34–45.

[71] R.E. Kalman, Contributions to the theory of optimal control, Boletin de la Sociedad Matematica Mexicana 5 (2009) 102–119.
[72] L.C. Kammer, R.R. Bitmead, P.L. Bartlett, Direct iterative tuning via spectral analysis, Automatica 36 (9) (2000) 1301–1307.
[73] Y. Kansha, Y. Hashimoto, M.-S. Chiu, New results on VRFT design of PID controller, Chemical Engineering Research and Design 86 (8) (2008) 925–931.
[74] A. Karimi, L. Miskovic, D. Bonvin, Convergence analysis of an iterative correlation-based controller tuning method, in: Proc. of the 15th IFAC World Congress, Barcelona, Spain, 2002, pp. 1546–1551.
[75] A. Karimi, L. Miskovic, D. Bonvin, Iterative correlation-based controller tuning with application to a magnetic suspension system, Control Engineering Practice 11 (6) (2003) 1069–1078.
[76] A. Karimi, K. Van Heusden, D. Bonvin, Non-iterative data-driven controller tuning using the correlation approach, in: Proc. of the European Control Conference, Kos, Greece, 2007.
[77] T. Katayama, Subspace Methods For System Identification, Springer, Heidelberg, 2005.
[78] J.H. Kim, F.L. Lewis, Model-free H[infinity] control design for unknown linear discrete-time systems via Q-learning with LMI, Automatica 46 (8) (2010) 1320–1326.
[79] S. Kissling, P. Blanc, P. Myszkorowski, I. Vaclavik, Application of iterative feedback tuning (IFT) to speed and position control of a servo drive, Control Engineering Practice 17 (7) (2009) 834–840.
[80] M. Kobayashi, Y. Konishi, H. Ishigaki, A lazy learning control method using support vector regression, International Journal of Innovative Computing Information and Control 3 (6B) (2007) 1511–1523.
[81] H.N. Koivo, J.T. Tanttu, Tuning of PID controllers: survey of SISO and MIMO techniques, in: Proc. of the IFAC Intelligent Tuning and Adaptive Control symposium, Singapore, 1991, pp. 75–80.
[82] T.Y. Kuc, J.S. Lee, K. Nam, An iterative learning control theory for a class of nonlinear dynamic systems, Automatica 28 (6) (1992) 1215–1221.
[83] W. Larimore, Statistical optimality and canonical variable analysis system identification, Signal Processing 52 (2) (1996) 131–144.
[84] J.M. Lee, J.H. Lee, Approximate dynamic programming-based approaches for input-output data-driven control of nonlinear processes, Automatica 41 (7) (2005) 1281–1288.
[85] J.K. Lim, J.M. Lim, K. Yoshimoto, K.H. Kim, T. Takahashi, Designing guide-path networks for automated guided vehicle system by using the Q-learning technique, Computers & Industrial Engineering 44 (1) (2003) 1–17.
[86] L. Ljung, System Identification: Theory for the User, Prentice-Hall, Englewood Cliffs, 1987.
[87] X.P. Lu, W. Li, Y.G. Lin, Load control of wind turbine based on model free adaptive controller, Transactions of the Chinese Society of Agricultural Machinery 42 (2) (2011) 109–114.
[88] F.L. Lv, H.B. Chen, C.J. Fan, S.B. Chen, A novel control algorithm for weld pool control, Industrial Robot: An International Journal 37 (1) (2010) 89–96.
[89] I. Markovsky, J. Willems, P. Rapisarda, B. De Moor, Data driven simulation with applications to system identification, in: Proc. of the 16th IFAC World Congress, Prague, Czech Republic, 2005.
[90] I. Markovsky, J. Willems, S. Van Huffel, B. De Moor, Exact and approximate modeling of linear systems: a behavioral approach, in: R. Haberman (Ed.), Monographs on Mathematical Modeling and Computation, SIAM Society for Industrial and Applied Mathematics, Philadelphia, 2006.
[91] I. Markovsky, P. Rapisarda, Data-driven simulation and control, International Journal of Control 81 (12) (2008) 1946–1959.
[92] L. Miskovic, A. Karimi, D. Bonvin, Correlation-based tuning of a restricted-complexity controller for an active suspension system, European Journal of Control 9 (1) (2003) 77–83.
[93] L. Miskovic, Data-Driven Controller Tuning Using The Correlation Approach, PhD dissertation, University of Belgrade, 2006.
[94] L. Miskovic, A. Karimi, D. Bonvin, M. Gevers, Correlation-based tuning of decoupling multivariable controllers, Automatica 43 (9) (2007) 1482–1494.
[95] M. Moonen, B.D. Moor, L. Vandenberghe, J. Vandewalle, On-and off-line identification of linear state-space models, International Journal of Control 49 (1) (1989) 219–232.
[96] K.L. Moore, Iterative Learning Control for Deterministic Systems, Springer-Verlag, New York, 1993.
[97] M. Nakamoto, An application of the virtual reference feedback tuning for an MIMO process, in: The SICE 2004 Annual Conference, 2004, pp. 2208–2213.
[98] P.V. Overschee, B.D. Moor, Subspace Identification for Linear Systems: Theory Implementation Applications, Kluwer Academic Publishers, Dordrecht, 1996.
[99] T.H. Pan, S.Y. Li, W.J. Cai, Lazy learning-based online identification and adaptive PID control: a case study for CSTR process, Industrial & Engineering Chemistry Research 46 (2) (2007) 472–480.
[100] T.H. Pan, S.Y. Li, A hierarchical search and updating database strategy for lazy learning, International Journal of Innovative Computing Information and Control 4 (6) (2008) 1383–1392.
[101] S.E. Papadakis, V.G. Kaburlasos, Piecewise-linear approximation of non-linear models based on probabilistically/possibilistically interpreted intervals' numbers (INs), Information Sciences 180 (24) (2010) 5060–5076.
[102] K.H. Park, Y.J. Kim, J.H. Kim, Modular Q-learning based multi-agent cooperation for robot soccer, Robotics and Autonomous Systems 35 (2) (2001) 109–122.
[103] U.S. Park, M. Ikeda, Stability analysis and control design of LTI discrete-time systems by the direct use of time series data, Automatica 45 (5) (2009) 1265–1271.
[104] R.E. Precup, S. Preitl, I.J. Rudas, M.L. Tomescu, J.K. Tar, Design and experiments for a class of fuzzy controlled servo systems, ASME Transactions on Mechatronics 13 (1) (2008) 22–35.
[105] F. Previdi, T. Schauer, S.M. Savaresi, K.J. Hunt, Data-driven control design for neuroprotheses: a virtual reference feedback tuning (VRFT) approach, IEEE Transactions on Control Systems Technology 12 (1) (2004) 176–182.
[106] F. Previdi, M. Ferrarin, S.M. Savaresi, S. Bittanti, Closed-loop control of FES supported standing up and sitting down using virtual reference feedback tuning, Control Engineering Practice 13 (9) (2005) 1173–1182.
[107] F. Previdi, F. Fico, D. Belloli, S.M. Savaresi, I. Pesenti, C. Spelta, Virtual Reference Feedback Tuning (VRFT) of velocity controller in self-balancing industrial manual manipulators, in: Proc. of the American Control Conference (ACC), Baltimore, MD, 2010, pp. 1956–1961.
[108] C.E. Rohrs, L. Valavani, M. Athans, G. Stein, Robustness of adaptive control algorithms in the presence of unmodeled dynamics, in: Proc. of the 21st IEEE Conference on Decision and Control, Orlando, USA, 1982, pp. 3–11.
[109] C.E. Rohrs, L. Valavani, M. Athans, G. Stein, Robustness of continuous-time adaptive control algorithms in the presence of unmodeled dynamics, IEEE Transactions on Automatic Control 30 (9) (1985) 881–889.
[110] M.G. Safonov, T.C. Tsao, The unfalsified control concept: a direct path from experiment to controller, in: B.A. Francis, A.R. Tannenbaum (Eds.), Feedback Control, Nonlinear Systems and Complexity, Springer-Verlag, Berlin, 1995, pp. 196–214.
[111] M.G. Safonov, T.C. Tsao, The unfalsified control concept and learning, IEEE Transactions on Automatic Control 42 (6) (1997) 843–847.
[112] M.G. Safonov, Data-driven robust control design: unfalsified control, 2003. <http://routh.usc.edu/pub/safonov/safo03i.pdf>.
[113] A. Sala, A. Esparza, Extensions to virtual reference feedback tuning: a direct method for the design of feedback controllers, Automatica 41 (8) (2005) 1473–1476.
[114] A. Sala, Integrating virtual reference feedback tuning into a unified closed-loop identification framework, Automatica 43 (1) (2007) 178–183.
[115] S. Schaal, C.G. Atkeson, Robot juggling: implementation of memory-based learning, IEEE Control Systems Magazine 14 (1) (1994) 57–71.
[116] C. Schaper, W. Larimore, D. Seborg, D. Mellichamp, Identification of chemical processes using canonical variable analysis, Computers and Chemical Engineering 18 (1) (1994) 55–69.
[117] J. Sivag, A. Datta, S.P. Bhattacharyya, New results on the synthesis of PID controllers, IEEE Transactions on Automatic Control 47 (2) (2002) 241–252.
[118] J. Sjoberg, M. Agarwal, Nonlinear controller tuning based on linearized time-variant model, in: Proc. of the 1997 American Control Conference, 1997, pp. 3336–3340.

[119] J. Sjoberg, F. De Bruyne, M. Agarwal, B.D.O. Anderson, M. Gevers, F.J. Kraus, N. Linard, Iterative controller optimization for nonlinear systems, Control Engineering Practice 11 (9) (2003) 1079–1086.
[120] J. Sjoberg, P.-O. Gutman, M. Agarwal, M. Bax, Nonlinear controller tuning based on a sequence of identifications of linearized time-varying models, Control Engineering Practice 17 (2) (2009) 311–321.
[121] R.E. Skelton, Model error concepts in control design, International Journal of Control 49 (5) (1989) 1725–1753.
[122] W.D. Smart, L.P. Kaelbling, Practical reinforcement learning in continuous spaces, in: Proc. of the 17th International Conference on Machine Learning, Stanford, CA, 2000, pp. 903–910.
[123] J.C. Spall, SPSA Algorithm, <www.jhuapl.edu/spsa/>.
[124] J.C. Spall, Multivariate stochastic approximation using a simultaneous perturbation gradient approximation, IEEE Transactions on Automatic Control 37 (3) (1992) 332–341.
[125] J.C. Spall, J.A. Cristion, Model-free control of general discrete-time systems, in: Proc. of the 32rd IEEE Conference on Decision and Control, San Antonio, USA, 1993, pp. 2792–2797.
[126] J.C. Spall, D.C. Chin, Traffic-responsive signal timing for system-wide traffic control, Transportation Research Part C 5 (3–4) (1997) 153–163.
[127] J.C. Spall, J.A. Cristion, Model-free control of nonlinear stochastic systems with discrete-time measurements, IEEE Transactions on Automatic Control 43 (9) (1998) 1198–1210.
[128] J.C. Spall, Adaptive stochastic approximation by the simultaneous perturbation method, IEEE Transactions on Automatic Control 45 (10) (2000) 1839–1853.
[129] J.C. Spall, Feedback and weighting mechanisms for improving Jacobian estimates in the adaptive simultaneous perturbation algorithm, IEEE Transactions on Automatic Control 54 (6) (2009) 1216–1229.
[130] M.X. Sun, B.J. Huang, Iterative Learning Control, National Defence Industry Press, Beijing, 1998.
[131] K.K. Tan, T.H. Lee, S.N. Huang, F.M. Leu, Adaptive predictive control of a class of SISO nonlinear systems, Dynamics and Control 11 (2) (2001) 151–174.
[132] M. Uchiyama, Formulation of high-speed motion pattern of a mechanical arm by trial, Control Engineering 14 (6) (1978) 706–712.
[133] J. Van Helvoort, Unfalsified Control: Data-Driven Control Design for Performance Improvement, PhD dissertation, Technische Universiteit Eindhoven, Eindhoven, Netherlands, 2007.
[134] J. Van Helvoort, B. de Jager, M. Steinbuch, Direct data-driven recursive controller unfalsification with analytic update, Automatica 43 (12) (2007) 2034–2046.
[135] J. Van Helvoort, B. de Jager, M. Steinbuch, Data-driven multivariable controller design using ellipsoidal unfalsified control, Systems & Control Letters 57 (9) (2008) 759–762.
[136] J. Van Helvoort, B. de Jager, M. Steinbuch, Data-driven controller unfalsification with analytic update applied to a motion system, IEEE Transactions on Control Systems Technology 16 (6) (2008) 1207–1217.
[137] K. Van Heusden, A. Karimi, D. Bonvin, Data-driven controller tuning with integrated stability constraint, in: Proc. of the 47th IEEE Conference on Decision and Control, Cancun, Mexico, 2008, pp. 2612–2617.
[138] K. Van Heusden, Non-Iterative Data-Driven Model Reference Control, PhD dissertation, Ecole Polytechnique Federale de Lausanne, Lausanne, Switzerland, 2010.
[139] M. Verhaegen, Identification of the deterministic part of MIMO state space models given in innovations form from input-output data, Automatica 30 (1) (1994) 61–74.
[140] H.B. Wang, Q.F. Wang, Nonparametric model adaptive control for underwater towed heave compensation system, Control Theory & Application 27 (4) (2010) 513–516.
[141] R.R. Wang, M.G. Safonov, The Comparison of Unfalsified Control and Iterative Feedback Tuning, University of Southern California, USA, 2002.
[142] R.R. Wang, A. Pau, M. Stefanovic, M.G. Safonov, Cost detectability and stability of adaptive control systems, International Journal of Robust and Nonlinear Control 17 (5–6) (2007) 549–561.
[143] W. Wang, J.T. Zhang, T.Y. Chai, A survey of advanced PID parameter tuning methods, Acta Automatica Sinica 26 (3) (2000) 347–355.
[144] C. Watkins, Learning From Delayed Rewards, PhD dissertation, Cambridge University, Cambridge, England, 1989.
[145] C. Watkins, P. Dayan, Q-learning, Machine Learning 8 (1992) 279–292.
[146] A. Weissensteiner, A Q-learning approach to derive optimal consumption and investment strategies, IEEE Transactions on Neural Networks 20 (8) (2009) 1234–1243.
[147] P.J. Werbos, A menu of designs for reinforcement learning over time, in: W.T. Miller III, R.S. Sutton, P.J. Werbos (Eds.), Neural Networks for Control, MIT Press, MA, 1991, pp. 67–95.
[148] P.J. Werbos, Approximate dynamic programming for real-time control and neural modeling, in: D.A. White, D.A. Sofge (Eds.), Handbook of Intelligent Control, Van Nostrand Reinhold, New York, 1992.
[149] J.X. Xu, Linear and Nonlinear Iterative Learning Control, Springer Verlag, Berlin, 2003.
[150] J.X. Xu, Z.S. Hou, On learning control: the state of the art and perspective, Acta Automatica Sinica 31 (6) (2005) 943–955.
[151] J.X. Xu, Z.S. Hou, Notes on data-driven system approaches, Acta Automatica Sinica 35 (6) (2009) 668–675.
[152] S. Yabui, K. Yubai, J. Hirai, Direct design of switching control system by VRFT-application to vertical-type one-link arm, in: Proc. of the 2007 SICE Annual Conference, Kagawa, Japan, 2007, pp. 120–123.
[153] B. Zhang, W.D. Zhang, Adaptive predictive functional control of a class of nonlinear systems, ISA Transactions 45 (2) (2006) 175–183.
[154] Z.J. Zhou, C.H. Hu, D.L. Xu, J.B. Yang, D.H. Zhou, New model for system behavior prediction based on belief rule based systems, Information Sciences 180 (2010) 4834–4864.
[155] J.G. Ziegler, N.B. Nichols, Optimum settings for automatic controllers, Transactions of the ASME 64 (1942) 759–768.