Active Online System Identification of Switch Mode DC–DC Power Converter Based on Efficient Recursive DCD-IIR Adaptive Filter

Maher Algreer, Matthew Armstrong, and Damian Giaouris

***Abstract*—This paper introduces a novel technique for online system identification. Specific attention is given to the parameter estimation of dc–dc switched-mode power converters; however, the proposed method can be applied to many alternative applications where efficient and accurate parameter estimation is required. The proposed technique is computationally efficient, based on a dichotomous coordinate descent algorithm, and uses an infinite impulse response adaptive filter as the plant model. The system identification technique reduces the computational complexity of existing recursive least squares algorithms. Importantly, the proposed method is also able to identify the parameters quickly and accurately, thus offering an efficient hardware solution that is well suited to real-time applications. Simulation analysis and validation based on experimental data obtained from a prototype synchronous dc–dc buck converter is presented. Results clearly demonstrate that the estimated parameters of the dc–dc converter are a very close match to those of the experimental system. The approach can be directly embedded into adaptive and self-tuning digital controllers to improve the control performance of a wide range of industrial and commercial applications.**

***Index Terms*—Adaptive filter, dichotomous coordinate descent (DCD), infinite impulse response (IIR) adaptive filter, recursive least squares (RLS), switch mode dc–dc power converter, system identification.**

I. Introduction

Many industrial and consumer devices rely on switched-mode power converters (SMPCs) to provide a reliable, well-regulated, dc power supply. A poorly performing power supply can potentially compromise the characteristic behavior, efficiency, and operating range of the device. To ensure accurate regulation of the SMPC, optimal control of the power converter output is required. However, SMPC uncertainties, such as component tolerances, unpredictable load changes, variation in ambient conditions, and ageing effects, affect the performance of the controller over time [1]–[3]. To compensate for these time-varying problems, there is now increasing interest in employing real-time adaptive control techniques in SMPC applications. Here, the controller tuning is based upon online system identification (parameter estimation) techniques and adjusted according to regular parameter updates.

Clearly, for a high-performance controller with good dynamic performance, accurate estimation of the system parameters is essential [4]. Normally, in digitally controlled systems, a discrete transfer function model of the plant is used for the control design [4], [5]. The actual form of the transfer function, and the numerical values of its coefficients, is dependent upon the individual parameters of the plant to be controlled. It is the fundamental role of the system identification process to evaluate each coefficient of the transfer function [6]. In many applications, it is very important that the coefficients are calculated as accurately as possible, since this will ultimately determine the closed-loop controller response. However, in SMPC applications, it is also necessary to acquire the system parameters rapidly. The time constants in pulse width modulation (PWM) switched power converters are often very short, and it is not uncommon for abrupt load changes to be observed. Any system identification scheme must be able to respond appropriately to these characteristics. However, achieving improved accuracy and/or speed also implies the need for a faster, more powerful microprocessor platform. This is not always viable in SMPC applications, especially small, high volume systems, where it is essential to keep system costs low and competitive. Therefore, there is a need for computationally light system identification schemes which enable these advanced techniques to be performed on lower cost hardware. This paper aims to address this issue by presenting a method of SMPC system identification.

II. System Identification Methods

When identifying the model of an unknown system, there are two system identification approaches that can be used: parametric and nonparametric estimation techniques [7]–[9]. Recent research demonstrates several productive parametric and nonparametric system identification techniques for power electronic converter applications. Nonparametric methods often use spectral analysis and correlation analysis to estimate the frequency response or impulse response of the system. The behavior of the system is then estimated from the frequency response without using any parametric modeling [7]–[9]. In SMPC applications, nonparametric methods often consider perturbing the duty cycle with a frequency rich input signal; for example, a pseudo random binary sequence (PRBS) [9]. Typically, Fourier transform

Manuscript received September 17, 2011; revised January 27, 2012; accepted March 1, 2012. Date of current version June 20, 2012. Recommended for publication by Associate Editor S. Williamson.

The authors are with the School of Electrical, Electronic and Computer Engineering, Newcastle University, Newcastle Upon Tyne, NE1 7RU, U.K. (e-mail: m.m.f.saber@ncl.ac.uk; matthew.armstrong@ncl.ac.uk; damian.giaouris@ncl.ac.uk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TPEL.2012.2190754

methods are then applied to find the frequency response of the system. Unfortunately, the identification process can take significant amounts of time to complete and may need to process long data sequences [2]. In addition, during the identification process, the system operates in open loop without regulation [10]. Therefore, Barkley and Santi [11] developed a technique where the discrete dynamic system can be identified without having to run open loop during the identification process. Following on from this, Roinila *et al*. [12] proposed a new method based on Fourier amplitude spectrum, using a maximum length PRBS to improve the sensitivity to disturbances in the system. While these methods are fairly straight forward to implement, designing a controller using nonparametric system identification methods is usually limited to frequency response methods only.

In parametric techniques, a model structure is proposed [9] and the parameters of the model are identified using information extracted from the system [7]. Therefore, in parametric identification, it is necessary to define the order and overall structure of the system model (number of poles, zeros) prior to estimating the plant [8]. The selected candidate model is always application dependent and its complexity is often subject to the approximations which can be made. For example, a dc–dc buck converter can be represented as a second-order infinite impulse response (IIR) filter [13]. This provides an “average model” of the converter and will characterize the basic operation of the system. It will not, however, show the PWM switching frequency component in the output voltage. Provided the switching behavior is not of immediate concern, the second order candidate model will suffice. Once the model has been chosen, several approaches can be used to identify the system parameters; for instance, least mean squares, recursive least squares (RLS), maximum likelihood, and subspace methods [7], [14]. Recursive identification methods are a very familiar approach in online applications. However, these methods, and in particular RLS, are not fully exploited in low-cost, low-power SMPCs due to the computational complexity of the identification algorithm, which may require a high-specification microprocessor to successfully implement. Clearly, this is not desirable from an industry point of view where minimal cost and low complexity are key design drivers. Zhenyu and Prodic [10] present a hardware-efficient online parametric estimation technique based on inserting limit cycle oscillations (LCO) [15] into the output response of the power converter during steady-state operation. Amplitude and frequency information are then extracted from the LCO signal to find the dc–dc converter parameters (corner frequency and quality factor) [16]. While hardware efficient, this method results in lower system identification accuracy [15].

Pitel and Krein [13] introduce a real-time parametric identification method using a classical RLS technique. It identifies the parameters of an open-loop buck converter during abrupt load changes from the control signal to inductor current transfer function. This study accurately estimates the parameters during initial start-up of the system, and during relatively slow load changes. Algreer *et al*. [6] successfully adopted a method from the telecommunication field (originally presented by Chang *et al*. [17]) using fuzzy variable forgetting factor RLS to identify abrupt load changes in a closed-loop dc–dc converter. Miao *et al*. [4] present a dual identification approach; here, parametric and nonparametric methods are combined to estimate the parameters of an SMPC and then to directly design a digital controller. Implementing two methods is clearly more complex and computationally heavy for online system identification purposes. Therefore, it is more suited to offline scenarios. Peretz and Bin-Yaakov [2] also propose an offline system identification approach to determine an open-loop, time-domain model of a dc–dc converter. The iterative least square minimization approach (Steiglitz and McBride method) is used to estimate system parameters and then to design the digital controller directly by the Ragazzini method [18], [19].

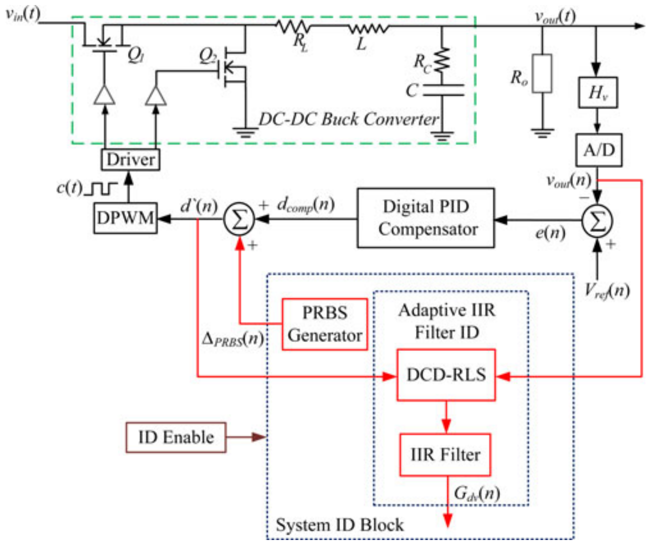

Fig. 1. Proposed closed-loop adaptive IIR identification method using DCD-RLS algorithm.

Unfortunately, in many of the methods presented, significant signal processing is required to implement these schemes and this ultimately has a cost penalty for the target application. Furthermore, the computational complexity impacts upon time of execution in the microprocessor, and this in turn makes it difficult to adopt in continuous parameter estimation for adaptive control applications [20]. For this reason, in this paper an RLS algorithm is implemented using a fast, computationally light, hardware efficient, adaptive algorithm, known as dichotomous coordinate descent (DCD) [21]. This algorithm has previously been developed for use in the field of telecommunications. Here, we adapt the algorithm and apply it for the first time in the system identification of power electronic circuits.

III. System Identification of DC–DC Converter Using Adaptive IIR/DCD-RLS Algorithm

Fig. 1 illustrates a block diagram of the proposed identification scheme. Here, a closed-loop dc–dc buck converter is controlled via a digital PID compensator. In addition, a real-time system identification algorithm is inserted alongside the controller, continually updating the parameters of a discrete model of the buck converter system on a sample by sample basis. The identification system can be enabled and disabled on demand during operation. For example, it may be applied at start-up, at regular set intervals, or enabled on detection of a system change such as a variation in the system load. Monitoring the voltage loop error is one simple way to detect a system change and enable the system identification process. When enabled, a small high-order excitation signal is injected into the control loop. This is required to improve the convergence time of the adaptive filter; this is the time to obtain optimal filter tap weights for accurate parameter estimation. For all online identification methods, some form of system perturbation is essential for the estimation process [22]. In this scheme, the PRBS is selected. As shown in Fig. 1, the PRBS signal is added to the PID controller output signal, $d_{\rm comp}(n)$. This creates a control signal $d'(n)$ with a superimposed persistent excitation component. Once applied to the PWM, a small disturbance in the output duty cycle $c(t)$ is generated. In this way, the duty cycle command signal at steady state will vary between $d_{\rm comp}(n) \pm \Delta_{\rm PRBS}(n)$. Here, the steady-state duty cycle is 0.33 and the magnitude of PRBS signal is given as $\Delta_{\rm PRBS} = \pm\ 0.025$; therefore, a change of approximately equal to 33% ± 2.5% in duty cycle signal will be observed. This will cause an excitation signal in the buck converter output voltage $v_{\rm out}(t)$. During this process, the excited output control signal and output voltage are sampled ($d'(n)$ and $v_{\rm out}(n)$ in Fig. 1). Practically, in order to focus the identification on the frequency range of interest and remove unwanted high-frequency measurement noise, the inputs to the DCD-RLS algorithm require filtering prior to identification. This can be accomplished by designing a digital low-pass, or bandpass, filter. In addition, offset in the input signals must be removed as the RLS algorithm assumes zero mean values in the input signals. In dc–dc SMPC applications, it is easier to remove offsets on a cycle-by-cycle basis from the input signals, where steady-state average values of the regulated output voltage and the average duty-cycle ratio are known. At each time instance, the average value of the input signal is directly subtracted from the excited signal. A low-pass filter can also be used to remove the offset from the input signals; however, this will add more computation to the overall system, which is not essential in the online system identification process. Once the samples have been processed, they are passed to the identification algorithm (DCD-RLS block in Fig. 1) to estimate the system parameters and update the discrete IIR filter model of the SMPC.

An adaptive filter can have different structures depending upon its application, which may be noise cancellation, signal prediction, or system identification [23], [24]. In this paper, we employ an adaptive IIR filter for system identification. An adaptive filter may be defined as a "self-designing" filter [23], where the filter coefficients are continuously varying until the objective function is achieved [24]. As shown in Fig. 2, the adaptive filter consists of two key components, the digital filter and the adaptive filter algorithm, which are used to vary the tap-weight coefficients in real time. In system identification, a major concern is minimizing the prediction error signal $e_p(n)$. Ideally, we want this signal to equal zero, indicating excellent parameter estimation. However, practical issues such as measurement errors, unwanted noise, quantization, and delay times make this difficult to achieve. By minimizing the prediction error signal, the output signal of the filter $\hat{y}(n)$ (estimated signal) approximately equals the output of the unknown system $d(n)$ (desired signal). Here, the desired signal is the sampled output voltage of the dc–dc converter. Based on this, we can write [23]

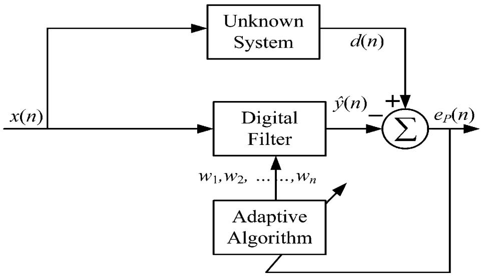

Fig. 2. Generic adaptive system identification block diagram.

$$\hat{y}(n) = \sum_{k=1}^{N} w_k x(n-k) = \mathbf{w}\mathbf{x} \tag{1}$$

$$\mathbf{w} = [\,w_1 \quad w_2 \quad \cdots \quad w_N\,]$$
$$\mathbf{x} = [\,x(n-1) \quad x(n-2) \quad \cdots \quad x(n-N)\,]^T \tag{2}$$

where the prefiltered input signal $x(n)$ is continuously adapted in response to the filter weight update [25]. The model of the unknown plant system (in this case, the dc–dc converter system) is defined by the transfer function of the adaptive filter. However, defining the digital filter coefficients requires analytical calculation of the linear system equations.

This can be achieved using Wiener equations but requires considerable computational effort [26]. Alternative methods, such as the Levinson–Durbin algorithm, can help to reduce the mathematical burden. Adaptive approaches can also be used to optimally calculate the tap weights and further trim the computational load [22], [23], [26]. Here, we employ an adaptive DCD-RLS algorithm to continuously adjust the filter coefficients and minimize $e_P(n)$. The error prediction is defined as [23]

$$e_P(n) = d(n) - \hat{y}(n) = d(n) - \sum_{k=1}^{N} w_k x(n-k). \tag{3}$$

According to (3), the error prediction signal is determined by applying the input signal to the digital filter to produce an estimation output signal $\hat{y}(n)$. The prediction error is then the difference between the desired signal, $d(n)$, and this generated estimation output signal [25]. When the prediction error is minimized, the adaptive filter tap weights reach steady state and no longer require updating. However, if any parameters of the plant change, the prediction error will deviate from the minimum point and the adaptive algorithm will start to determine the new filter tap weights in response to this change. To minimize the error signal, the adaptive algorithm must solve a series of linear equations to estimate the vector coefficients $\mathbf{w}$ where [27]

$$\mathbf{w} = \mathbf{R}^{-1}\boldsymbol{\beta}. \tag{4}$$

Here, $\mathbf{R}$ is an autocorrelation matrix of size $N \times N$, and $\boldsymbol{\beta}$ is an elements vector of length N. $\mathbf{R}$ and $\boldsymbol{\beta}$ are continually updated at each sample interval n. For the proposed DCD-RLS, $\mathbf{R}(n)$ and

$\boldsymbol{\beta}(n)$ may be described as follows [21]:

$$\mathbf{R}(n) = \lambda\mathbf{R}(n-1) + \mathbf{x}(n)\mathbf{x}^T(n)$$
$$\boldsymbol{\beta}(n) = \lambda\boldsymbol{\beta}(n-1) + d(n)\mathbf{x}(n). \qquad (5)$$

In (5), λ is a forgetting factor that applies a weighting to previously calculated elements of $\mathbf{R}$ and $\boldsymbol{\beta}$. When $\lambda = 1$, the system behaves like a classical RLS algorithm. $d(n)$ is the scalar desired signal, which relates the actual adaptive filter output estimation $\hat{y}(n)$ to the estimation error $e(n)$, according to $e(n) = d(n) - \hat{y}(n)$. Importantly, from a practical point of view, it is possible to find vector coefficients $\mathbf{w}$ without any mathematical division operations [21]. It will be shown in the following analysis that each divisor can be set as a power of two; thus, all division processes can conveniently be replaced with a computationally efficient shift right register.

IV. RLS and DCD Algorithm Theory

Least squares estimation techniques are fundamental in adaptive signal processing applications. In real-time applications [27], the solution is normally based on matrix inversion, which is computationally heavy and presents implementation difficulties. However, there are alternative algorithms for solving the linear equations expressed in (4). Amongst them, the DCD algorithm appears to be a particularly effective method [21], [27], [28]. Attractively, the computation is based on an efficient, fixed-point, iterative approach with no explicit division operations [28]. This makes it very appropriate for real-time hardware implementation. The computational requirement of the DCD algorithm depends mainly upon the number of iterations N_u used to update the parameters. The iteration number also determines the speed and accuracy of the process [27]. This section describes the operation of the DCD algorithm in an exponentially weighted RLS method.

A. Exponentially Weighted RLS Algorithm

Exponentially weighted recursive least squares (ERLS) is commonly used in dynamic systems to track time-varying parameters [23]. The ERLS algorithm minimizes the weighted sum of the squared error [21]

$$E_{\min}(n) = \lambda^{n+1}\mathbf{w}^T(n)\prod\mathbf{w}(n) + \sum_{k=0}^{n}\lambda^{n-k}[d(k) - \mathbf{w}^T(n)\mathbf{x}(k)]^2. \qquad (6)$$

Here, λ is again a forgetting factor, where $0 < \lambda < 1$. It ensures that past samples are gradually "forgotten" if the operating point of the system is constantly changing [23]. $\mathbf{\Pi}$ is a regulation matrix, usually selected as $\mathbf{\Pi} = \delta\times\mathbf{I}_N$. $\mathbf{I}_N$ is an $N \times N$ identity matrix, and δ is a small positive parameter (often referred to as the regulation parameter) [21]. At each sample, the ERLS solves the linear equation described in (4). Table I summarizes the steps to find the parameter vector $\mathbf{w}$ [21].

TABLE I
ERLS Algorithm

Step	Equation	×	+
	Initialize: $\hat{\mathbf{w}}(-1) = 0$, $\mathbf{r}(-1) = 0$, $\mathbf{R}(-1) = \mathbf{\Pi}$		
	for $n = 0,1,\ldots$		
1	$\mathbf{R}(n) = \lambda\,\mathbf{R}(n-1) + \mathbf{x}(n)\mathbf{x}^T(n)$	$2N^2$	N^2
2	$y(n) = \mathbf{x}^T(n)\hat{\mathbf{w}}(n-1)$	N	$N-1$
3	$e(n) = d(n) - y(n)$	–	1
4	$\boldsymbol{\beta}_o(n) = \lambda\,\mathbf{r}(n-1) + e(n)\mathbf{x}(n)$	$2N$	N
5	$\mathbf{R}(n)\Delta\mathbf{w}(n) = \boldsymbol{\beta}_o(n) \Rightarrow \Delta\hat{\mathbf{w}}(n), \mathbf{r}(n)$	M_n	A_n
6	$\hat{\mathbf{w}}(n) = \hat{\mathbf{w}}(n-1) + \Delta\hat{\mathbf{w}}(n)$	–	N

As mentioned earlier, direct methods require a complex matrix inversion operation to solve the linear equation in (4). However, in this method (first proposed by Zakharov *et al.* [21] in the field of communications), an alternative solution is presented by converting (4) into a sequence of auxiliary normal equations that can be solved using iterative techniques. First, at time instance $(n-1)$, the solution to the system equation $\mathbf{R}(n-1)\,\mathbf{w}(n-1) = \boldsymbol{\beta}(n-1)$ can be approximated; the approximate solution is $\hat{\mathbf{w}}(n-1)$. The residual vector of this solution can be written as [21]

$$\mathbf{r}(n-1) = \boldsymbol{\beta}(n-1) - \mathbf{R}(n-1)\hat{\mathbf{w}}(n-1). \qquad (7)$$

The system in (4) is then solved at each time instance n. From which

$$\Delta\mathbf{R}(n) = \mathbf{R}(n) - \mathbf{R}(n-1), \Delta\boldsymbol{\beta}(n) = \boldsymbol{\beta}(n) - \boldsymbol{\beta}(n-1),$$
$$\text{and } \Delta\mathbf{w}(n) = \mathbf{w}(n) - \hat{\mathbf{w}}(n-1). \qquad (8)$$

Then, by using the previous solution $\hat{\mathbf{w}}(n-1)$ and the residual vector $\mathbf{r}\,(n-1)$, a solution for $\hat{\mathbf{w}}(n)$ in (4) is found

$$\mathbf{R}(n)[\hat{\mathbf{w}}(n-1) + \Delta\mathbf{w}(n)] = \boldsymbol{\beta}(n). \qquad (9)$$

From (7)–(9) with respect to the unknown vector $\Delta\mathbf{w}$, the system equations can then be represented as [21]

$$\begin{aligned}\mathbf{R}(n)\Delta\mathbf{w}(n) &= \boldsymbol{\beta}(n) - \mathbf{R}(n)\hat{\mathbf{w}}(n-1)\\ &= \boldsymbol{\beta}(n) - \mathbf{R}(n-1)\hat{\mathbf{w}}(n-1) - \Delta\mathbf{R}(n)\hat{\mathbf{w}}(n-1)\\ &= \mathbf{r}(n-1) + \Delta\boldsymbol{\beta}(n) - \Delta\mathbf{R}(n)\hat{\mathbf{w}}(n-1). \qquad (10)\end{aligned}$$

From this, one can find a solution $\Delta\hat{\mathbf{w}}$ of the auxiliary system equations

$$\mathbf{R}(n)\Delta\mathbf{w}(n) = \boldsymbol{\beta}_o(n) \qquad (11)$$

where

$$\boldsymbol{\beta}_o(n) = \mathbf{r}(n-1) + \Delta\boldsymbol{\beta}(n) - \Delta\mathbf{R}(n)\hat{\mathbf{w}}(n-1). \qquad (12)$$

The approximate solution of the original system (4) can then be determined as

$$\hat{\mathbf{w}}(n) = \hat{\mathbf{w}}(n-1) + \Delta\hat{\mathbf{w}}(n). \qquad (13)$$

Considering (12), this approach requires vector $\mathbf{r}(n)$ to be known at each time instance n. However, it can be shown that the residual vector for the solution $\Delta\hat{\mathbf{w}}(n)$ to the auxiliary system

(4) is actually equal to $\mathbf{r}(n)$. Therefore [21]

$$\begin{aligned}\mathbf{r}(n) &= \boldsymbol{\beta}(n) - \mathbf{R}(n)\hat{\mathbf{w}}(n)\\ &= \boldsymbol{\beta}_o(n) - \mathbf{R}(n)\Delta\hat{\mathbf{w}}(n).\end{aligned} \tag{14}$$

At each time instance n, this approach requires a solution to an auxiliary problem (11) which deals with the increment of the filter weights $\Delta\mathbf{w}(n)$ rather than the actual filter weights $\mathbf{w}(n)$, as described in the original problem, (4) [21]. This approach is preferable since it takes into account the accuracy of the previous solution through the residual vector $\mathbf{r}(n-1)$, as well as the variation of the problem to currently be solved through the increments $\Delta\mathbf{R}(n)$ and $\Delta\boldsymbol{\beta}(n)$. It requires similar computational effort to the conventional method and will typically converge on an accurate solution within a small number of iterations [21]. In the ERLS algorithm, the autocorrelation matrix and cross-correlation are computed as in (5). The cross-correlation vector $\boldsymbol{\beta}_o(n)$ is expressed in terms of the filter inputs $\mathbf{x}(n)$ and the desired signal $d(n)$. By inserting (5) into (8), this yields [21]

$$\begin{aligned}\Delta\mathbf{R}(n) &= (\lambda-1)\mathbf{R}(n-1) + \mathbf{x}(n)\mathbf{x}^T(n)\\ \Delta\boldsymbol{\beta}(n) &= (\lambda-1)\boldsymbol{\beta}(n-1) + d(n)\mathbf{x}(n).\end{aligned} \tag{15}$$

From (7) and (15), we achieve

$$\Delta\mathbf{R}(n)\hat{\mathbf{w}}(n-1)=(\lambda-1)[\boldsymbol{\beta}(n-1)-\mathbf{r}(n-1)]+d(n)\hat{y}(n). \tag{16}$$

Then, based on (1) and (16), the vector $\boldsymbol{\beta}_o(n)$ can be described as

$$\boldsymbol{\beta}_o(n) = \lambda\mathbf{r}(n-1) + e_p(n)\mathbf{x}(n). \tag{17}$$

Table I also shows the computational effort of each step. The overall complexity of the algorithm can be shown to be $2N^2+3N+M_n$ multiplications and N^2+3N+A_n additions, where N is the filter order and M_n and A_n are the number of multiplications and additions required to solve the linear equation in step 5. These numbers depend significantly on the specific algorithms chosen to solve this particular step [21]. For example, matrix inversion lemma is one familiar technique to complete the division process in step 5. In this paper, we consider the use of the DCD algorithm to achieve a computationally light solution to solving this problem.

B. DCD Algorithm

In order to solve the linear equations in step 5 of Table I, the DCD algorithm typically offers a computationally efficient solution compared to many other iterative approaches [21], [27], [28]. It is based on an iterative approach to estimating N parameters within an estimation parameters vector $\Delta\hat{\mathbf{w}}$. The DCD algorithm begins to evaluate the residual vector and, based on its amplitude, will update the parameters vector. Initially, each parameter is assumed to reside within a defined amplitude range [–H, H]. The iteration step size d is chosen such that it equals H. Then, during each pass of the algorithm, the step size is halved ($d = d/2$, step 1). This divide-by-two process is very important from a hardware point of view. It allows a division operation to be replaced with a more computationally efficient shift register [28]. In fixed-point implementations, the filter coefficients are represented by a series of bits M. The exact representation (number of bits) depends on the accuracy required by the application. The algorithm starts the iterative search from the most significant bit M_b for each element in the parameters vector $\Delta\hat{\mathbf{w}}$. Once complete, the algorithm determines the next most significant bit, M_{b-1}, and so on until M_0. At this point, the binary representation of M is fully updated. In floating point implementations, a subtle variation is employed. Rather than updating individual bits, an iterative loop is configured with M iterations.

TABLE II
CYCLIC DCD ALGORITHM DESCRIPTION

Step	Equation	+
	Initialization: $\Delta\hat{\mathbf{w}} = 0, \mathbf{r} = \boldsymbol{\beta}_o,$ $d = H,\ k = 0$	
	for m = 1,..,M	
1	$d = d/2$	
2	Flag = 0	
	for n = 1...,N	
3	If $\lvert r_n\rvert > (d/2)R_{n,n}$	1
4	$\Delta\hat{w}_n = \Delta\hat{w}_n + sign\,(r_n)d$	1
5	$\mathbf{r} = \mathbf{r}\text{-}sign(r_n)d\,\mathbf{R}^{(n)}$	N
6	$k = k + 1$, $Flag = 1$	–
7	If $k > N_u$, algorithm stop	–
8	If Flag =1, repeat for step 2	–

Table II shows the operational steps of the cyclic DCD algorithm [21], [28]. Step 1: On each pass of the algorithm, the step size is reduced until the update is complete and the required level of accuracy is reached [28]. Steps 2 and 3: The magnitude of the residual vector $\mathbf{r}$ is analyzed during each iteration. Two outcomes are possible: 1) an unsuccessful iteration, where the condition set out in step 3 is not met. In this case, the step size is unchanged and the solution is not updated. 2) A successful iteration, where the condition in step 3 is met. Here, the step size is halved and the solution in steps 4 and 5 is updated [21]. Steps 4 and 5: If the residual is sufficiently large (Step 3: successful iteration), one element of the parameter vector is updated by adding or subtracting the value of d, depending upon the polarity of $\mathrm{r}_{n\,.}$ Following this, the residual vector $\mathbf{r}$ is updated (Step 5). The algorithm repeats this process until all elements in the residual vector $\mathbf{r}$ become small enough that the set condition in step 3 results in an unsuccessful iteration [28] or the number of iterations reaches a predefined limit number N_u [21]. The iteration limit may be used to control the execution time of the algorithm [22].

As shown in Table II, a major advantage of the fixed-point DCD algorithm is that both multiplication and division operations can be avoided. This is advantageous from a digital hardware implementation point of view. However, in the worst case, the number of additions is still $A_n = N(2N_u + M - 1) + N_u$ [21]. Therefore, if $N_u \gg M$, the complexity of the DCD can be approximated by $2NN_u$. However, if N_u is small and $N_u \ll M$, the term NM will dominate the DCD computational effort [21]. The dominant term will be application specific. Here, in the system

TABLE III
LEADING DCD ALGORITHM DESCRIPTION

Step	Equation	+
	Initialization: $\Delta\hat{\mathbf{w}} = 0$, $\mathbf{r} = \boldsymbol{\beta}_o$, $d = H$, $m = 1$	
	for $k = 1,\ldots, N_u$	
1	$n = arg\ max_{p=1,..,N}\ \{\lvert r_p\rvert\}$, go to step 4	$N-1$
2	$d = d/2$, $m = m + 1$	
3	if $m > M$, algorithm stops	
4	if $\lvert r_n\rvert \le (d/2)R_{n,n,}$ then go to step 2	1
5	$\Delta\hat{w}_n = \Delta\hat{w}_n + sign\,(r_n)d$	1
6	$\mathbf{r} = \mathbf{r} - sign(r_n)d\,\mathbf{R}^{(n)}$	N

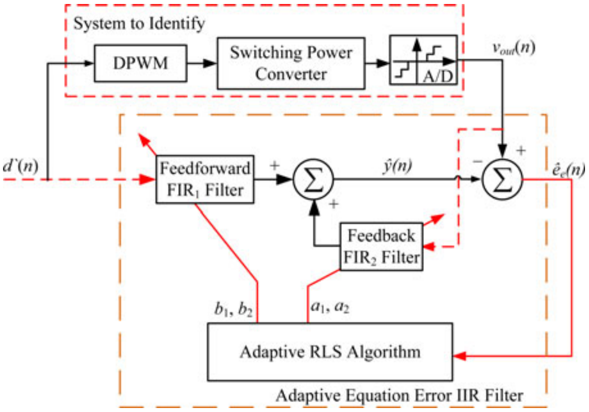

Fig. 3. System identification based adaptive IIR filter using equation error block diagram.

identification of a dc–dc converter, it is found that the second case is generally true, $N_u \ll M$. For this reason, we consider a refined form of the DCD algorithm (also proposed by Zakharov *et al*. [21]) to further trim the computational complexity-Leading DCD. In this particular version of the algorithm, it is possible to eliminate the *NM* dominant term. Table III summarizes the operational steps of the Leading DCD algorithm [21].

V. Discrete Modeling of DC–DC Converter and Adaptive IIR Filter

Discrete time modeling of an SMPC is essential for a parametric identification process. The primary candidate model for system identification in this paper is the voltage transfer function (control-to-output transfer function). Starting with the state-space equivalent model of the buck converter circuit in continuous current mode, it can be shown that the control signal $d'(s)$ to output voltage $v_{\text{out}}(s)$ transfer function is described as follows [29], [30]:

$$G_{dv}(s) = \frac{v_{\text{out}}(s)}{d`(s)}$$

$$= \frac{V_{\text{in}}\,(CR_c s + 1)}{s^2 LC\left(\frac{R_o+R_c}{R_o+R_L}\right) + s\left(R_c C + C\left(\frac{R_o R_L}{R_o+R_L}\right) + \frac{L}{R_o+R_L}\right) + 1} \tag{18}$$

where V_{in} is the input voltage, C is the output capacitance, L is output inductance, R_o is the load resistance, R_L is inductance equivalent resistance (ESR), and R_c is the capacitance ESR. The average continuous-time transfer function described in (18) can be converted to a discrete equivalent model using conventional continuous to discrete transformation methods, resulting in a second-order discrete transfer function

$$G_{dv}(z) = \frac{b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}. \tag{19}$$

Here, b_1, b_2, a_1, and a_2 are the parameters to be identified. They all depend on circuit component values and the sampling frequency. The input–output relation given in (19) may also be described as a linear difference equation. Several methods exist to obtain this [13], [14]. Here, an autoregressive-moving-average (ARMA) model is used, as it is a good match to the IIR filter form [23]. From this, it is possible to derive the following difference equation:

$$\begin{aligned} v_{\text{out}}(n) + a_1 v_{\text{out}}(n-1) + a_2 v_{\text{out}}(n-2) \\ = b_1 d'(n-1) + b_2 d'(n-2). \end{aligned} \tag{20}$$

In this paper, an IIR adaptive filter is employed to model the SMPC. However, the DCD-RLS algorithm described in Section IV is normally applied with FIR adaptive filters. For this reason, an equation error approach [23] is used whereby an IIR filter is effectively derived from an equation error structure of two FIR filters, as shown in Fig. 3. In the equation error structure, the input of the second FIR filter is assumed to be approximately equal to the desired signal, as illustrated in Fig. 3. The second FIR filter does not use past adaptive filter output samples as in the output error structure. Instead, it uses the delayed samples of the desired signal [23], [24]

$$\hat{y}(n) = \sum_{k=0}^{M} b_k(n) d'(n-k) + \sum_{k=1}^{N} a_k(n) v_{\text{out}}(n-k). \tag{21}$$

And the error output is defined as

$$\hat{e}_e(n) = v_{\text{out}}(n) - \hat{y}(n). \tag{22}$$

Practically, observing the input and output of the unknown system is required for the identification algorithm. Consequently, the same data vector that is used in the basic ARMA model structure system is observed in the equation error scheme. However, the update sequence for each FIR filter in Fig. 3 is not optimal using the DCD algorithm. Each filter requires an independent input data vector and adaptive algorithm to update a separate autocorrelation and cross-correlation matrix, as defined previously in (5). Accordingly, the overall complexity of the adaptive filter is increased. For this reason, we simplify by combining the input and output data from the unknown system into a single data and parameters vector (23). In order to match the ARMA model in (20) with the general form of IIR filter in (21), $N = M = 2$ and $b_0 = 0$. As a result, the data and parameters vectors can be described as

$$\boldsymbol{\varphi}(n) =$$

$$[\, v_{\text{out}}(n-1) \quad \cdots \quad v_{\text{out}}(n-k) \quad d'(n-1) \quad \cdots \quad d'(n-k)\,]$$

$$\mathbf{w} = [\, -a_1 \quad \cdots \quad -a_n \quad b_1 \quad \cdots \quad b_n \,]^T . \tag{23}$$

This allows the filter parameters to be arranged in a single correlation matrix

$$\mathbf{R}_1 = \begin{bmatrix} r_{d'}(0) & r_{d'}(1) \\ r_{d'}(1) & r_{d'}(0) \end{bmatrix}, \quad \mathbf{R}_2 = \begin{bmatrix} r_{v\text{out}}(0) & r_{v\text{out}}(1) \\ r_{v\text{out}}(1) & r_{v\text{out}}(0) \end{bmatrix},$$

$$\Downarrow$$

$$\mathbf{R}_{\text{T}} = \begin{bmatrix} r_{v\text{out}}(0) & r_{v\text{out}}(1) & r_{d'}(0) & r_{d'}(1) \\ r_{v\text{out}}(1) & r_{v\text{out}}(0) & r_{v\text{out}}(1) & r_{d'}(0) \\ r_{d'}(0) & r_{v\text{out}}(1) & r_{v\text{out}}(0) & r_{v\text{out}}(1) \\ r_{d'}(1) & r_{d'}(0) & r_{v\text{out}}(1) & r_{v\text{out}}(0) \end{bmatrix}. \tag{24}$$

Equation (24) represents the complete correlation matrix $\mathbf{R}_T$ of the equation error IIR filter; $\mathbf{R}_1$ shows the input correlation matrix of the feed-forward FIR filter and $\mathbf{R}_2$ represents the output correlation matrix of the feed-back FIR filter. In this simple way, a single DCD algorithm can successfully be used to update all tap weights of an IIR filter.

VI. Model Example and Simulation Results

Usually, system identification performance is measured using particular metrics such as convergence time, parameter accuracy, and prediction error [13]. These metrics determine how closely the identified model matches the actual system transfer function [6], and they are used to evaluate the proposed method in this paper. To test the concept of the proposed DCD-RLS identification scheme (see Fig. 1), a voltage controlled synchronous dc–dc buck SMPC circuit has been simulated using MATLAB/Simulink. The circuit parameters of the buck converter are $R_o = 5\ \Omega$, $R_L = 68\ \text{m}\Omega$, $R_c = 25\ \text{m}\Omega$, $L = 220\ \mu\text{H}$, $C = 330\ \mu\text{F}$, $V_o = 3.3$ V, $V_{\text{in}} = 10$ V, $H_v = 0.5$ (voltage sensor), and $R_{\text{dson}} = 3.5\ \text{m}\Omega$. The shunt resistance ($R_S = 5\ \text{m}\Omega$) is added to measure the inductor current; thus, the equivalent series resistance $R_q = R_L + R_S + R_{\text{dson}} = 76.5\ \text{m}\Omega$. The buck converter is switched at 20 kHz and the output voltage is also sampled at the same switching frequency rate. Consequently, the control-to-output voltage discrete transfer function of the SMPC can be calculated as follows:

$$G_{dv}(z) = \frac{0.226z^{-1} + 0.1118z^{-2}}{1 - 1.914z^{-1} + 0.949z^{-2}}. \tag{25}$$

For the exponentially weighted DCD-RLS algorithm, the parameters are as follows: $N_u = 1$, $H = 1$, $M = 8$, $\lambda = 0.95$, and $\delta = 0.001$. For completeness, the simulation model includes all digital effects, such as ADC quantization and sample/hold delays. To present the viability of the proposed DCD-RLS algorithm, an equivalent system based on a conventional exponentially weighted RLS (using matrix inversion lemma) is also simulated. This algorithm is summarized in Table IV [23]. In Table IV, $\varphi(n)$ is the data vector (regression vector), $\hat{\mathbf{w}}(n)$ is the estimated tap weights , $\varepsilon(n)$ is *a priori* error (prediction error), $\mathbf{P}(n)$ is an $N\times N$ inverse correlation matrix, $\mathbf{k}(n)$ is an $N\times 1$ adaptation gain vector, and λ is the forgetting factor [23]. The same settings and initial conditions are used for both DCD-RLS and conventional RLS algorithms.

TABLE IV
Exponentially Weighted Recursive Least Algorithm-Based Matrix Inversion Lemma

Step	Equation
	Initialization: $\hat{\mathbf{w}} = 0$, $\mathbf{P}(0) = \dfrac{1}{\delta}\mathbf{I}_N$
	for $n = 1, 2, \ldots$
1	$\mathbf{S}(n) = \mathbf{P}(n-1)\boldsymbol{\varphi}(n)$
2	$\mathbf{k}(n) = \dfrac{\mathbf{S}(n)}{\lambda + \boldsymbol{\varphi}^T(n)\mathbf{S}(n)}$
3	$\varepsilon(n) = d(n) - \hat{\mathbf{w}}^T(n-1)\boldsymbol{\varphi}(n)$
4	$\hat{\mathbf{w}}(n) = \hat{\mathbf{w}}(n-1) + \mathbf{k}(n)\varepsilon(n)$
5	$\mathbf{p}(n) = \dfrac{1}{\lambda}\Big[\, \mathbf{p}(n-1) - \mathbf{k}(n)\boldsymbol{\varphi}^T(n)\mathbf{P}(n-1) \Big]$

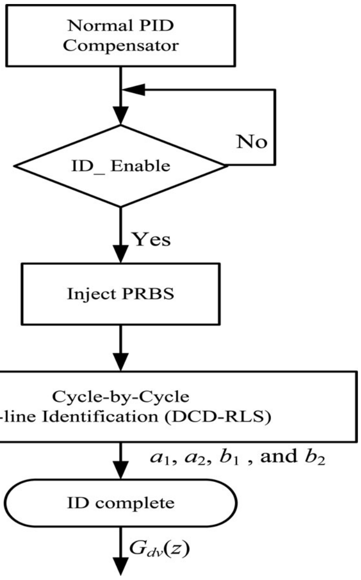
Normal PID Compensator
ID_ Enable
No
Yes
Inject PRBS
Cycle-by-Cycle On-line Identification (DCD-RLS)
a_1, a_2, b_1 , and b_2
ID complete
$G_{dv}(z)$

Fig. 4. System identification flowchart.

For a regulated SMPC, the digital PID gains are tuned using a well-recognized pole-zero matching technique [10], [31]. The PID controller is expressed as follows:

$$G_c(z) = \frac{q_0 + q_1 z^{-1} + q_2 z^{-2}}{1 - z^{-1}} \tag{26}$$

where $q_0 = 3.9$, $q_1 = -6.89$, and $q_2 = 3.1$. Based on the system in Fig. 1, the system identification sequence is described by the flowchart in Fig. 4, while the corresponding step-by-step results are illustrated in Fig. 5. Initially, the system is operating normally and is regulated by the PID compensator. When the identification process is enabled [see Fig. 5(e)], a 9-bit PRBS is injected into the feedback loop as a frequency rich excitation signal. Here, as an example, the PRBS signal is injected during the steady-state period for 20 ms [see Fig. 5(a) and (b)]. This

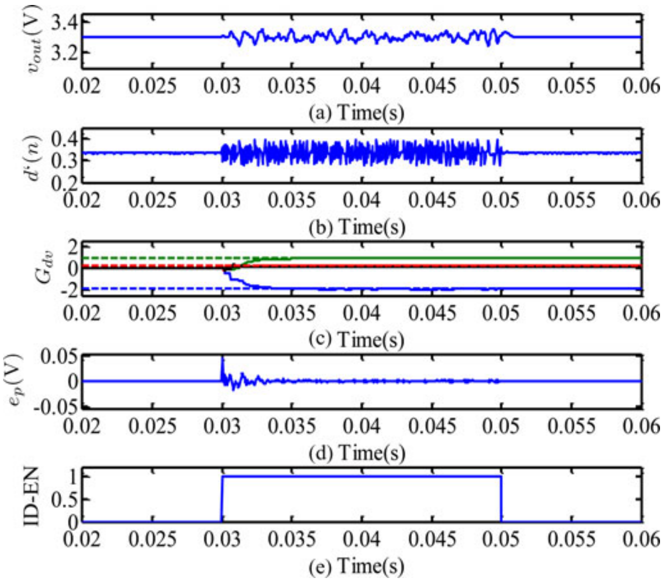

Fig. 5. Identification sequence. (a) Output voltage during ID. (b) Control signal during ID. (c) Voltage model parameters ID. (d) Voltage error prediction. (e) ID-enable signal.

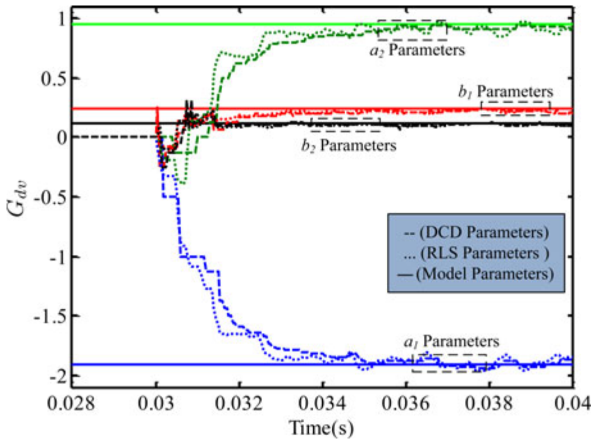

Fig. 6. Tap-weight estimation for IIR filter using DCD-RLS and classical RLS methods, compared with calculated model.

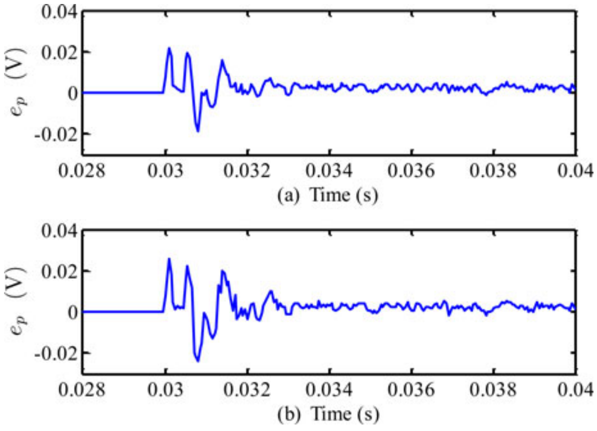

Fig. 7. Prediction error signals. (a) Ordinary RLS. (b) DCD-RLS.

TABLE V
DUTY-TO-OUTPUT PARAMETER ACCURACY COMPARISON BETWEEN DCD-RLS AND CLASSICAL RLS

	Parameters Accuracy			
	b_1	b_2	a_1	a_2
DCD-RLS	± 0.2%	± 0.7%	± 0.9%	± 1%
RLS	± 0.3%	±0.7%	±1 %	±1.1 %

is sufficient to determine the parameter convergence time. The PRBS sampling frequency f_{PRBS} is selected as 20 kHz. From this, the maximum PRBS pulse length is 511 ($L = 2^n - 1$), and the magnitude of PRBS signal is $\Delta_{\mathrm{PRBS}} = \pm$ 0.025. This is sufficiently small to cause excitation in the PWM output, but not enough to significantly compromise the normal operation of the SMPC; the output voltage ripple caused by this perturbation signal is approximately $\pm 1.5\%$ of the dc output voltage, as shown in Fig. 5(a). As each PRBS sample is injected, the DCD-RLS measures the control output signal $d'(n)$ and the power converter output voltage $v_{\mathrm{out}}(n)$. The algorithm is implemented and the IIR filter tap-weight estimation is updated. Finally, the SMPC parameters are estimated via calculation.

The effectiveness of the algorithm is verified in Fig. 5(c) and (d). The algorithm rapidly estimates the SMPC parameters $\{a_1, a_2, b_1$, and $b_2\}$ and then minimizes the error prediction signal. It is worth noting that the initial value for each parameter is assumed to be zero. This demonstrates that prior knowledge of the SMPC parameters is not essential for convergence of the algorithm. Fig. 6 shows a comparison between the DCD-RLS identification algorithm and classical RLS identification method. As depicted in Fig. 6, the DCD-RLS algorithm converges quickly (less than 10 ms) and identifies the unknown IIR filter coefficients. This in turn minimizes the prediction error signal, as shown in Fig. 7. Both techniques appear to converge to the same estimation values. The actual estimation accuracy is summarized in Table V, where it can be seen that the performance of the DCD-RLS is comparable with the conventional RLS scheme. Here, the parameters estimation accuracy has been measured at the final convergence values. It is worth noting that the DCD-RLS estimation accuracy can be further improved by increasing the number of iterations N_u, or the number update step size M, in the algorithm. However, this will increase the execution time of the algorithm and the parameter estimation convergence time will be longer. As with many systems, a compromise between complexity and accuracy must be established. The versatility of the proposed DCD-RLS scheme has been verified with a range of dc–dc discrete time models (duty-to-output voltage transfer function). In each case, the proposed method shows very promising results and can handle a wide range of uncertainty in the SMPC parameters.

VII. INITIAL EXPERIMENTAL VALIDATION

To further validate the proposed identification method, an experimental synchronous dc–dc buck converter has been designed and tested for the 5 W operation. This is used to generate real-time practical data for direct input into the DCD-RLS algorithm discussed. For easy comparison with the original simulation

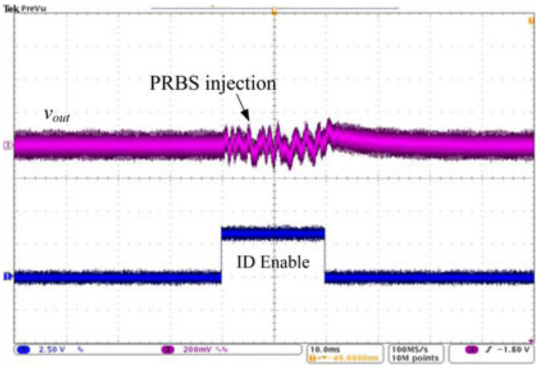

Fig. 8. Experimental output voltage waveform when identification enabled (ac coupled).

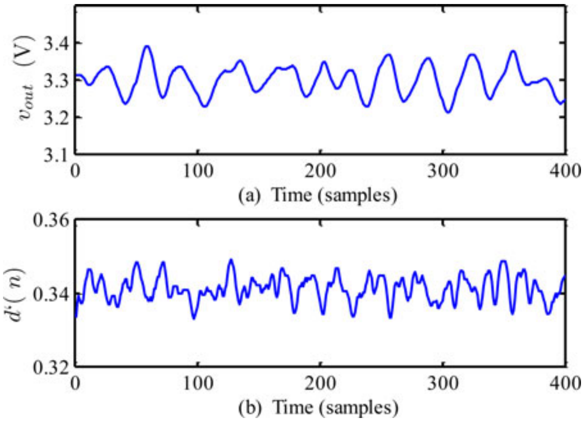

Fig. 9. Experimental output voltage and persistence excitation signal (duty signal + Δ_{PRBS}) results during ID, based on sampled data collected from DSP.

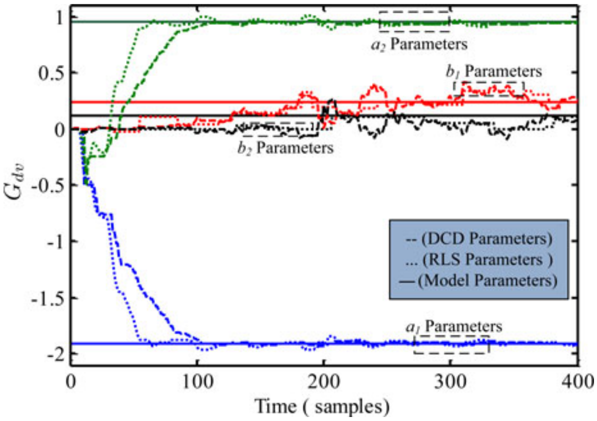

Fig. 10. Experimental tap-weight estimation for IIR filter with DCD-RLS and classical RLS methods, compared with calculated model (M-parameters).

results, similar parameters and component values to those outlined in Section VI are chosen. A Texas Instruments TMS320F28335 digital signal processor (DSP) platform is used to implement the digital controller and to inject the digital PRBS. The DSP has on-board digital PWM and 12-bit A/D converter channels. The A/D resolution is 0.7 mV and it samples the output voltage at 20 kHz, the same rate as the PWM switching frequency. A 9-bit PRBS is generated and implemented in the DSP. The PRBS amplitude, $\Delta_{\text{PRBS}} = \pm 0.008$, and the total date length is 511; therefore, a complete PRBS sequence is $L/f_s =$ 25 ms. The PID compensator, described in (26), is tuned using the aforementioned pole-zero matching method [10], [31] and set to regulate the buck converter output voltage to 3.3 V.

During the practical work, the same procedure shown in Fig. 4 is followed. Fig. 8 highlights the output voltage waveform of the experimental buck converter when the PRBS disturbance is injected to allow for system identification. Initially, the SMPC is working under normal conditions (system identification disabled). The system identification process is then enabled; the PRBS signal is injected into the loop and the system begins to estimate the unknown parameters of the buck converter model. The disturbance in the output voltage, created by the PRBS, is clearly visible in Fig. 8. The voltage ripple is approximately $\pm 3\%$ with respect to the nominal dc output voltage. However, it can also be seen that this disturbance only exists when the identification process is enabled. After 20 ms, the process is complete, and the buck converter reverts back to normal operation. The PRBS injection time is deliberately increased in this example test to fully demonstrate the convergence rate of the parameter estimation. The actual length of time of the excitation can be significantly reduced in the final optimized solution.

Now, the measurement data from the dc–dc converter are stored in the DSP memory and exported to MATLAB for post-processing after the full test sequence has been applied to the power converter. Fig. 9 shows the sampled output voltage and duty cycle data from the dc–dc converter during the identification process. From the measured data, the DCD-RLS performs the cycle-by-cycle parameter estimation algorithm previously described to identify the tap weights of the IIR filter and minimize the prediction error signal. The experimental parameters of the DCD-RLS algorithm are as follows: $N_u = 1$, $H = 1$, $M = 8$, $\lambda = 0.95$, and $\delta = 0.001$. These parameters are chosen to match the initial buck converter simulation settings and allow for easy comparison of results.

The results from experimental measurement are shown in Fig. 10. Importantly, there is excellent agreement with the original simulation results in Fig. 6. The practical-based results show both the classical RLS method and the DCD-RLS algorithm converge quickly (<10 ms) to virtually the same parameter estimation values. Furthermore, it is apparent from Fig. 11 that the voltage prediction error signal for both algorithms (RLS and DCD-RLS) converges quickly to zero. In this way, both techniques successfully identify the discrete model of the SMPC from real-time experimental data. However, as shown in earlier analysis, the computational effort of the DCD-RLS is substantially lower. It is worth noting that in both methods the convergence time of the pole coefficients (a_1, a_2) is faster and more accurate than the zero coefficients (b_1, b_2). The poles converge in about 5 ms with parameter estimation accuracy better than ± 1%. In comparison, the zeros converge in approximately 10 ms with parameter estimation accuracy of $\pm 5\%$. This is reassuring since in many control systems, including SMPCs,

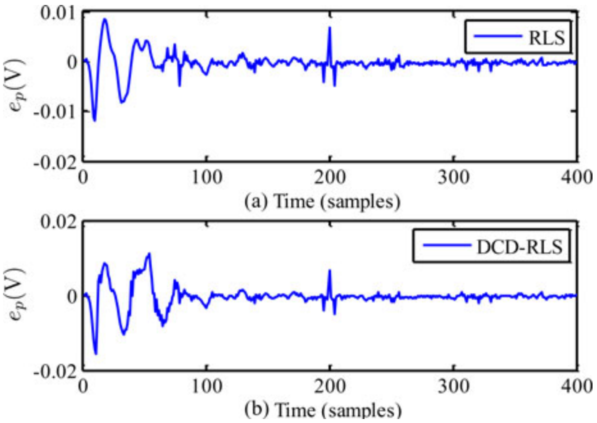

Fig. 11. Experimental error prediction results. (a) Ordinary RLS. (b) DCD-RLS.

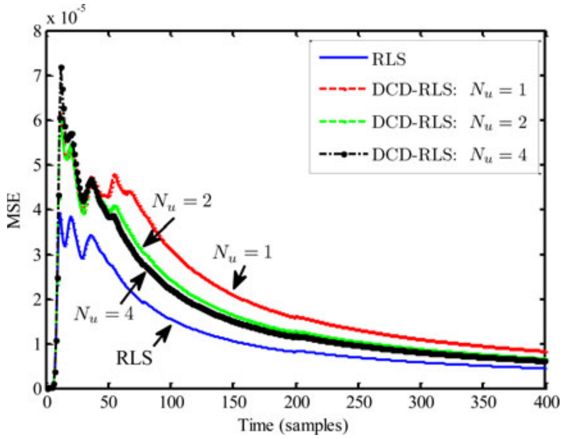

Fig. 12. Experimental learning curves comparison results of classical RLS against DCD-RLS at different iteration values.

accurate knowledge of the pole locations is important for stability analysis and controller design [6].

It has already been noted that the computation of the DCD-RLS algorithm can be reduced by decreasing the update step size M of the algorithm; however, there is a cost to pay in terms of estimation accuracy. For this reason, results are also presented where the effective resolution is reduced, $M = 4$. In this case, the computational time of the algorithm is halved. Fig. 12 compares the mean square error (MSE) performance of the DCD-RLS algorithm with different iteration values N_u against the conventional RLS technique. It can be seen that the conventional RLS convergence rate and MSE magnitude are lower than the DCD-RLS; however, the convergence rate of DCD-RLS can be improved by increasing the number of iterations, albeit at the cost of increased computational complexity. As in many applications, a compromise must be made between performance and complexity. In this particular case, $N_u = 1$ is sufficient for fast SMPC parameters estimation with acceptable estimation error.

VIII. Conclusion

With recent advances in microprocessor technology and continual improvement in pricing, significantly more advanced control solutions are now possible in many industrial and commercial systems. However, in the area of small-power electronic systems, such as SMPC applications, cost and complexity are clearly a major concern. In the area of system identification, least squares methods, like the basic RLS algorithm, provide promising results in terms of fast convergence rate, small prediction error, and accurate parametric identification. However, they often have limited application in SMPC and other low-power, low-cost applications due to computationally heavy calculations demanding significant hardware resources. For this reason, this paper has introduced a computationally efficient DCD-RLS method to overcome some of the limitations of many classic RLS algorithms. The process is based on an error equation IIR adaptive filter scheme, which is well suited for SMPC parameter estimation. The system identifies the IIR filter tap weights on a cycle-by-cycle basis by injecting a perturbed input signal and monitoring the corresponding output response. The proposed solution demonstrates that the identification method is able to work continuously in the control loop and quickly minimize the prediction error power, thus estimating the model parameters. Simulation and initial experimental results demonstrate that this approach exhibits very good identification metrics (convergence rate, parameters estimation, and prediction error) and the performance is comparable to more complex solutions such as RLS techniques. System identification is the first step to developing digital adaptive and self-tuning controller designs. The proposed method can be easily accompanied with many adaptive control solutions. Ongoing research is, therefore, set to focus on complete solutions with emphasis on hardware optimization for efficiency and low-cost implementation.

References

[1] M. Shirazi, R. Zane, and D. Maksimovic, “An auto-tuning digital controller for DC–DC power converters based on online frequency-response measurement,” *IEEE Trans. Power Electron.*, vol. 24, no. 11, pp. 2578–2588, Nov. 2009.

[2] M. M. Peretz and S. Ben-Yaakov, “Time domain identification of PWM converters for digital controllers design,” in *Proc. IEEE Power Electron. Spec. Conf.*, 2007, pp. 809–813.

[3] L. Corradini, P. Mattavelli, and D. Maksimovic, “Robust relay-feedback based autotuning for DC–DC converters,” in *Proc. IEEE Power Electron. Spec. Conf.*, 2007, pp. 2196–2202.

[4] B. Miao, R. Zane, and D. Maksimovic, “Automated digital controller design for switching converters,” in *Proc. IEEE Power Electron. Spec. Conf.*, 2005, pp. 2729–2735.

[5] N. Kong, A. Davoudi, M. Hagen, E. Oettinger, X. Ming, H. Dong Sam, and F. C. Lee, “Automated system identification of digitally-controlled multi-phase DC–DC converters,” in *Proc. IEEE Appl. Power Electron. Conf. Expo.*, 2009, pp. 259–263.

[6] M. Algreer, M. Armstrong, and D. Giaouris, “System identification of PWM DC–DC converters during abrupt load changes,” in *Proc. IEEE Ind. Electron. Conf.*, 2009, pp. 1788–1793.

[7] B. Johansson and M. Lenells, “Possibilities of obtaining small-signal models of dc–dc power converters by means of system identification,” in *Proc. Telecommun. Energy Conf.*, 2000, pp. 65–75.

[8] M. Shirazi, J. Morroni, A. Dolgov, R. Zane, and D. Maksimovic, “Integration of frequency response measurement capabilities in digital controllers for DC–DC converters,” *IEEE Trans. Power Electron.*, vol. 23, no. 5, pp. 2524–2535, Sep. 2008.

[9] B. Miao, R. Zane, and D. Maksimovic, "System identification of power converters with digital control through cross-correlation methods," *IEEE Trans. Power Electron.*, vol. 20, no. 5, pp. 1093–1099, Sep. 2005.

[10] Z. Zhenyu and A. Prodic, "Limit-cycle oscillations based auto-tuning system for digitally controlled DC–DC power supplies," *IEEE Trans. Power Electron.*, vol. 22, no. 6, pp. 2211–2222, Nov. 2007.

[11] A. Barkely and E. Santi, "Improved online identification of a DC–DC converter and its control loop gain using cross- correlation methods," *IEEE Trans. Power Electron.*, vol. 24, no. 8, pp. 2021–2031, Aug. 2009.

[12] T. Roinila, M. Vilkko, and T. Suntio, "Fast loop gain measurement of a switched- mode converter using a binary signal with a specified Fourier amplitude spectrum," *IEEE Trans. Power Electron.*, vol. 24, no. 12, pp. 2746–2755, Dec. 2009.

[13] G. E. Pitel and P. T. Krein, "Real-time system identification for load monitoring and transient handling of DC–DC supplies," in *Proc. IEEE Power Electron. Spec. Conf.*, 2008, pp. 3807–3813.

[14] L. Ljung, *System Identification: Theory for the User*, 2nd ed. Upper Saddle River, NJ: Prentice-Hall, 1999.

[15] L. Corradini, P. Mattavelli, W. Stefanutti, and S. Saggini, "Simplified model reference-based autotuning for digitally controlled SMPS," *IEEE Trans. on Power Electron.*, vol. 23, pp. 1956–1963, 2008.

[16] Y.-F. Liu, E. Meyer, and X. Liu, "Recent developments in digital control strategies for DC/DC switching power converters," *IEEE Trans. Power Electron.*, vol. 24, no. 11, pp. 2567–2577, Nov. 2009.

[17] C. C. Hu, H.Y. Lin, and J. H. Wen, "An adaptive fuzzy-logic variable forgetting factor RLS algorithm," in *Proc. IEEE Veh. Technol. Conf.*, 2005, pp. 1412–1416.

[18] M. M. Peretz and S. Ben-Yaakov, "Time-domain design of digital compensators for PWM DC–DC converters," *IEEE Trans. Power Electron.*, vol. 27, no. 1, pp. 284–293, Jan. 2011.

[19] G. F. Franklin, J. D. Powell, and M. L. Workman, *Digital Control of Dynamic System*, 3rd ed. Upper Saddle River, NJ: Prentice-Hall, 1998.

[20] J. Morroni, L. Corradini, R. Zane, and D. Maksimovic, "Robust adaptive tuning of digitally controlled switched-mode power supplies," in *Proc. IEEE Appl. Power Electron. Conf. Expo.*, 2009, pp. 240–246.

[21] Y. V. Zakharov, G. P. White, and L. Jie, "Low-complexity RLS algorithms using dichotomous coordinate descent Iterations," *IEEE Trans. Signal Process.*, vol. 56, no. 7, pp. 3150–3161, Jul. 2008.

[22] M. Algreer, M. Armstrong, and D. Giaouris, "Adaptive PD+I control of a switch mode DC–DC power converter using a recursive FIR predictor," *IEEE Trans. Ind. Appl.*, vol. 47, no. 5, pp. 2135–2144, Sep. 2011.

[23] S. Haykin, *Adaptive Filter Theory*, 4th ed. Upper Saddle River, NJ: Prentice-Hall, 2002.

[24] P. S. R. Diniz, *Adaptive Filtering Algorithms and Practical Implementation*, 2nd ed. Norwell, MA: Kluwer, 2002.

[25] P. Dobra, R. Duma, D. Moga, and M. Trusca, "Adaptive system identification and control using DSP for automotive power generation," in *Proc. 16th Mediterranean Conf. Control Autom.*, 2008, pp. 1302–1307.

[26] A. Kelly and K. Rinne, "A self-compensating adaptive digital regulator for switching converters based on linear prediction," in *Proc. IEEE Appl. Power Electron. Conf. Expo.*, 2006, pp. 712–718.

[27] Y. V. Zakharov, B. Weaver, and T. C. Tozer, "Novel signal processing technique for real-time solution of the least squares problem," in *Proc. 2nd Int. Workshop Signal Process. Wireless Commun.*, 2004, pp. 155–159.

[28] L. Jie, Y. V. Zakharov, and B. Weaver, "Architecture and FPGA design of dichotomous coordinate descent algorithms," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 56, no. 11, pp. 2425–2438, Nov. 2009.

[29] L. Guo, J. Y. Hung, and R. M. Nelms, "Evaluation of DSP-based PID and fuzzy controllers for DC–DC converters," *IEEE Trans. Ind. Electron.*, vol. 56, no. 6, pp. 2237–2248, Jun. 2009.

[30] L. Guo, "Implementation of digital PID controllers for DC–DC converters using digital signal processors," in *Proc. IEEE Int. Conf. Electro/Inf. Technol.*, 2007, pp. 306–311.

[31] V. Yousefzadeh, W. Narisi, Z. Popovic, and D. Maksimovic, "A digitally controlled DC/DC converter for an RF power amplifier," *IEEE Trans. Power Electron.*, vol. 21, no. 1, pp. 164–172, Jan. 2006.

Maher Algreer received the B.Sc. degree in electronic and communication engineering in 1999, and the M.Sc. degree in computer engineering in 2005, both from Mosul University, Mosul, Iraq, where he specialized in the field of digital self-tuning PID controllers using fuzzy logic controller. He is currently working toward the Ph.D. degree in electrical, electronic, and computer engineering at Newcastle University, Newcastle Upon Tyne, U.K.

His research interests include embedded control, adaptive digital control, fuzzy logic, signal processing, system identification, and power electronic control design.

Matthew Armstrong received the M.Eng. and Ph.D. degrees from Newcastle University, Newcastle Upon Tyne, U.K. in 1998 and 2007, respectively.

He is currently a Lecturer in control of electrical power at Newcastle University. Prior to his university lectureship, he spent eight years as a Research Associate with the Newcastle University Power Electronics, Drives and Machines Group. His current research interests include real-time digital control of power electronic converters and electrical drive systems, advanced control of grid connected renewable energy systems, and hardware-in-the-loop emulation systems.

Damian Giaouris was born in Munich, Germany, in 1976. He received the Diploma degree in automation engineering from the Technological Educational Institute of Thessaloniki, Thessaloniki, Greece, in 2000, and the M.Sc. degree in automation and control and the Ph.D. degree in the area of control and stability of induction machine drives from Newcastle University, Newcastle Upon Tyne, U.K., in 2001 and 2004, respectively. He also received the B.Sc. degree in mathematics from the Open University, Milton, Keynes, U.K., in 2010.

His research interests include advanced nonlinear control, estimation, digital signal processing methods applied to electric drives, and nonlinear phenomena in power electronic converters.