IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 3 January 2023, accepted 27 January 2023, date of publication 13 February 2023, date of current version 16 February 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3244577

RESEARCH ARTICLE

Experimental Assessment of a Tiny AI-Empowered Output Filter Parameter Extraction Framework for Digital Power

KELVIN YI-WEN HONG[1], HENRY SHU-HUNG CHUNG[1], (Fellow, IEEE), YANGXIAO XIANG[1], AND ALAN WAI-LUN LO[2]

[1]Department of Electrical Engineering, Centre for Smart Energy Conversion and Utilization Research, City University of Hong Kong, Hong Kong
[2]Department of Computer Science, Chu Hai College of Higher Education, Tsuen Wan, Hong Kong

Corresponding author: Henry Shu-Hung Chung (eeshc@cityu.edu.hk)

This work was supported in part by the Green Tech Fund of the Hong Kong Special Administration Region, China, under Project GTF202020166; and in part by the Innovation Fund Denmark, Denmark, through the Project Advanced Power Electronic Technology and Tools (APETT) under Grant 6154-00010B.

ABSTRACT An experimental assessment of a ''plug-and-play'' tiny artificial-intelligence-empowered output filter parameter extraction framework for digital power is presented. The framework can be incorporated into an existing digital controller to perform online parameter extraction without adding extra sensors or modifying the power conversion stage. The idea is based on firstly transferring the control from the default control law to the framework, which consists of a predefined control law to perform output regulation for a few switching cycles, then introducing a small-signal perturbation into the control signal, and finally utilizing a long short-term memory (LSTM) network to recognize the dynamic response of the control signal to perform either regression or classification of filter parameters. The LSTM network is trained with a reconfigurable output filter. The proposed framework has been successfully evaluated on a 240W, 100V/48V buck DC/DC converter prototype. The framework's performance is studied by extracting the parameters of a second-order output filter and a fourth-order output filter. For the second-order output filter, the root-mean-square errors (RMSEs) in performing the filter inductor and capacitor regressions are 2.35% and 2.25%, respectively, and the F1-scores in classifying the inductance and capacitance are 0.805 and 0.815, respectively. The framework occupies 0.93% of the memory space of the controller with 512kB flash memories. The extraction time is 17.3ms. For the fourth-order filter, the maximum RMSEs in performing the regression of the filter inductors and capacitors are 2.04% and 6.49%, respectively. The framework occupies 3.62% of the memory space and the extraction time is 328ms.

INDEX TERMS Artificial intelligence, digital power, system identification, neural network, control, power converters, diagnosis.

I. INTRODUCTION

The latest developments in microelectronics and embedded system technology have accelerated the penetration of *Digital Power* (DP) to meet the increasingly complex power electronic systems [1]. DP offers the merits of high flexibility, high reliability, and system miniaturization. Apart from the monolithic integration of the power conversion stage (PCS) and controller, such as proposing sophisticated digital pulse-width modulation (PWM) techniques to improve power efficiency in [2] and enhance transient response in [3], mainstream research also emphasizes dealing with the effects of sampling and computation delays and nonlinearities [2], [3], enhancing the characteristics of individual functional block and control law [4], [5], and optimizing system performance [6], [7]. Reference [2] proposes a digital PWM and low-power self-tracking zero current detector that can achieve high linearity and low power consumption. Reference [3]

The associate editor coordinating the review of this manuscript and approving it for publication was Ali Raza.

proposes a digitally controlled buck converter that utilizes multiple-outputs bang-bang phase detector in a digital phase locked loop to reduce design complexity and improve transient response. Reference [4] proposes an FPGA-based hysteretic multimode control scheme with two PWM modes and pulse frequency modulation (PFM) modes for mixed conduction mode is proposed. Reference [5] proposes a digital predictive control algorithm for boost-type power factor corrector with the peak-to-peak inductor current estimated by a triangular modulation-based predictive control and a known value of output inductor. In [6], a digital average voltage and digital average current predictive control achieves a high output voltage accuracy, wide stability range, and fast transient response. In [7], the effective resolution of the PWM is enhanced by using a dithering technique on an "SoC FPGA" that can reduce peak output voltage ripple and lower switching frequency subharmonics. The control laws are typically designed by taking circuit parameters into account [5], [6]. For example, in [5], the peak-to-peak inductor is estimated by taking the value of inductor into account. As discussed in [6], the control scheme proposed is robust and can keep converter operate in stable when the circuit parameters have small variation.

In principle, if the filter parameters can be identified, the digital controller can adaptively tune the parameters in the control law to maintain optimal system performance [8], [9]. For examples, the optimal sliding surface function in the sliding mode control for dynamic voltage restorers in [8] is dependent on the output filter parameters. The switching functions in the trajectory prediction control proposed in [9] requires using output filter parameters to aim at reaching the steady state in one switching cycle after supply/load disturbances.

The identified filter parameters can also be used as information for monitoring the condition of the filter components, thereby preventing possible catastrophic faults [10], [11], [12], [13] and/or parametric faults [14], [15], [16], and thus reducing the failure rate. The entire system will not break down unexpectedly and create unplanned downtime. In [11], different fault conditions are identified by using a Rogowski coil to sense inductor voltage and a voltage sensor to sensor diode voltage. The determination of the inductor voltage is obtained by considering the inductance of the output inductor and the coupling inductance. A detection method is proposed in [12] to classify transient fault current and switching inrush current. An overview of different condition monitoring techniques for DC-link capacitors is given in [13], because the capacitor is known to be a reliability critical component [14]. A degradation model of capacitors is presented in [15]. In [16], a reference submodule-based capacitor monitoring method for capacitance estimation in the modular multilevel converter is proposed. The idea is based on utilizing the relationships between one capacitor and the others to estimate the capacitance ratios.

The converter output filter is composed of passive components, including inductors, capacitors, and resistors. Among them, capacitors have the shortest life expectancy and are susceptible to parametric faults. A large body of literature is devoted to extracting the intrinsic parameters of output capacitor, including capacitance and equivalent series resistance (ESR) [13], [16]. The ESR is sometimes chosen as an indicator to indirectly determine the health condition of the capacitor as degraded capacitors have their capacitance reduced and ESR increased.

In theory, the value of each filter component can be computed by processing the sensed voltage and current waveforms associated with the component under test [17], [18], [19]. In addition to requiring extra sensors and increasing cost, the reliability of the entire system is also negatively affected. In [17], the idea is based on firstly introducing a low-frequency signal to the control signal, and then calculating the capacitor value by sampling the low-frequency voltage and current associated with the capacitor. In [18], odd harmonic frequencies are injected by the inverter during the night time. The capacitor impedances at different frequencies are obtained using a tuned bandpass filter. The ESR and capacitance are estimated by using Least Mean Square algorithm. In [19], the ESR is calculated by using high-frequency harmonics in capacitor voltage and current and the capacitance is calculated by using low-high frequency harmonics. The harmonics of interest are extracted by the Goertzel algorithm.

Practically, the operation bandwidth of sensors limits the measurement of high-frequency voltage and current waveforms, such as ripple voltage and ripple current. For example, reference [20] requires sensing the inductor current peak and output voltage jump. This is particularly critical for future power electronic systems that will operate at a high switching frequency.

Another extraction approach is based on first devising a circuit model to describe the system dynamics, and then applying deterministic or stochastic techniques to determine filter parameters [21], [22], [23]. Since the accuracy depends on the level of abstraction and sophistication of the model, characterization of high-order filters and nonlinear behavior will pose challenges in mathematical modeling and guaranteeing the numerical convergence.

To cope with the above issues, this paper gives another perspective by proposing a "plug-and-play" tiny artificial intelligence (AI)-empowered output filter parameter extraction framework for digital power. The idea is based on extracting the features of the dynamics of the duty cycle with a long short-term memory (LSTM) network to perform either 1) classification or 2) regression of the output filter parameters, after the system is subject to a small-signal perturbation in the control signal. The proposed framework has the following features, and a comparison with other methods has been given in Table 1:

TABLE 1. Comparison of existing related methods.

	Description	Properties
Signal-processing based methods	[17] Calculate the capacitor value by sampling low-frequency voltage and current associated with the capacitor	***Pros***: Precision, Universality. ***Cons***: Sensor-consuming, high bandwidth sensor required.
	[18] Calculate capacitor impedances at different frequencies through processing v_i, i_L. The ESR and capacitance are then estimated by using Least Mean Square algorithm.	
	[19] The ESR and the capacitance are calculated by using high-frequency and low-high frequency harmonics in capacitor voltage and current, separately.	
Model-based methods	[21] Use Biogeography-based optimization to estimate the system parameters based on the state-space model	***Pros***: Precision, easy to implement. ***Cons***: Sensor-consuming, characterization of high-order filters and nonlinear behavior posing challenges
	[22] Identify the fault by establishing system fault models	
Data-driven Based	Proposed method	*Pros*: Precision, no additional sensor, model-free, able to incorporate into existing digital controllers *Cons*: Require data generation and NN training

1. No additional voltage and current sensors are needed. Many online parameter identification methods require multiple sensors. Besides increasing system complexity and cost, additional sensors have a detrimental effect on system reliability.

2. The framework is a standalone software module. It can be incorporated into existing digital controllers to perform online extraction of output filter parameters.

3. No detailed mathematical models are needed, as the framework is purely data-driven. Conversely, many parameter identification methods require sophisticated system modeling.

4. The weights and parameters in the LSTM network are trained under all operational conditions. They are unnecessary to be trained for individual converters.

Supervised learning for determining the parameters of the LSTM network is adopted. The training and testing datasets are obtained under pre-determined output filter parameters, realized by a reconfigurable output filter (ROF). The parameters of the LSTM network are trained by a stochastic gradient descent algorithm. Data pre-processing, network operation, post-processing of the network output, and network training are given.

The proposed framework has been experimentally evaluated on a 240W, 100V/48V buck DC/DC converter with its output filter configured as either a second-order *LC* or fourth-order *LCLC* filter. The perturbation causes the output voltage with ±3.3% variation only.

II. PROPOSED FRAMEWORK

Fig. 1 shows the system architecture of a buck DC/DC converter with an output filter. The output filter is either a second-order *LC* or fourth-order *LCLC* filter. The detailed configuration of the output filter is given in Sec. IV-A. The output voltage v_o is regulated at the reference value V_{ref} by a "default controller". The error voltage is $v_e = V_{ref} - v_o$. For the sake of illustration, the default controller executes typical proportional-plus-integral (PI) control with the proportional gain of $K_{p,d}$ and integral gain of $K_{i,d}$. It generates a control signal to dictate the state of the main switch S_1. The duty cycle of the control signal is d_d, which is fed to the driving circuit via a logical selector S_{sel}.

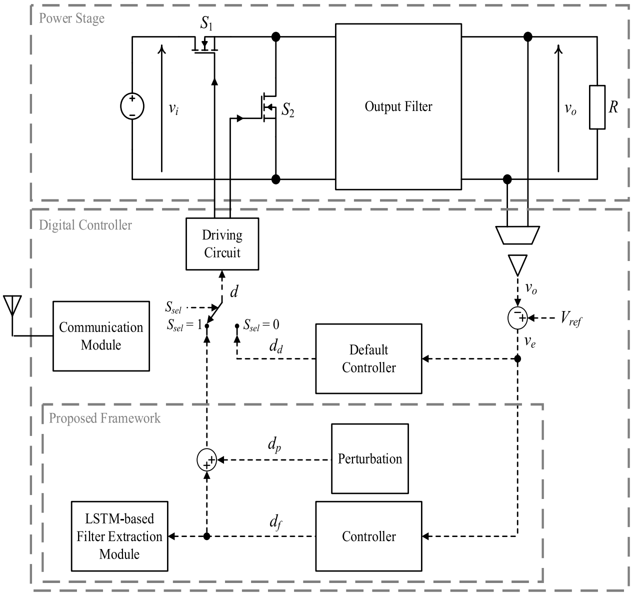


FIGURE 1. System architecture.

The proposed framework performs the same objective as the default controller to regulate the output voltage. Upon estimating the filter parameters, the controller will transfer the output voltage regulation function from the default controller to the framework temporarily via S_{sel}. The framework "Controller" in Fig. 1 implements a PI control with the proportional gain $K_{p,f}$ and integral gain $K_{i,f}$. It generates a control signal with the duty cycle d_f in the n-th switching cycle. Mathematically,

$$\begin{aligned} d_f[n] &= G_f(v_e[n]) \\ &= K_{p,f}\, v_e[n] + K_{i,f} \sum_{i=1}^{n} v_e[i]\, T_{\text{int}} \end{aligned} \tag{1}$$

and

$$v_e[n] = V_{ref} - v_o[n] \tag{2}$$

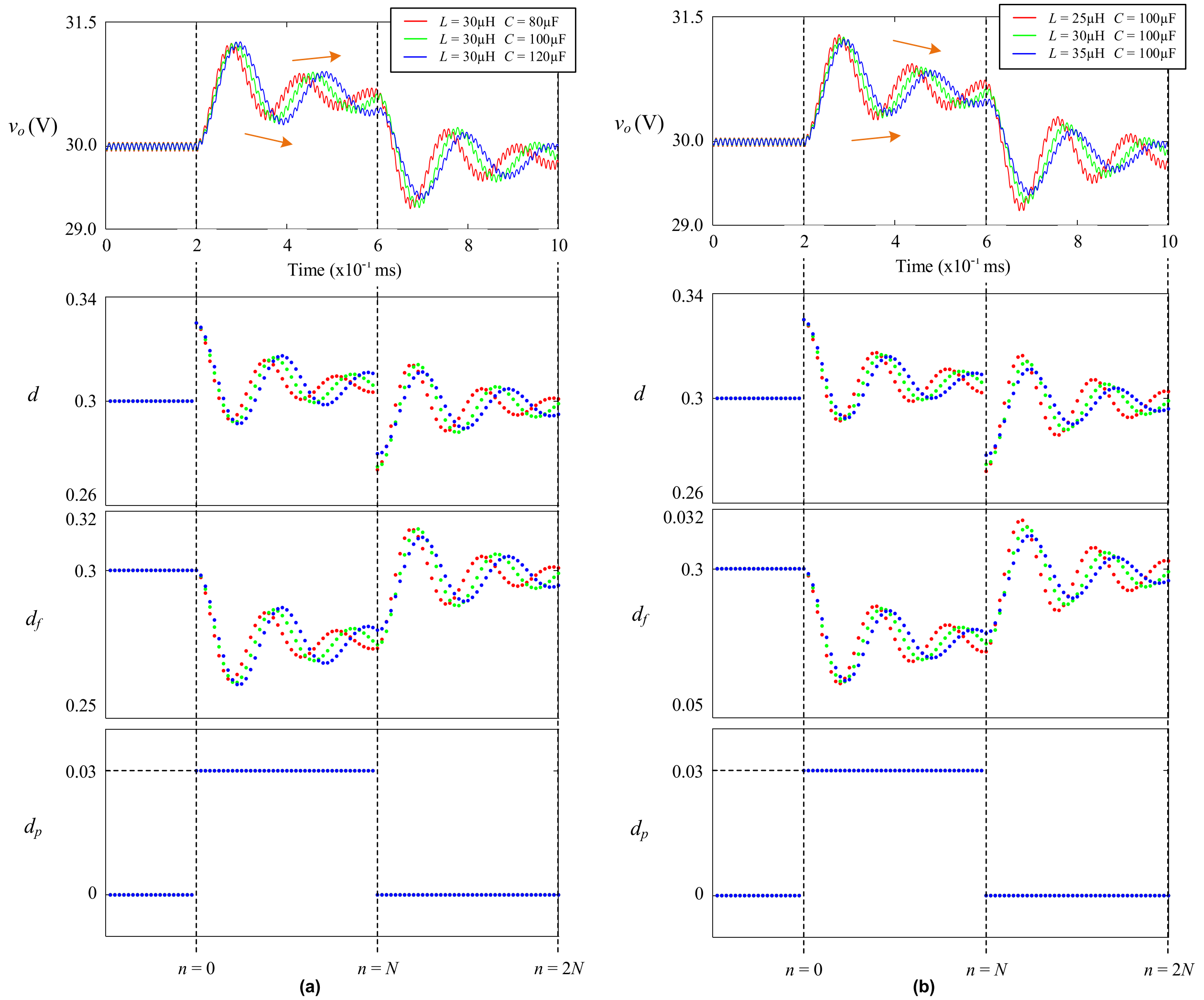


FIGURE 2. Waveforms of v_o, d, d_f, and d_p. (a) Fixed L and variable C. (b) Fixed C and variable L.

TABLE 2. Parameters used in the analysis.

Component	Value	Component	Value
L_{norm}	30 μH	$K_{p,f}$	0.03
C_{norm}	100 μF	$K_{i,f}$	1
A_p	0.03	T_{int}	10 μs
$V_{o,\max}$	31.28 V	V_{ref}	30 V
$V_{o,\min}$	29.17 V	Switching frequency	100 kHz

where T_{int} is the sampling time interval generated by the timer interrupt and is the same as the switching period of the main switch.

The duty cycle of S_1, d, is superimposed by a perturbed signal d_p, which is defined as

$$d_p[n] = \begin{cases} A_p & 0 \le n \le N-1 \\ 0 & N \le n \le 2N-1 \end{cases} \tag{3}$$

where A_p is the magnitude of the perturbation.

The duration of the perturbation lasts $2NT_{\text{int}}$. After engaging the framework, v_o starts increasing at $n = 0$, where d_p is changed from zero to A_p. Conversely, v_o starts reducing at $n = N$, where d_p is changed from A_p to zero. The magnitude of A_p is chosen such that the peak output voltage overshoot $V_{o,\max}$ and undershoot $V_{o,\min}$ should be less than that defined in the specification of the converter.

Therefore, when $S_{sel} = 1$,

$$d[n] = d_f[n] + d_p[n] \tag{4}$$

Fig. 2 shows the waveforms of d, d_f, d_p, and v_o after the control is switched between the default controller and the framework with different combinations of the values of L and C in a second-order output filter (Fig. 1). The nominal values of L and C are L_{norm} and C_{norm}, respectively. The parameters used in the analysis are tabulated in Table 2. It can be observed that the values of L and C determine the amplitude and frequency of oscillation of d and v_o. Thus, the waveform of d can be taken as a *unique signature* to identify the values

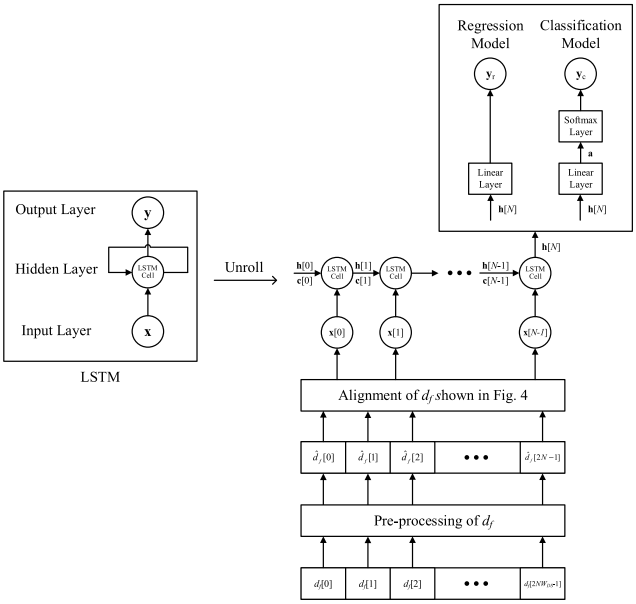


FIGURE 3. LSTM network for filter parameter extraction.

of L and C without using extra sensors to measure the voltage and current waveforms associated with the components. Moreover, the transfer function G_f in (1) exhibits a low-pass filtering function that can attenuate high-frequency noise in v_o. To make the *signature* recognize easily, the values of $K_{p,f}$ and $K_{i,f}$ are chosen, such that the entire converter is operated in a lightly damped condition, i.e., with damping factor close to zero. It is worth noting that the stability of the proposed framework can be confirmed as long as $K_{p,f}$ and $K_{i,f}$ are designed according to the phase margin under all possible combinations of inductance and capacitance [24], [25], [26].

III. PARAMETER IDENTIFICATION USING LSTM

Two LSTM networks having the output using regression layer and classification layer, respectively, are described. The regression model is used to estimate the value of each filter component. It is suitable for providing the default controller with information to adjust control parameters. The classification model is used to classify the value of each filter parameter to a predefined bin. The number of bins and bin size are user-defined. It is suitable for one to estimate the condition of the output filter.

Fig. 3 shows the LSTM network utilizing a sequence-to-one model [27]. It consists of an input layer, a hidden layer, and an output layer. The hidden layer contains LSTM cell. For illustration, the network is unrolled through time to process input data $\mathbf{x}$ at every time step. The hidden state $\mathbf{h}[n]$ and the cell state $\mathbf{c}[n]$ are updated by the LSTM cell and then passed to the next time step. At the end of the calculation, i.e., $n = N$, the network gives the output $\mathbf{y}$. The elements of $\mathbf{y_r}$ are estimated values for the filter parameters in the regression model while the elements of $\mathbf{y_c}$ are probabilities of each bin in the classification model.

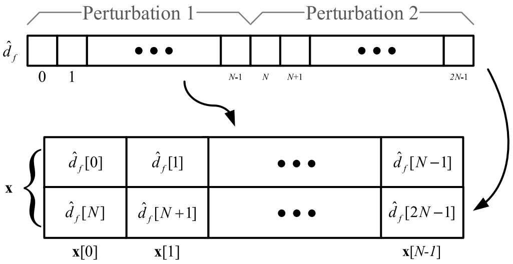


FIGURE 4. Formulation of the input layer.

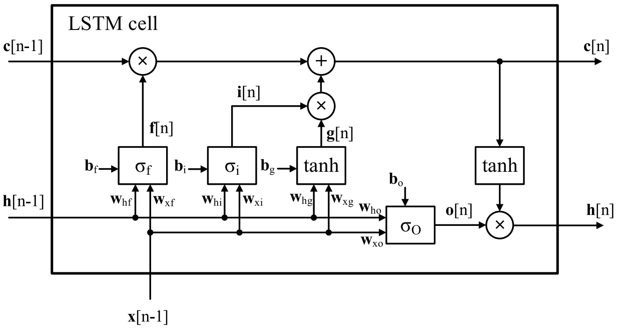


FIGURE 5. Structure of LSTM cell.

A. PRE-PROCESSING OF SAMPLED DUTY CYCLE

The sampled duty cycle is filtered, down-sampled, normalized to the steady-state characteristics and performed differential from the time series of the best-fit averaged duty cycle response in the training process. The duty cycle d_f is firstly filtered by a moving average filter. The time series of the averaged duty cycle $\tilde{d}_f$

$$\tilde{d}_f[n] = \frac{1}{W_{df}} \sum_{i=n}^{n+W_{df}-1} d_f[i]s \tag{5}$$

where W_{df} is the window size of the moving average filter.

Since the natural frequency of the output filter is significantly less than the switching frequency, the input to the LSTM network is down-sampled. The averaged duty cycle is taken for the network in every W_{DS} samples. The time series of the down-sampled duty cycle $\tilde{d}_{f,DS}$ is

$$\tilde{d}_{f,DS}[n] = \tilde{d}_f[n\, W_{DS}] \tag{6}$$

for $0 \le n \le 2N - 1$.

Thus, the data length of the time series of d_f and $\tilde{d}_f$ equals $2N\, W_{DS}$. $\tilde{d}_{f,DS}$ is then normalized by multiplying the ratio between the nominal steady-state duty cycle D_{norm} in the training process and the actual steady-state duty cycle D_{act} in the operation. The normalized duty cycle $\tilde{d}_{f,DS,norm}$ is

$$\tilde{d}_{f,DS,norm}[n] = \frac{D_{norm}}{D_{act}} \tilde{d}_{f,DS}[n] \tag{7}$$

Its difference, denoted by $\hat{d}_f$, from the time series of the averaged duty cycle $\bar{d}_{train}$, which is obtained from the a-th

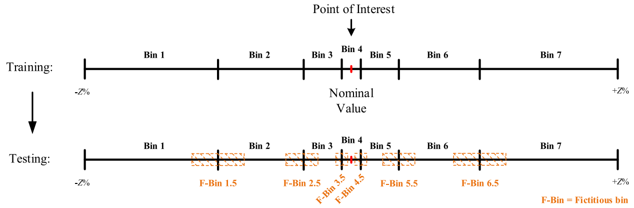


FIGURE 6. Example of the bin assignment.

dataset in the training process, is calculated by

$$\hat{d}_f[n] = \tilde{d}_{f,DS,norm}[n] - \bar{d}_{train}[n]\,|_a \tag{8}$$

where $\bar{d}_{train}[n]\,|_a = \frac{1}{E}\sum_{i=1}^{E}\tilde{d}_{f,DS,norm,i}[n]\,|_a$, E is the number of the time series of the duty cycle in the a-th dataset in the training process, and $\tilde{d}_{f,DS,norm,i}[n]\,|_a$ is the n-th sample of the i-th time series in the a-th dataset.

The a-th dataset is chosen to be the one that has minimum Euclidean distance between $\tilde{d}_{f,DS,norm}$ and $\bar{d}_{train}$ among ρ datasets in the training process. Mathematically,

$$a = \arg\min_{i}\Big(\sum_{n=0}^{2N-1}(\tilde{d}_{f,DS,norm}[n] - \bar{d}_{train}[n]\,|_i)^2\Big), \qquad \forall i \in \{1,\ldots,\rho\} \tag{9}$$

B. ALIGNMENT OF DATA FOR THE INPUT LAYER OF THE LSTM

The input layer records the time series of $\hat{d}_f$ at every step. However, to assist the LSTM to extract the features from the input data effectively, the input data $\mathbf{x}$ is composed of aligned time series of $\hat{d}_f$ under different perturbations. As shown in Fig. 4, each perturbation section of $\hat{d}_f$, as defined in (3), is considered as an individual feature. Based on Fig. 4, there are perturbations, including "Perturbation 1" with $d_p = A_p$ and "Perturbation 2" with $d_p = 0$. The input data $\mathbf{x}$ is constructed as a matrix with each row corresponding to one feature. Mathematically,

$$\mathbf{x}[k] = \begin{bmatrix} \hat{d}_f[k] \\ \hat{d}_f[k+N] \end{bmatrix}, \quad \forall k \in [0, N-1] \tag{10}$$

C. OPERATION OF THE LSTM CELL

The structure of the LSTM cell is shown in Fig. 5 [28]. It contains two memory contents $\mathbf{c}[n]$ and $\mathbf{h}[n]$, and three gating operations, including input gate σ_i, forget gate σ_f and output gate σ_o, to control the flow of information. $\mathbf{c}[n-1]$ contains the information aggregated from historical data and operations. $\mathbf{h}[n]$, which contains short-term historical information, is passed to the next step for cell operation. $\mathbf{h}[N]$ acts as the output of LSTM cell at the last step. σ_i governs the proportion of $\mathbf{x}[n]$ that enters the network cell. σ_f determines whether $\mathbf{c}[n]$ should be reset. σ_{O} determines the magnitude of the scaled $\mathbf{c}[n]$ in calculating $\mathbf{h}[n]$. σ_i, σ_f, and σ_o are all dependent on the historical information and current input. The computations within the LSTM cell include

$$\mathbf{i}[n] = \sigma_i(\mathbf{w}_{xi}\mathbf{x}[n-1] + \mathbf{w}_{hi}\mathbf{h}[n-1] + \mathbf{b}_i) \tag{11}$$

$$\mathbf{f}[n] = \sigma_f(\mathbf{w}_{xf}\mathbf{x}[n-1] + \mathbf{w}_{hf}\mathbf{h}[n-1] + \mathbf{b}_f) \tag{12}$$

$$\mathbf{o}[n] = \sigma_o(\mathbf{w}_{xo}\mathbf{x}[n-1] + \mathbf{w}_{ho}\mathbf{h}[n-1] + \mathbf{b}_o) \tag{13}$$

$$\mathbf{g}[n] = \tanh(\mathbf{w}_{xg}\mathbf{x}[n-1] + \mathbf{w}_{hg}\mathbf{h}[n-1] + \mathbf{b}_g) \tag{14}$$

$$\mathbf{c}[n] = \mathbf{f}[n] \odot \mathbf{c}[n-1] + \mathbf{i}[n] \odot \mathbf{g}[n] \tag{15}$$

$$\mathbf{h}[n] = \mathbf{o}[n] \odot \tanh(\mathbf{c}[n]) \tag{16}$$

where

$$\sigma_i = \sigma_f = \sigma_o = \sigma(k) = \frac{1}{1+e^{-k}} \tag{17}$$

$$\tanh(k) = \frac{e^{2k}-1}{e^{2k}+1} \tag{18}$$

and $\mathbf{w}_*$ and $\mathbf{b}_*$ represent the network weight and bias, respectively. For example, $\mathbf{w}_{xi}$ is the weight of $\mathbf{x}$ for σ_i, $\mathbf{w}_{hi}$ is the weight of $\mathbf{h}$ for σ_i, and $\mathbf{b}_i$ is the bias of σ_i. Element-wise sigmoid functions σ_* is used as an activation function of each gate, hyperbolic tangent tanh is used as an input activation and cell state activation and $\odot$ is an element-wise product. Training of $\mathbf{w}_*$ and $\mathbf{b}_*$ is described in Sec. IV.

D. OUTPUT LAYER

After obtaining the output of the LSTM cell at the last step, i.e., $\mathbf{h}[N]$, the output of the framework is determined by two possible output layers. They are regression layer and classification layer. Their corresponding models are described as follows.

1) USING REGRESSION LAYER

The output of the regression model, $\mathbf{y}_r$, is calculated by

$$\mathbf{y}_r = \mathbf{w}_r\mathbf{h}[N] + \mathbf{b}_r \tag{19}$$

where $\mathbf{w}_r$ and $\mathbf{b}_r$ are the weight and bias of the output layer of the regression model, respectively.

The dimension of $\mathbf{y}_r$ equals the number of filter parameters to be identified. For example, for a second-order LC filter, the dimension of $\mathbf{y}_r$ is two with one of its elements representing the value of L and another element representing the value of C.

2) USING CLASSIFICATION LAYER

For classification model, the dimension of the output vector $\mathbf{a}$ is firstly transformed by the following equation,

$$\mathbf{a} = \mathbf{w}_c\mathbf{h}[N] + \mathbf{b}_c \tag{20}$$

where $\mathbf{w}_c$ and $\mathbf{b}_c$ are the weight and bias of the classification layer, respectively. The dimension of $\mathbf{a}$ equals the number of bins in the classification model. Softmax function is used to calculate the probability of the estimated parameter in each bin. The probability of the i-th bin $\mathbf{y}_{c,i}$ is calculated by

$$\mathbf{y}_{c,i} = \frac{e^{\mathbf{a}_i}}{\sum_{j=1}^{Q} e^{\mathbf{a}_j}} \tag{21}$$

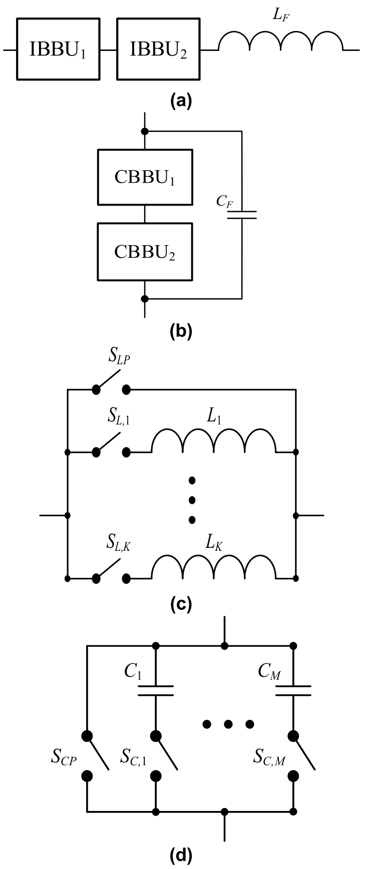


FIGURE 7. ROF. (a) IB. (b) CB. (c) IBBU. (d) CBBU.

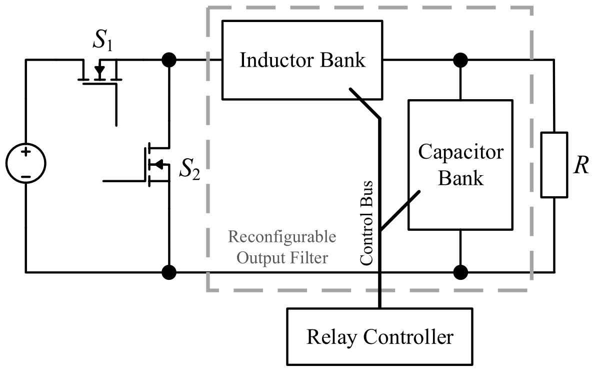


FIGURE 8. Connection of the ROF to a buck converter with second-order output filter.

where a_i is the i-th element of $\mathbf{a}$ and Q is the total number of bins.

The bin with the highest probability is the one that the considered filter parameter is classified.

Fig. 6 shows the allowable variation of a filter parameter, ranging from -Z % to +Z %. The range is divided into seven sub-ranges. Each sub-range is called "bin". Depending on the classification requirement, the bin size can be evenly or unevenly distributed. Fig. 6 shows an example that the bin size around the nominal value is smaller than the one away from the nominal value. Thus, the system will recognize slight change or degradation of the filter parameters at the early stage. It is considered as the point of interest (POI).

If the value of a filter parameter is close to the boundary of two adjacent bins, the estimated probabilities falling into the two bins are similar and thus the classification becomes uncertain. A post-processing technique is used. Q-1 fictitious bins, namely F-Bins, are introduced to represent filter parameter values around bin boundaries. The considered filter parameter will be classified into bin b_{new} with the following criteria (22), as shown at the bottom of the page, where b_{orig} is the original bin predicted without post-processing and α is the threshold to perform the post-processing.

For example, if $Z\ \% = 25\%$, $Q = 7$, $\alpha = 0.2$, the seven originally defined bins are $[-25\%, -13\%)$, $[-13\%, -5\%)$, $[-5\%, -1\%)$, $[-1\%, 1\%)$, $[1\%, 5\%)$, $[5\%, 13\%)$, and $[13\%, 25\%]$, and the six fictitious bins are $[-15.6\%, -10.4\%]$, $[-6\%, -4\%]$, $[-1.2\%, -0.8\%]$, $[0.8\%, 1.2\%]$, $[4\%, 6\%]$ and $[10.4\%, 15.6\%]$.

The performance of the post-processing technique is evaluated by comparing the predicted bin b_{new} (using eq. (22)) with the actual bin and/or actual F-Bin. Table 3 shows the evaluation criteria. The principle is that the evaluation is correct if the calculated bin number, no matter originally defined bin or fictitious one, correctly matches with the corresponding bin type.

E. ROBUSTNESS AGAINST SAMPLING NOISE

After Considering that sampling noise might influence the estimation performance of the LSTM network. As discussed in [29] and [30], input-to-state stability (ISS) can be used to verify the robustness of the trained LSTM network against sampling noise. With the LSTM network defined in (11)-(18), the ISS of a trained LSTM network can be assured if the following inequalities are met,

$$(1+\sigma(|\,\mathbf{w}_{xo}\ \mathbf{w}_{ho}\ \mathbf{b}_o\,|_\infty))\sigma(|\,\mathbf{w}_{xf}\ \mathbf{w}_{hf}\ \mathbf{b}_f\,|_\infty) < 1 \tag{23}$$

$$(1+\sigma(|\,\mathbf{w}_{xo}\ \mathbf{w}_{ho}\ \mathbf{b}_o\,|_\infty))\sigma(|\,\mathbf{w}_{xi}\ \mathbf{w}_{hi}\ \mathbf{b}_i\,|_\infty)|\mathbf{w}_{hg}|_1 < 1 \tag{24}$$

$$b_{new} = \begin{cases} \dfrac{2b_{orig}-1}{2}, & \text{if } b_{orig}-1 \geq 1 \text{ and } y_{c,b_{orig}-1} > y_{c,b_{orig}+1} \text{ and } \left|y_{c,b_{orig}-1} - y_{c,b_{orig}}\right| < \alpha \\ \dfrac{2b_{orig}+1}{2}, & \text{if } b_{orig}+1 \leq Q \text{ and } y_{c,b_{orig}-1} < y_{c,b_{orig}+1} \text{ and } \left|y_{c,b_{orig}+1} - y_{c,b_{orig}}\right| < \alpha \\ b_{orig}, & \text{otherwise} \end{cases} \tag{22}$$

where σ denotes the sigmoid activator defined in (17) and $|\cdot|_n$ denotes the induced n-norm of the matrix.

IV. NETWORK TRAINING

The LSTM network is trained by using a ROF that can be programmed to give different combinations of the values of the output inductance and output capacitance. The parameters in the LSTM network are trained by a stochastic-gradient-descent-based algorithm [31].

A. ROF

As shown in Fig. 7, the ROF is constructed by an inductor bank (IB) and a capacitor bank (CB). The IB consists of several series-connected basic units, namely (IBBUs), with a series inductor L_F. The CB consists of two series-connected capacitor bank basic units (CBBUs) and a fixed capacitor C_F connected across them. The numbers of IBBU and CBBU used in the IB and CB, respectively, depends on the required resolution of the combinations.

TABLE 3. Evaluation of the post-processing method.

	Filter parameter lies in		b_{new}		Evaluation result*	
Case	Bin	F-Bin	= Bin?	= F-Bin?	Bin	F-Bin
1	Yes	No	Yes	-	Correct	-
2			No	-	Incorrect	-
3			-	No	-	Incorrect
4		Yes	Yes	-	Correct	-
5			-	Yes	-	Correct
6			No	-	Incorrect	-
7			-	No	-	Incorrect

*Note:
"Correct" – Correct prediction
"Incorrect" – Incorrect prediction
"-" – Not applicable

Each IBBU is constructed by K (or $K + 1$) relays, $S_{L,1}$, $S_{L,2}$, …, $S_{L,K}$, and/or S_{LP} and K fixed-value inductors, L_1, L_2, …, L_K. A relay is connected in series with each inductor, forming a branch between the two terminals of the IBBU. S_{LP} is connected between the two terminals of the IBBU.

Each CBBU is constructed by M (or $M+ 1$) relays, $S_{C,1}$, $S_{C,2}$, …, $S_{C,M}$, and/or S_{CP} and M fixed-value capacitors, C_1, C_2, …, C_M. A relay is connected in series with capacitor, forming a branch between the two terminals of the CBBU. S_{CP} is connected between the two terminals of the CBBU.

With multiple IBBUs and CBBUs connected in series, the number of combinations of equivalent output inductance and capacitance can be increased. An output filter with different frequency characteristics can then be configured. Fig. 8 illustrates how the IB and CB are connected to a buck converter. The state of the relays is controlled by a low-speed microcontroller because the operating frequency of the relays is very low.

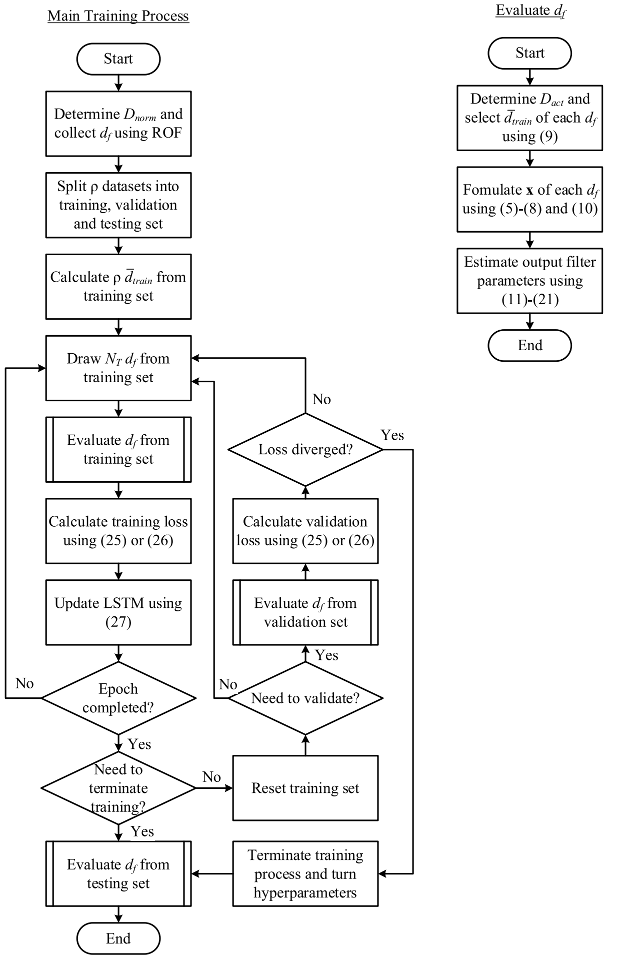

Main Training Process
Start
Determine D_{norm} and collect d_f using ROF
Split ρ datasets into training, validation and testing set
Calculate ρ $\bar{d}_{train}$ from training set
Draw N_T d_f from training set
Evaluate d_f from training set
Calculate training loss using (25) or (26)
Update LSTM using (27)
Epoch completed?
No
Yes
Need to terminate training?
No
Yes
Evaluate d_f from testing set
End
Loss diverged?
No
Yes
Calculate validation loss using (25) or (26)
Evaluate d_f from validation set
Yes
Need to validate?
No
Reset training set
Terminate training process and turn hyperparameters
Evaluate d_f
Start
Determine D_{act} and select $\bar{d}_{train}$ of each d_f using (9)
Fomulate **x** of each d_f using (5)-(8) and (10)
Estimate output filter parameters using (11)-(21)
End

FIGURE 9. Training procedure of the LSTM network.

B. TRAINING OF THE LSTM NETWORK

The time responses of the duty cycle upon subjecting the same perturbation method, as depicted in Sec. III, under different combinations of the output filter parameters with the ROF and the output powers are conducted to train, validate, and test the LSTM network parameters in Fig. 5. Let N_{LC} be the number of combinations of the output filter parameters and ρ be the number of output power levels in the entire training process. The time responses are divided into ρ datasets. Each dataset represents the system response at one power level. It contains N_{LC} time series of the duty-cycle response d_f corresponding to N_{LC} output filter parameter combinations. In other words, there are $N_{LC} \times \rho$ time series. They are randomly classified into three sets. 70% of the time series is used to train the LSTM network, 20% of the time series is used to validate the hyperparameters, and 10% of the time series is used to test the LSTM network.

Let N_T be the number of time series chosen for the training in one batch. At the start of the training process, the network parameters are randomly chosen. Then, they are determined through a stochastic-gradient-decent-based algorithm. The procedure is shown in Fig. 9. By using the ROF, D_{norm} is determined and d_f are collected and divided into ρ datasets. For each d_f, $\tilde{d}_{f,DS,norm}$ are calculated by (5)–(8). Then, $\rho\bar{d}_{train}$ are obtained. No matter in the training, validation or testing, each d_f is being evaluated with the following procedure. First, D_{act} is taken from the steady-state duty cycle before the framework is engaged. $\bar{d}_{train}$ is chosen among ρ datasets, where the a-th dataset is selected by using (9). Then, the network input $\mathbf{x}$ is formulated with (5) – (8) and (10), as described in Sec. III-A and III-B. Finally, the output filter parameters are estimated with the LSTM using (11)-(22).

During the training process, the estimated output filter parameters are compared to the actual filter parameters. The difference is called "loss". For the regression model, the loss ℓ of the N_T estimated output filter parameters is defined as

$$\ell = \frac{1}{2}\sum_{i=1}^{N_T}\sum_{j=1}^{R}(\mathbf{y}_{r,e,i,j} - \mathbf{y}_{r,t,i,j})^2 \tag{25}$$

where R is the number of components in the output filter, and $y_{r,e,i,j}$ and $y_{r,t,i,j}$ are the estimated and actual value of the j-th component value associated with the i-th time series, respectively. $y_{r,e,i,j}$ is obtained by (19).

TABLE 4. Nominal, Minimum and Maximum Values of the Output Filter Components.

Second-order filter				Fourth-order filter			
	Nom	Min	Max		Nom	Min	Max
$L_{2,1}$ (μH)	470	352.5	587.5	$L_{4,1}$ (μH)	68	51	85
$C_{2,1}$ (μF)	330	247.5	412.5	$C_{4,1}$ (μF)	22	16.5	27.5
				$L_{4,2}$ (μH)	68	51	85
				$C_{4,2}$ (μF)	47	35.2	58.8

TABLE 5. Network and training parameters.

Parameter	Value
Batch size	64
Epoch	100
Learning rate	0.001

For the classification model, the loss ℓ is defined as

$$\ell = -\frac{1}{N_T}\sum_{i=1}^{N_T}\sum_{j=1}^{Q} y_{c,t,i,j}\log y_{c,e,i,j} \tag{26}$$

where $y_{c,e,i,j}$ is the estimated probability of the j-th bin of the i-th time series, and $y_{c,t,i,j}$ is the actual probability of the j-th bin of the i-th time series. $y_{c,e,i,j}$ is obtained by (21). $y_{c,t,i,j}$ is either 0 or 1. It is equal to 1 if the j-th bin is the actual class of the i-th time series.

The network parameter set θ is updated by using gradient descent algorithm

$$\theta_{t+1} = \theta_t - \lambda\frac{\partial \ell}{\partial \theta_t} \tag{27}$$

where λ is the learning rate, and $\theta \in \{\mathbf{w}_{xi}, \mathbf{w}_{hi}, \mathbf{w}_{xf}, \mathbf{w}_{hf}, \mathbf{w}_{xo}, \mathbf{w}_{ho}, \mathbf{w}_{xg}, \mathbf{w}_{hg}, \mathbf{b}_i, \mathbf{b}_f, \mathbf{b}_o, \mathbf{b}_g, \mathbf{w}_r, \mathbf{b}_r\}$ in the regression model or $\theta \in \{\mathbf{w}_{xi}, \mathbf{w}_{hi}, \mathbf{w}_{xf}, \mathbf{w}_{hf}, \mathbf{w}_{xo}, \mathbf{w}_{ho}, \mathbf{w}_{xg}, \mathbf{w}_{hg}, \mathbf{b}_i, \mathbf{b}_f, \mathbf{b}_o, \mathbf{b}_g, \mathbf{w}_c, \mathbf{b}_c\}$ in the classification model.

After an epoch has finished, the entire training set is reused for the next epoch. The training process is terminated if the loss remains over a few epochs, implying that the network performance has been maximized. The entire validation set is evaluated after a number of epochs. The loss obtained with the validation set is used to 1) monitor the training process, 2) prevent the network from overfitting the training data, and 3) tune the hyperparameters used in the training process. Upon completing the training process, the performance of the framework is evaluated with the testing set. The training process will be terminated manually if the losses obtained with the training set and the validation set diverge.

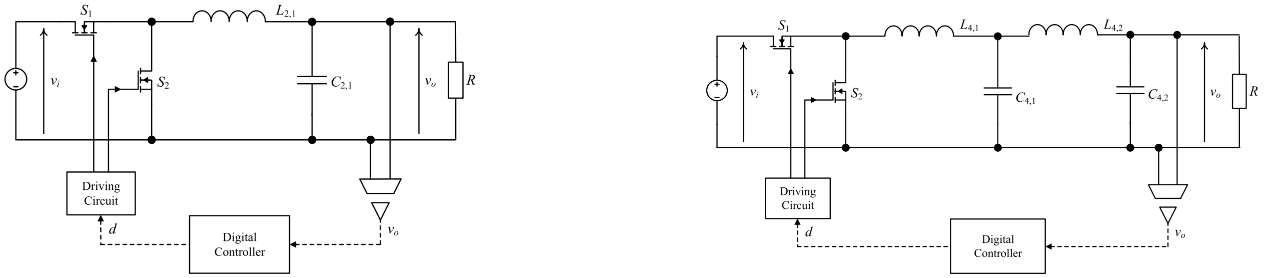
V. EXPERIMENTAL VERIFICATIONS

The effectiveness of the proposed framework has been evaluated on a 240W, 100V / 48V DC-DC buck converter. Two types of output filters have been tested. They are second-order and fourth-order filters. Fig. 10(a) and 10(b) show the schematics of the converter with the two types of filters, respectively. The nominal values of the components for the two types of filters are tabulated in Table 4.

The default controller is of Type-II and the control parameters are given in Table 2. The switching frequency and the sampling frequency of the controller are both 100kHz. Fig. 11 shows the photo of the ROF, which was built in the laboratory. Each IBBU consists of multiple fixed-value inductors and CBBU consists of multiple fixed- value capacitors. Two IBBUs and two CBBUs are used. The nominal (Nom), minimum (Min) and maximum (Max) values of the inductors and capacitors in the output filters in Fig. 10(a) and (b) are tabulated in Table 4.

The training process is performed on a computer using AMD Ryzen 3950X CPU and Nvidia RTX3080Ti GPU, the average training time is less than 13 minutes for the second-order output filter and less than 60 minutes for the fourth-order output filter. The network and training parameters are given in Table 5. During the training process, 64 samples are randomly sampled as a batch to update the network and the entire training set is processed 100 times (Epoch).

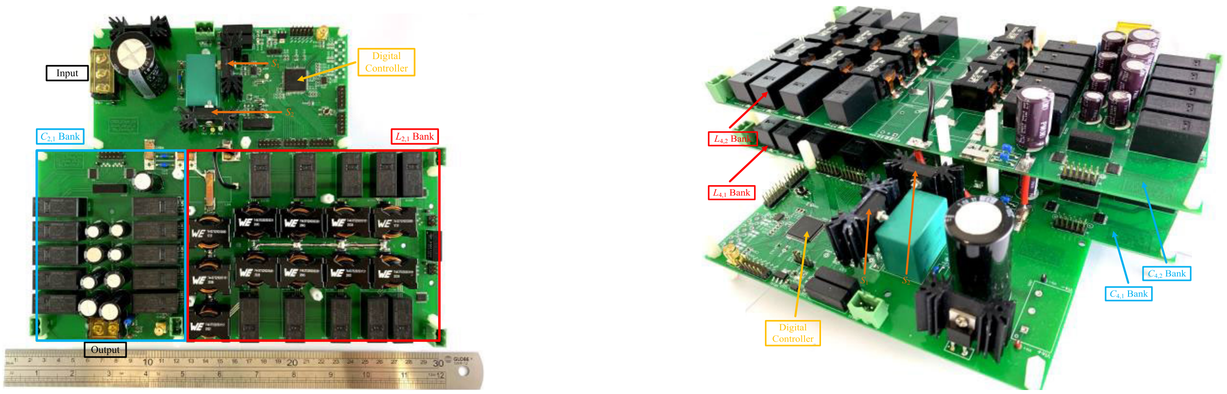
After the training process on a PC is completed, the weights and parameters in the LSTM network are downloaded to the digital controller to control the PCS and identify output filter parameters. Since they have already included all operational conditions in the PCS, it is unnecessary to train the LSTM network for individual converters. Thus, in the following discussions, the resources used in the training process are not



(a) With second-order output filter

(b) With fourth-order output filter

FIGURE 10. Circuit schematics of the converter.



(a) With second order output filter

(b) With fourth order output filter

FIGURE 11. Photos of the converter with the ROF.

included. The additional computational resources, including storage space and computation time, required to perform the proposed parameter estimation framework (‘‘framework’’) on the digital controller are briefly given in Table 6.

Nowadays, microcontrollers contain a lot of memory. For example, STM32F4 has 512kB flash memories. The framework only occupies 0.93% (4.768 / 512) of the memory space in executing the framework for a converter with *LC* filter, and 3.62% (18.54 / 512) of memory space in executing the framework for a converter with *LCLC* filter. It can be easily integrated into a microcontroller. Even if the operation of the framework is non-time critical, it only takes 9.3ms and 320ms to extract *LC* and *LCLC* filter parameters, respectively, in the LSTM computation, and the two perturbations take 8ms. The overall time taken to execute the framework is 17.3ms and 328ms to extract *LC* and *LCLC* filter parameters, respectively.

By substituting the trained weights and bias of the LSTM obtained in the experiment into (23) and (24), both inequalities are satisfied. It implies that if the sampling noise is bounded, the state of the LSTM would also be bounded and global asymptotic stability, demonstrating the robustness of the trained LSTM network to sampling noise.

Fig. 12 shows the waveforms of the output voltage v_o and inductor current i_L with the second-order output filter. Fig. 13 shows the waveforms of capacitor voltage $v_{C_{4,1}}$, the output voltage v_o, and inductor currents $i_{L_{4,1}}$ and $i_{L_{4,2}}$ with the fourth-order output filter. In both cases, four operating intervals, including $[t_0, t_1]$, $[t_1, t_2]$, $[t_2, t_3]$, and $[t_3, t_4]$, are identified. During the time intervals $[t_0, t_1]$ and $[t_3, t_4]$, v_o is regulated by the default controller. During the time interval $[t_1, t_2]$ and $[t_2, t_3]$, the control is transferred to the framework. The framework lasts for 8ms. In the time interval $[t_1, t_2]$, the duty cycle of the main switch is initially perturbed by 1%

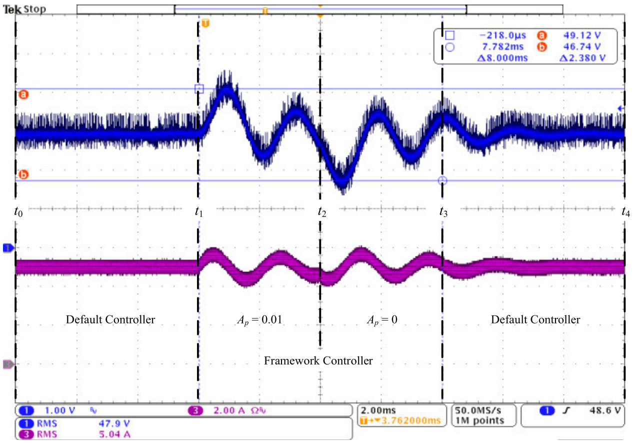


FIGURE 12. Waveforms of v_o (1 V/div, 45 V offset) [blue] and i_L (2 A/div) [purple]. (Timebase: 2 ms/div).

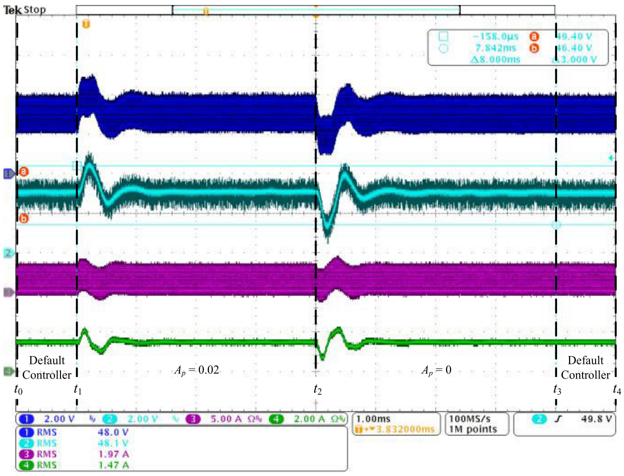


FIGURE 13. Waveforms of $v_{C4,1}$ (2 V/div, 45 V offset) [blue], v_o (2 V/div, 45 V offset) [cyan], $i_{L4,1}$ (5 A/div) [purple] and $i_{L4,2}$ (5 A/div) [green].

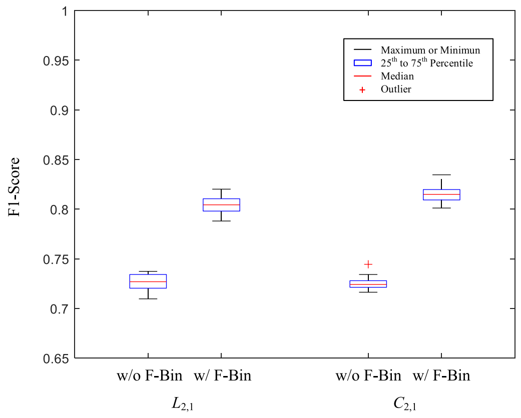


FIGURE 14. Histograms of the NRMSE of the second-order output filter. (a) $L_{2,1}$. (b) $C_{2,1}$.

for the converter with the second-order output or 2% for the converter with the fourth-order output filter. Then in the time interval [t_2, t_3], the injected signals are removed. It causes v_o having a peak-to-peak variation of 2.38V and 3.0V, respectively for the two types of output filters. The performance in extracting the two types of output filters is given below.

A. SECOND-ORDER OUTPUT FILTER

1) REGRESSION

The results are obtained from evaluating the testing set that includes datasets of the output power changing from 20% of the rated power to 100% of the rated power and input voltage changing from 80V to 120V. The histograms of the normalized root-mean-square error (NRMSE) of each sample in the testing set in 20 runs are shown in Fig. 14. Table 7 shows the mean RMSE, NRMSE, and the 95% confidence interval (CI) of the results. The results show that the NRMSE of the estimated values of $L_{2,1}$ and $C_{2,1}$ are 2.35% and 2.25%, respectively. Majority of the NRMSE is about 2%, which is practically acceptable.

TABLE 6. Additional computational resources for the framework.

Resource	Requirements / Operations	Functions
Storage Space		
Data storage	4 bytes	These two variables are used to define (2 bytes) and keep track (2 bytes) the operational phase of the framework. 0 : Framework disengaged 1 : Preparation for the parameter extraction 2 : Launch of perturbation by increasing duty cycle 3 : Launch of perturbation by decreasing duty cycle 4 : Execution of the parameter extraction procedure and Transfer of control to default controller
	8 bytes	The memories are used to store the control parameters, including $K_{p,f}$ (4 bytes) and $K_{i,f}$ (4 bytes), for the framework.
	80 bytes[1] 800 bytes[2]	The memories are used to store the time series of the duty cycle for the parameter extraction procedure.
	4,768 bytes[1] 17,728 bytes[2]	The memories are used to store weights and parameters used in the LSTM.
Total	4,860 bytes[1] or 4.75 kB[1] 18,540 bytes[2] or 18.11 kB[2]	Total additional storage space required.
Computation		
Multiplication	43,200[1] 1,273,600[2]	The total number of multiplications conducted in the parameter extraction procedure.
Addition	26,560[1] 940,800[2]	The total number of additions conducted in the parameter extraction procedure.
Total	9.3ms[1] 320ms[2]	Total LSTM computation time on a 168MHz, STM32F4 with FPU microcontroller.

Note:
[1] *LC* filter.
[2] *LCLC* filter.

TABLE 7. Mean RMSE, NRMSE, and 95% CI of the regression model of the second-order output filter.

	Output filter parameter	
	$L_{2,1}$	$C_{2,1}$
RMSE	11.031 μH	7.410 μF
NRMSE	2.35 %	2.25 %
95% CI	± 0.258 μH	± 0.108 μF

TABLE 8. Mean F1-Score and 95% CI of the classification models of the second-order output filter.

	Output filter parameter	
	$L_{2,1}$	$C_{2,1}$
F1-Score	0.805	0.815
95% CI	± 0.0041	± 0.0038

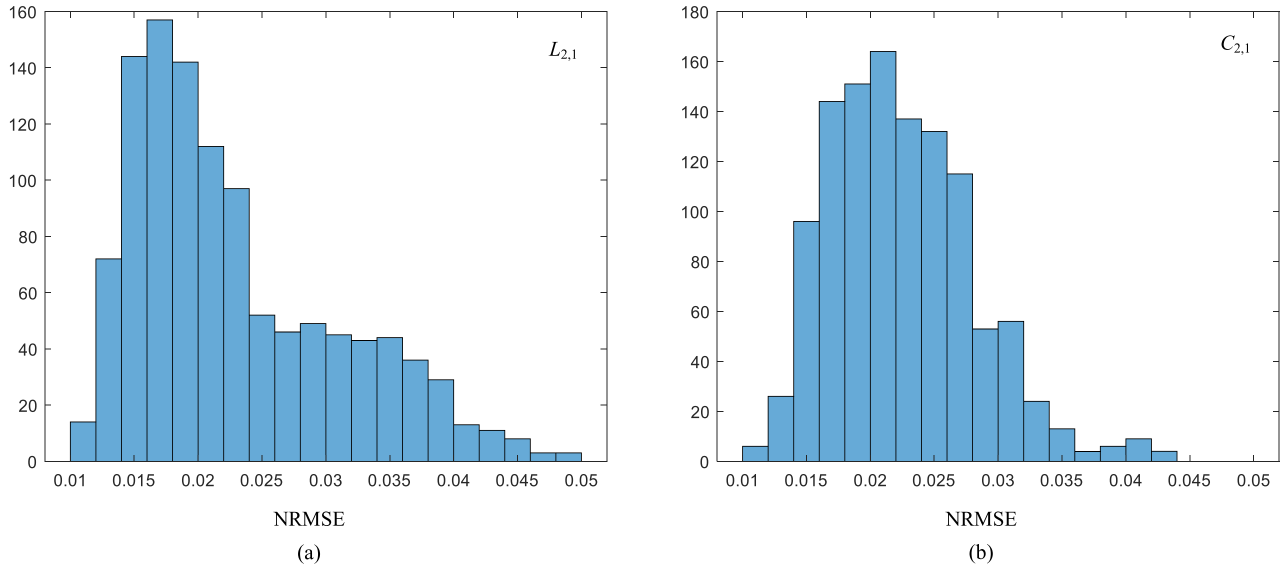


FIGURE 15. Statistical populations of the F1-scores of the $L_{2,1}$ classifiers and the $C_{2,1}$ classifiers.

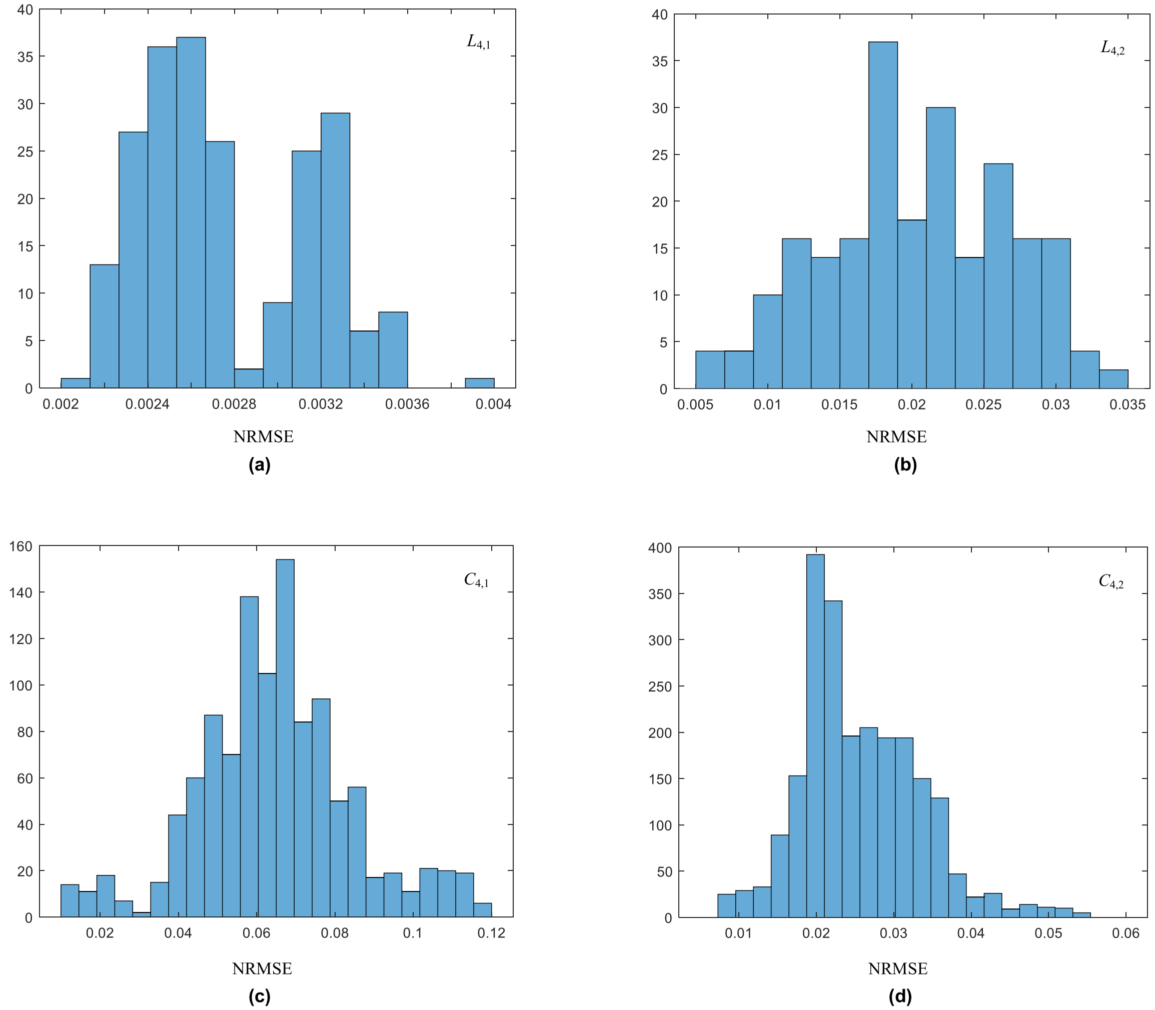


FIGURE 16. Histograms of the NRMSE of the fourth-order output filter. (a) $L_{4,1}$. (b) $L_{4,2}$. (c) $C_{4,1}$. (d) $C_{4,2}$.

TABLE 9. Mean RMSE, NRMSE, and 95% CI of the regression model of the fourth-order output filter.

	Output filter parameter			
	$L_{4,1}$	$L_{4,2}$	$C_{4,1}$	$C_{4,2}$
RMSE	0.188 μH	1.387 μH	1.428 μF	1.316 μF
NRMSE	0.28 %	2.04 %	6.49 %	2.80 %
95% CI	± 0.004 μH	± 0.0567 μH	± 0.0259 μF	± 0.0265 μF

2) CLASSIFICATION

The statistical populations of the F1-scores of the $L_{2,1}$ classifiers and the $C_{2,1}$ classifiers trained in 20 runs are shown in Fig. 15. Identical to the regression model, the testing set contains the same output power and input voltage variations. Table 8 shows the mean F1-scores and the 95% CI of the results. The results show that the F1-scores of the $L_{2,1}$ classifiers and the $C_{2,1}$ classifiers are 0.805 and 0.815, respectively. By introducing the F-Bin, the F1-scores of both $L_{2,1}$ classifiers and $C_{2,1}$ classifiers are improved by more than 0.078.

B. FOURTH-ORDER OUTPUT FILTER

For the fourth-order output filter, regression models are trained to estimate the four filter parameters. It is because, on the one hand, the capabilities of the proposed framework have been demonstrated in the second-order case. On the other hand, the regression model provides better understanding of the filter parameter. The histograms of the NRMSE of each output filter parameter are shown in Fig. 16. Table 9 shows the mean RMSE, NRMSE, and the 95% CI of the results. The results show that the NRMSE of the estimated values of $L_{4,1}$, $L_{4,2}$, $C_{4,1}$ and $C_{4,2}$ are 0.28%, 2.04%, 6.49% and 2.80%, respectively, which is generally acceptable except there is room for improvement in estimating $C_{4,1}$.

VI. CONCLUSION

A "plug-and-play" tiny artificial-intelligence-empowered output filter parameter extraction framework for digital power has been presented. The framework is a software module that can be incorporated into a digital controller to extract output filter parameters without adding extra sensors or modifying the circuit. An LSTM network extracts the filter parameters by recognizing the response of the duty cycle through the framework after subjecting a perturbation into the duty cycle. Detailed descriptions on the pre-processing of sampled duty cycle, alignment of data for the input layer of the LSTM, the operation of the LSTM cell, the training of the LSTM network, and method to test the robustness of the LSTM against sampling noise have been given. The proposed framework has been tested and evaluated successfully on a 240W 100V/48V buck DC/DC converter with its output filter configured as either a second-order *LC* or a fourth-order *LCLC* filter. Regression models and classification models are both trained to estimate the filter parameter with acceptable results in practice. The proposed framework can be applied in industrial applications where the online identification of filter parameters is required especially if no additional sensors are expected to be added. These applications include but are not limited to fault diagnosis, aging monitoring, model predictive control, EMI design, etc. In the future, the proposed framework can be further improved by incorporating software simulator training instead of building the ROF for data collection.

REFERENCES

[1] (2022). *Digital Power Supply, Texas Instruments*. [Online]. Available: https://www.ti.com/power-management/digital-power/overview.html

[2] S.-Y. Kim, Y.-J. Park, I. Ali, T. T. K. Nga, H.-C. Ryu, Z. H. N. Khan, S.-M. Park, Y. G. Pu, M. Lee, K. C. Hwang, Y. Yang, and K.-Y. Lee, "Design of a high efficiency DC–DC buck converter with two-step digital PWM and low power self-tracking zero current detector for IoT applications," *IEEE Trans. Power Electron.*, vol. 33, no. 2, pp. 1428–1439, Feb. 2018.

[3] Q. Huang, C. Zhan, and J. Burm, "A 4-MHz digitally controlled voltage-mode buck converter with embedded transient improvement using delay line control techniques," *IEEE Trans. Circuits Syst. I, Reg. Papers, Reg. Papers*, vol. 67, no. 11, pp. 4029–4040, Nov. 2020.

[4] C. Wang, S. Xu, S. Lu, and W. Sun, "A low-cost constant current control method for DCM and CCM in digitally controlled primary-side regulation flyback converter," *IEEE J. Emerg. Sel. Topics Power Electron.*, vol. 6, no. 3, pp. 1483–1494, Sep. 2018.

[5] H. S. Nair and N. L. Narasamma, "An improved digital algorithm for boost PFC converter operating in mixed conduction mode," *IEEE J. Emerg. Sel. Topics Power Electron.*, vol. 8, no. 4, pp. 4235–4245, Dec. 2020.

[6] G. Zhou, G. Mao, H. Zhao, W. Zhang, and S. Xu, "Digital average voltage/digital average current predictive control for switching DC–DC converters," *IEEE J. Emerg. Sel. Topics Power Electron.*, vol. 6, no. 4, pp. 1819–1830, Dec. 2018.

[7] P. S. Crovetti, M. Usmonov, F. Musolino, and F. Gregoretti, "Limit-cycle-free digitally controlled DC–DC converters based on dyadic digital PWM," *IEEE Trans. Power Electron.*, vol. 35, no. 10, pp. 11155–11166, Oct. 2020.

[8] S. Biricik and H. Komurcugil, "Optimized sliding mode control to maximize existence region for single-phase dynamic voltage restorers," *IEEE Trans. Ind. Informat.*, vol. 12, no. 4, pp. 1486–1497, Aug. 2016.

[9] J. C.-T. Lai, H. S.-H. Chung, Y. He, W. Wu, and F. Blaabjerg, "Wideband series harmonic voltage compensator for enhancing stability of microgrids," *IEEE Trans. Power Electron.*, vol. 37, no. 8, pp. 9687–9702, Aug. 2022.

[10] R. Medjoudj, H. Bediaf, and D. Aissani, "Power system reliability: Mathematical models and applications," in *System Reliability*. London, U.K.: IntechOpen, 2017, pp. 279–298.

[11] H. Givi, E. Farjah, and T. Ghanbari, "A comprehensive monitoring system for online fault diagnosis and aging detection of non-isolated DC–DC Converters' components," *IEEE Trans. Power Electron.*, vol. 34, no. 7, pp. 6858–6875, Jul. 2019.

[12] T. Patcharoen and A. Ngaopitakkul, "Transient inrush and fault current signal extraction using discrete wavelet transform for detection and classification in shunt capacitor banks," *IEEE Trans. Ind. Appl.*, vol. 56, no. 2, pp. 1226–1239, Apr. 2020.

[13] Z. Zhao, P. Davari, W. Lu, H. Wang, and F. Blaabjerg, "An overview of condition monitoring techniques for capacitors in DC-link applications," *IEEE Trans. Power Electron.*, vol. 36, no. 4, pp. 3692–3716, Apr. 2021.

[14] H. Wang, M. Liserre, and F. Blaabjerg, "Toward reliable power electronics: Challenges, design tools, and opportunities," *IEEE Ind. Electron. Mag.*, vol. 7, no. 2, pp. 17–26, Jun. 2013.

[15] K. Ma, H. Wang, and F. Blaabjerg, "New approaches to reliability assessment: Using physics-of-failure for prediction and design in power electronics systems," *IEEE Power Electron. Mag.*, vol. 3, no. 4, pp. 28–41, Dec. 2016.

[16] F. Deng, Q. Wang, D. Liu, Y. Wang, M. Cheng, and Z. Chen, "Reference submodule based capacitor monitoring strategy for modular multilevel converters," *IEEE Trans. Power Electron.*, vol. 34, no. 5, pp. 4711–4721, May 2019.

[17] M. W. Ahmad, N. Agarwal, P. N. Kumar, and S. Anand, "Low-frequency impedance monitoring and corresponding failure criteria for aluminum electrolytic capacitors," *IEEE Trans. Ind. Electron.*, vol. 64, no. 7, pp. 5657–5666, Jul. 2017.

[18] N. Agarwal, M. W. Ahmad, and S. Anand, "Quasi-online technique for health monitoring of capacitor in single-phase solar inverter," *IEEE Trans. Power Electron.*, vol. 33, no. 6, pp. 5283–5291, Jun. 2018.

[19] P. Sundararajan, M. H. M. Sathik, F. Sasongko, C. S. Tan, J. Pou, F. Blaabjerg, and A. K. Gupta, "Condition monitoring of DC-link capacitors using Goertzel algorithm for failure precursor parameter and temperature estimation," *IEEE Trans. Power Electron.*, vol. 35, no. 6, pp. 6386–6396, Jun. 2020.

[20] L. Ren, C. Gong, and Y. Zhao, "An online ESR estimation method for output capacitor of boost converter," *IEEE Trans. Power Electron.*, vol. 34, no. 10, pp. 10153–10165, Oct. 2019.

[21] B. X. Li and K. S. Low, "Low sampling rate online parameters monitoring of DC–DC converters for predictive-maintenance using biogeography-based optimization," *IEEE Trans. Power Electron.*, vol. 31, no. 4, pp. 2870–2879, Apr. 2016.

[22] J. Poon, P. Jain, I. C. Konstantakopoulos, C. Spanos, S. K. Panda, and S. R. Sander, "Model-based fault detection and identification for switching power converters," *IEEE Trans. Power Electron.*, vol. 32, no. 2, pp. 1419–1430, Feb. 2017.

[23] K. Laadjal, M. Sahraoui, A. J. M. Cardoso, and A. M. R. Amaral, "Online estimation of aluminum electrolytic-capacitor parameters using a modified prony's method," *IEEE Trans. Ind. Appl.*, vol. 54, no. 5, pp. 4764–4774, Sep./Oct. 2018.

[24] R. W. Erickson and D. Maksimovic, "Controller design," in *Fundamental of Power Electronics*, 2nd ed. Norwell, MA, USA: Kluwer, 2001, ch. 9, pp. 331–369.

[25] G. F. Franklin, J. D. Powell, and A. Emami-Naeini, *Feedback Control of Dynamic Systems*. 4th ed. Upper Saddle River, NJ, USA: Prentice-Hall, 2001.

[26] C. H. Lee, "A survey of PID controller design based on gain and phase margins," *Int. J. Comput. Cognition.*, vol. 2, no. 3, pp. 63–100, Sep. 2004.

[27] R. Imaizumi, R. Masumura, S. Shiota, and H. Kiya, "Sequence-to-one neural networks for Japanese dialect speech classification," in *Proc. IEEE 9th Global Conf. Consum. Electron. (GCCE)*, Oct. 2020, pp. 933–935.

[28] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," *Neural Comput.*, vol. 12, no. 10, pp. 2451–2471, 2000.

[29] F. Bonassi, E. Terzi, M. Farina, and R. Scattolini, "LSTM neural networks: Input to state stability and probabilistic safety verification," in *Proc. 2nd Conf. Learn. Dyn. Control.*, vol. 120, 2020, pp. 85–94.

[30] P. Edmond, "Robustness and stability of long short-term memory recurrent neural networks," M.S. thesis, Dept. Eng. Cybern., NTNU, Trondheim, Norway, 2021.

[31] R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural Comput.*, vol. 1, no. 2, pp. 270–280, 1989.