Learning-based current estimation for power converters operating in continuous and discontinuous conduction modes

Gerardo Becerra, Fredy Ruiz, Diego Patino, Minh Pham, Xuefang Lin-Shi

Learning-based Current Estimation for Power Converters Operating in Continuous and Discontinuous Conduction Modes

Gerardo Becerra[a], Fredy Ruiz[b], Diego Patino[c], Minh T. Pham[d], Xuefang Lin-Shi[d]

[a] *Universidad Nacional Abierta y a Distancia (UNAD)*
CEAD José Acevedo y Gómez, Transversal 31 No. 12-38 sur, Bogotá, Colombia
[b] *Politecnico di Milano, Piazza Leonardo da Vinci, 32, Milano, 20133, Italia*
[c] *Pontificia Universidad Javeriana*
Facultad de Ingeniería - Edificio 42, Cra. 7 No. 40-62, Bogotá, Colombia
[d] *Univ Lyon, INSA Lyon, Université Claude Bernard Lyon 1, Ecole Centrale de Lyon, CNRS, Ampère, UMR5005, , Villeurbanne, F-69100, , France*

Abstract

The problem of current estimation of switched power converters operating in continuous and discontinuous conduction modes is considered. A method is presented for the direct design of an estimator without exact knowledge of the mathematical model of the system. The structure of the proposed method is simpler than other approaches found in the literature, which use hybrid or averaged models to represent the dynamics of the power converter in each operating mode. An algorithm implementation using parallel computation and dimensionality reduction techniques for improving the execution performance is described. The method is demonstrated in the case of the pulse-width modulated SEPIC DC-DC converter, where simulation and experimental results are discussed. The proposed method shows better estimation results with respect to other well-known model-based and data-based approaches.

Keywords:
state estimation, optimal filtering, power conversion, conduction modes, dimensionality reduction

1. Introduction

In switched power converter applications, having accurate measurements of currents and voltages is required to achieve high performance in control and monitoring tasks. In particular, current measurement is critical in different feedback control strategies based on the design of separate current and voltage control loops. In this setting, special sensing circuits are required.

The most common approach is to use current transformers to induce a current for measurement. This method has good performance and is used in several industrial applications. However, the coil inductors are susceptible to saturation at high switching frequencies [1]. Another approach involves indirect current measurement using the voltage drop on a resistor. This method is simple but may have significant losses in high-power applications. For instance, in [2], a current sensor is integrated into a current-mode controller for a buck converter. Special CMOS circuitry is designed to measure the current using a resistive element. This approach couples the sensing circuit with the controller on the same chip, decreasing the flexibility of the design or its use with other control schemes. Hall-effect sensors are a popular alternative for current measurement, showing good performance with low power consumption. However, these sensors have high sensitivity to temperature drift and DC offset variation, requiring the implementation of additional compensation mechanisms [1].

An alternative approach is to estimate the currents using observers instead of measuring them directly. The basic observation approach relies on the measurement of other signals and the availability of a mathematical model for the system. With this information, the observer is able to make predictions on the system behavior, providing estimates of the unmeasured states. The classic asymptotic observer structure consists of a prediction term using a simulation of the system model subject to the measured inputs, and a correction term depending on the measured outputs. The observer design problem consists of the assignment of the correction term gains to guarantee stability in the estimation error [3]. Usually, an argument using Lyapunov functions is established to prove the observer's stability. For example, in [4], a current observer is proposed for a boost power factor correction (PFC) converter, where the nonlinear model is obtained using the averaging principle.

Most of the observers described in the literature for current estimation are based on average models for power converters operating in continuous con-

duction mode (CCM). However, under certain conditions, these converters may operate in discontinuous conduction modes (DCM). This may happen autonomously by sudden load changes or may be forced by the designer for efficiency reasons. Under DCM operation the system dynamics change, affecting the performance of observers based on the average CCM model. In the case where both CCM and DCM are present, hybrid observers [5] that simultaneously estimate the continuous and discrete states can be designed. However, these observers are hard to implement in a practical setting.

Existing works in the literature regarding the problem of observation in power converters operating in CCM and DCM rely on the availability of a mathematical model for the power converter and require using an estimator to obtain the operation mode at each time instant. For example, in [6], the design of a current observer for a PFC boost converter was presented based on an average model representing CCM and DCM. [7] considers observer-based control for a buck converter operating in CCM and DCM. The observer was based on the discrete-time LC filter model in CCM, and an integral compensation loop was included to correct the estimates when the system enters DCM. In [8], Luenberger-type switched observers were presented, and the observer gains for different operation modes were obtained by solving a system of linear matrix inequalities.

We consider an alternative approach to the problem of current estimation in PWM converters operating in CCM and DCM. A data-based method is implemented for directly providing discrete-time estimates of the average inductor current without requiring an estimator for the operating mode, and without formulating a complex mathematical model for the system. The estimates are computed from the history of recent measured inputs and outputs, and from a dataset of measurements prepared from experiments performed on the system under different operating conditions. The algorithm computes optimal error bounds on the estimated variable, yielding information on the estimation accuracy under different operating conditions. The problem structure is exploited to obtain a parallel algorithm implementation running on a graphics processing unit (GPU), and a dimensionality reduction technique is used to improve the execution performance. The resulting approach can be extended to full state observation by implementing separate estimators for each state variable.

The paper is organized as follows: Section 2 presents the problem statement, Section 3 introduces the basic concepts of optimal filtering for state estimation and Section 4 presents the main contributions. Section 5 intro-

duces the case study of a SEPIC converter, where simulation and experimental results are presented. Finally, Section 6 includes a brief discussion of the results and some perspectives for future work.

2. Problem Statement

A switched power converter can be modeled by considering all possible circuit topologies obtained from the different conducting and blocking states of the semiconductor devices. These configurations are represented by a system of switched linear differential-algebraic equations.

$$\begin{aligned} \mathbf{P}_{\sigma(t)}\dot{\mathbf{x}}(t) &= \mathbf{A}_{\sigma(t)}\mathbf{x}(t) + \mathbf{B}_{\sigma(t)}\mathbf{u}(t) + \mathbf{B}_x w(t) \\ \mathbf{y}(t) &= \mathbf{C}_{\sigma(t)}\mathbf{x}(t) + \mathbf{D}_{\sigma(t)}\mathbf{u}(t) + \mathbf{B}_y w(t) \end{aligned} \tag{1}$$

where $\mathbf{x}(t) \in \mathbb{R}^{n_x}$ is the state, $\mathbf{u}(t) \in \mathbb{R}^{n_u}$ the input and $\mathbf{y}(t) \in \mathbb{R}^{n_y}$ is the output. The terms $\mathbf{B}_x w(t)$ and $\mathbf{B}_y w(t)$ correspond to process and measurement noise, respectively. The system matrices $\mathbf{P}_\sigma$, $\mathbf{A}_\sigma$, $\mathbf{B}_\sigma$, $\mathbf{C}_\sigma$, $\mathbf{D}_\sigma$ are selected depending on the system mode $\sigma(t) \in I$, where I is a finite index set. Only one system mode is active at each time instant, and it depends on the controlled $\mathbf{s}(t)$ and uncontrolled $\boldsymbol{\delta}(t)$ switch states: $\sigma(t) = g(\mathbf{s}(t), \boldsymbol{\delta}(t))$, $\mathbf{s}(t) \in \{0,1\}^{n_s}$, $\boldsymbol{\delta}(s) \in \{0,1\}^{n_\delta}$. The state of the uncontrolled switches depends on the system states and inputs: $\boldsymbol{\delta}(t) = h(\mathbf{x}(t), \mathbf{u}(t))$. This interaction between the continuous and discrete variables in system (1) produces a hybrid dynamical system.

Assumption 1. *The modes σ in system* (1) *are activated in sequence $I = \{1, 2, \ldots, 2^{n_s+n_\delta}\}$, each with a duration of $d_\sigma T$, where T is the total switching period.*

Under assumption 1, an average model approximation of system (1) can be obtained:

$$\begin{aligned} \bar{\mathbf{P}}\dot{\mathbf{x}}(t) &= \bar{\mathbf{A}}\mathbf{x}(t) + \bar{\mathbf{B}}\mathbf{u}(t) + \mathbf{B}_x w(t) \\ \mathbf{y}(t) &= \bar{\mathbf{C}}\mathbf{x}(t) + \bar{\mathbf{D}}\mathbf{u}(t) + \mathbf{B}_y w(t) \end{aligned} \tag{2}$$

with $\bar{\mathbf{M}} = \sum_{\sigma \in I} d_\sigma \mathbf{M}_\sigma$, $\bar{\mathbf{M}} \in \{\bar{\mathbf{P}}, \bar{\mathbf{A}}, \bar{\mathbf{B}}, \bar{\mathbf{C}}, \bar{\mathbf{D}}\}$, $\mathbf{M}_\sigma \in \{\mathbf{P}_\sigma, \mathbf{A}_\sigma, \mathbf{B}_\sigma, \mathbf{C}_\sigma, \mathbf{D}_\sigma\}$. d_σ corresponds to the proportion of the total switching period duration T that mode σ is active: $\sum_{\sigma \in \mathcal{I}} d_\sigma = 1$.

A difficulty in obtaining this model is that the mode transitions due to uncontrolled switching events can not be measured directly, but need to

be estimated indirectly. An example of this approach is presented in [9] where an adaptive control law is designed for the boost converter operating in DCM based on the computation of the switching times for the sequence of modes, assuming exact knowledge of the system model. When no uncontrolled switching events are present, the model in Eq. (2) corresponds to the classical average model commonly used in the context of power converters. In the present approach, we have no knowledge of the system matrices of this model but instead assume that a dataset of input and output measurements is available.

Problem 1. *Consider the system in Eq.* (2)*. Assume matrices* $\bar{\mathbf{P}}$*,* $\bar{\mathbf{A}}$*,* $\bar{\mathbf{B}}$*,* $\bar{\mathbf{C}}$*,* $\bar{\mathbf{D}}$*,* $\mathbf{B}_x$*,* $\mathbf{B}_y$ *are unknown. Based on discrete-time measurements of* $\mathbf{u}$*,* $\mathbf{y}$ *and* $\mathbf{s}$*, obtain discrete-time estimates* $\hat{\mathbf{x}}$ *of the unmeasured state* $\mathbf{x}$*.*

3. Optimal Filtering for State Estimation of Unknown Systems

The following approach is based on the Set Membership (SM) framework for the identification of nonlinear systems [10], [11], [12]. While traditional identification methods require some *a priori* assumptions on the system class (linear, bilinear, polynomial, etc), this approach does require less restrictive assumptions in the form of bounds on the function gradients. Consider the discrete-time nonlinear system

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{F}(\mathbf{x}_k, \mathbf{u}_k, \mathbf{d}_k, w_k) \\ \mathbf{y}_k &= \mathbf{G}(\mathbf{x}_k, \mathbf{u}_k, \mathbf{d}_k, w_k) \end{aligned} \tag{3}$$

where $\mathbf{x}_k \in \mathbb{R}^{n_x}$ corresponds to the state, $\mathbf{d}_k \in \mathbb{R}^{n_d}$ the controlled input, $\mathbf{u}_k \in \mathbb{R}^{n_u}$ the uncontrolled input, $\mathbf{y}_k \in \mathbb{R}^{n_y}$ the output and w_k the noise. A causal estimator for $x_k \in \mathbf{x}_k$ is a function f mapping the past m measured values for $\{\tilde{\mathbf{d}}, \tilde{\mathbf{y}}, \tilde{\mathbf{u}}\}$ into an estimate $\hat{x}_k$:

$$\hat{x}_k = f(\tilde{\mathbf{d}}_k, \tilde{\mathbf{y}}_k, \tilde{\mathbf{u}}_k, \ldots, \tilde{\mathbf{d}}_{k-m+1}, \tilde{\mathbf{y}}_{k-m+1}, \tilde{\mathbf{u}}_{k-m+1}). \tag{4}$$

This estimator exists if system (3) is observable. The objective is to obtain a causal filter with a small estimation error $x_k - \hat{x}_k$. The filter structure in this case corresponds to a nonlinear finite impulse response (NFIR). This method is optimal in the sense of approximating a filter function f such that the worst-case estimation error is minimized. This provides a means to compute not only the estimate but the tightest error bounds, allowing

us to quantify the uncertainty associated with the estimation process. One advantage of this method is that the considered NFIR estimator is guaranteed to be bounded-input bounded-output (BIBO) stable.

Assumption 2.

1. *Functions* $\mathbf{F}$, $\mathbf{G}$ *are unknown.*
2. *The system is n-step observable [12].*
3. *The noise* w_k *is unknown but known to be bounded in* l_p*-norm.*
4. *A dataset* $\mathcal{D} = \{\tilde{\mathbf{d}}_k, \tilde{\mathbf{y}}_k, \tilde{\mathbf{u}}_k, \tilde{x}_k, k = 1, 2, \ldots, N\}$ *is available, where* $\tilde{x}_k = x_k + \xi_k$ *is the noise corrupted measurement of* x_k*, with* ξ_k *the unknown but bounded measurement noise.*

Consider the filter $\hat{x}_k = f(\tilde{\boldsymbol{\varphi}}_k)$ where the regressor $\tilde{\boldsymbol{\varphi}}_k = [\tilde{\mathbf{d}}_k^m; \tilde{\mathbf{y}}_k^m; \tilde{\mathbf{u}}_k^m]$ corresponds to a column vector concatenation of the m most recent measured values $\{\tilde{\mathbf{d}}, \tilde{\mathbf{y}}, \tilde{\mathbf{u}}\}$. The learning process allows us to find an approximation $\hat{f}$ of the unknown optimal filter f^0 (see definition 2 in [11]) using the dataset $\mathcal{D}$. Let us assume that the filter function f belongs to the set:

$$\mathcal{F}(\gamma) = \left\{ f \in C^1 : \|f'(\boldsymbol{\varphi})\| \leq \gamma, \forall \boldsymbol{\varphi} \in \boldsymbol{\Phi} \right\} \tag{5}$$

where f' is the gradient of f and $\boldsymbol{\Phi} \subseteq \mathbb{R}^{(n_d+n_y+n_u)m}$ is the regressor domain. Given a set of N samples of $\tilde{x}_i$, $\tilde{\boldsymbol{\varphi}}_i$, the feasible filter set (FFS) is defined as:

$$FFS \doteq \{ f \in \mathcal{F}(\gamma) : |\tilde{x}_i - f(\tilde{\boldsymbol{\varphi}}_i)| \leq \varepsilon, i = 1, \ldots, N \}. \tag{6}$$

This set corresponds to the smallest set containing f^0, or equivalently, the set of all systems consistent with prior information and measured data. The problem here consists of learning the values for parameters γ and ε such that the set FFS is not empty. The worst-case bounds for the estimation are given by:

$$\overline{f}(\tilde{\boldsymbol{\varphi}}_k) = \min_{i=1,\ldots,N} \left(\tilde{x}_i + \varepsilon + \gamma \left\| \tilde{\boldsymbol{\varphi}}_k - \tilde{\boldsymbol{\varphi}}_i \right\| \right), \tag{7a}$$

$$\underline{f}(\tilde{\boldsymbol{\varphi}}_k) = \max_{i=1,\ldots,N} \left(\tilde{x}_i - \varepsilon - \gamma \left\| \tilde{\boldsymbol{\varphi}}_k - \tilde{\boldsymbol{\varphi}}_i \right\| \right). \tag{7b}$$

The following result is useful for checking the validity of *a priori* data:

Theorem 1.

1. *A necessary condition for the* FFS *to be non-empty is* $\overline{f}(\tilde{\boldsymbol{\varphi}}_i) \geq \tilde{x}_i - \varepsilon$, $i = 1, \ldots, N$.

2. *A sufficient condition for the FFS to be non-empty is* $\overline{f}\left(\tilde{\boldsymbol{\varphi}}_i\right) > \tilde{x}_i - \varepsilon$, $i = 1, \ldots, N$.

Proof. See [10]. □

Using these conditions, the values γ and ε can be learned from the training dataset. A procedure for achieving this is presented in [10]. In summary, for each ε value we can find a boundary value γ^* separating the region where $FFS = \emptyset$ from the region where $FFS \neq \emptyset$. Therefore, if an estimate for ε is available, it is reasonable to choose γ as a slightly higher value than $\gamma^*(\varepsilon)$. Using these parameters, the direct filter (DF) is defined in terms of the optimal tightest bounds in Eqs. (7) as:

$$\hat{x}_k = f_c\left(\tilde{\boldsymbol{\varphi}}_k\right) \doteq \frac{1}{2}\left[\overline{f}\left(\tilde{\boldsymbol{\varphi}}_k\right) + \underline{f}\left(\tilde{\boldsymbol{\varphi}}_k\right)\right]. \tag{8}$$

4. Main Result

4.1. Current Estimation for Power Converters Operating in CCM and DCM

The approach introduced in the previous section is used to design a discrete-time estimator for average current in power converters operating in CCM and DCM. This approach does not require knowledge of the operation mode because the changes in dynamic behavior are encoded in the dataset when transitions occur between CCM and DCM.

Let us assume that the averaged power converter model operating in CCM and DCM is represented by the discrete-time unknown model in Eq. (3), where $\mathbf{x}_k$ is the state (inductor currents and capacitor voltages), $\mathbf{d}_k$ the duty cycle of the PWM signals controlling the switching devices, $\mathbf{u}_k$ the uncontrolled measured input voltage, $\mathbf{y}_k$ the measured output voltage and x_k an unmeasured unknown state variable. Also, let us assume that the nonlinearities associated with CCM—DCM transitions are represented by the nonlinear functions $\mathbf{F}$, $\mathbf{G}$ in Eq. (3). Algorithm 1 specifies the offline procedure for learning the parameters γ and ϵ required to design a direct filter for current estimation in power converters operating in CCM and DCM.

A sufficient number of experiments must be performed to prepare the dataset $\mathcal{D}$. The design of these experiments requires special considerations to obtain a sufficiently informative dataset [13]. Once a dataset $\mathcal{D}$ is available and the parameters γ and ε have been obtained, the filter estimates are computed online using algorithm 2.

Algorithm 1: Direct filter design (offline)

Result: $\mathcal{D}$, γ, ε

1. Design random test signals $\tilde{\mathbf{d}}(t)$, $\tilde{\mathbf{u}}(t)$ for driving the power converter to operate under varied conditions (CCM and DCM)
2. Measure $\tilde{\mathbf{y}}(t)$, $\tilde{x}(t)$. The sampling time should be sufficiently high to obtain detailed waveforms of the switching behavior
3. Compute $\bar{\mathbf{y}}(t) = \texttt{average}(\tilde{\mathbf{y}}(t))$, $\bar{x}(t) = \texttt{average}(\tilde{x}(t))$, with $\texttt{average}(\cdot)$ a non-causal filter
4. Obtain $\tilde{\mathbf{y}}_k = \texttt{resample}(\bar{\mathbf{y}}(t), T_s)$, $\tilde{x}_k = \texttt{resample}(\bar{x}(t), T_s)$, $\tilde{\mathbf{d}}_k = \texttt{resample}(\tilde{\mathbf{d}}(t), T_s)$, $\tilde{\mathbf{u}}_k = \texttt{resample}(\tilde{\mathbf{u}}(t), T_s)$, with T_s a sample time sufficiently small to capture the low frequency behavior of the system
5. Prepare dataset $\mathcal{D} = \{\tilde{\boldsymbol{\varphi}}_i, \tilde{x}_i, i = 1, 2, \ldots, N\}$ using $\tilde{\mathbf{d}}_k$, $\tilde{\mathbf{y}}_k$, $\tilde{\mathbf{u}}_k$, $\tilde{x}_k$
6. Take $\varepsilon = \|\tilde{x}(t) - \bar{x}(t)\|_\infty$
7. Take $\gamma^* = \min \gamma$, subject to $\overline{f}(\tilde{\boldsymbol{\varphi}}_i) > \tilde{x}_i - \varepsilon$, $i = 1, \ldots, N$ (sufficient condition in Theorem 1 [11])

Algorithm 2: Current estimation for power converters (online)

Data: $\mathcal{D}$, γ, ε

Result: $\hat{x}_k$

while *true* **do**

1. Measure $\tilde{\mathbf{d}}_k$, $\tilde{\mathbf{y}}_k$, $\tilde{\mathbf{u}}_k$
2. Construct the regressor $\tilde{\boldsymbol{\varphi}}_k$
3. Compute the optimal tightest bounds $\overline{f}(\tilde{\boldsymbol{\varphi}}_k)$, $\underline{f}(\tilde{\boldsymbol{\varphi}}_k)$ using Eqs. (7), with $\tilde{\boldsymbol{\varphi}}_i$, $\tilde{x}_i$ available in dataset $\mathcal{D}$
4. Compute the DF estimate $\hat{x}_k$ using Eq. (8);

end

4.2. Parallel Implementation of the DF Estimator

In practice, obtaining the tightest upper and lower bounds in Eqs. (7) requires computing at each sample time k the distance between the measured regressor $\tilde{\boldsymbol{\varphi}}_k$ with respect to each of the regressors $\tilde{\boldsymbol{\varphi}}_i$, $i = 1, \ldots, N$ in the dataset. This computation can become prohibitively expensive as the number of regressors in the dataset grows. To address this problem, [14] proposed an approximation of the optimal filter f_c by computing the DF estimates on a grid defined over the regressor space. The estimates are stored in a look-up table, reducing the computation speed at the cost of a decrease in estimation accuracy. The problem with this approach is that increasing the regressor length m exponentially increases the amount of memory required to store the resulting look-up table.

In this work we propose an alternative approach: we exploit the direct filter algorithm structure to obtain a parallel implementation that can be efficiently executed using concurrent processing units, such as those available on graphical processing units (GPUs).

Notice that the most expensive task in algorithm 2 is step 3: computing the optimal tightest bounds $\overline{f}(\tilde{\boldsymbol{\varphi}}_k)$, $\underline{f}(\tilde{\boldsymbol{\varphi}}_k)$. This step can be broken up into basic operations to be executed independently over large amounts of data, allowing a parallel implementation. This implementation was achieved using CUDA (Compute Unified Device Architecture), a framework for general-purpose GPU programming [15] created by NVIDIA. In this framework, the parallel tasks are known as *kernels*, which are executed simultaneously over thousands of concurrent threads. Three *kernels* were implemented for each iteration of algorithm 2:

1. $\psi_i^j = (\tilde{\varphi}_k^j - \tilde{\varphi}_i^j)^2, \;\; i = 1, \ldots, N, \;\; j = 1, \ldots, 3m.$
2. $\Delta_i = \sqrt{\sum_{j=1}^{3m} \psi_i^j}, \;\; \overline{f}_i = \tilde{v}_i + \varepsilon + \gamma\Delta_i, \;\; \underline{f}_i = \tilde{v}_i - \varepsilon - \gamma\Delta_i.$
3. $\overline{f} = \min_{i=1,\ldots,N}(\overline{f}_i), \;\; \underline{f} = \max_{i=1,\ldots,N}(\underline{f}_i).$

The first kernel computes, at each sample time k, the squared difference between the j-th element in the regressor $\tilde{\boldsymbol{\varphi}}_k$ and the j-th element in each regressor $\tilde{\boldsymbol{\varphi}}_i$ in the dataset.This computation is performed independently by $3mN$ threads. The second kernel computes the sum of the squared differences using a parallel reduction using N concurrent threads. The third kernel computes the minimum and maximum values also as parallel reductions using N concurrent threads. Notice that these *kernels* compute the 2-norm of the regressor difference, but other norms can be similarly implemented.

4.3. PCA dimensionality reduction in regressor datasets

Principal Component Analysis (PCA) [16] is a technique used in many image processing and machine learning applications. It is used in the present context to reduce the regressor lengths in the DF datasets. The method considers the regressors as $3m$-dimensional vectors and finds a change of basis that maps them into a new space where the components are ranked according to the amount of information they contain. This is achieved by constructing the covariance matrix of the regressor dataset and computing its eigenvalues and eigenvectors. The eigenvectors correspond to the independent directions where the variability of the data is maximized. The associated eigenvalues provide a measure of the information contents for each of these directions and provide a way to rank them. The dimensionality reduction is achieved by taking a subset of the eigenvectors associated with most of the information content and using them to map the original regressors into a lower dimensional space. This mapping is performed offline on the regressor dataset and online on the measured regressor.

Consider the dataset $\mathcal{D}$ formed by $3m$-dimensional column vectors $\tilde{\boldsymbol{\varphi}}_i$. Organize the N vectors into a matrix $\tilde{\Phi} \in \mathbb{R}^{3m\times N}$. The goal is to find a linear transformation $\tilde{\Omega} = Z^\intercal \tilde{\Phi}$ with $Z \in \mathbb{R}^{3m\times l}$, such that the reduced-dimension regressors in $\tilde{\Omega} \in \mathbb{R}^{l\times N}$ describe the data with less variables: $l < 3m$. Algorithm 3 presents the general approach to obtain the reduced-size dataset $\tilde{\Omega}$ and the transformation matrix Z using the covariance method. Notice that the parameter ζ refers to the relative information content of the principal components compared to the complete dataset. A rule of thumb to choose this value is $\zeta = 0.9$. The columns in matrix $\tilde{\Omega}$ represent reduced-size vectors of the columns in $\tilde{\Phi}$. The same transformation is used online to map the regressor $\tilde{\boldsymbol{\varphi}}_k$ measured at time k into the lower-dimensional space: $\tilde{\boldsymbol{\omega}}_k = Z^\intercal \tilde{\boldsymbol{\varphi}}_k$.

5. Application Example: SEPIC Converter

Consider the DC-DC single-ended primary inductor converter (SEPIC) presented in Fig. 1. We used a PWM signal with a controlled duty cycle to drive the converter in open loop. The state vector is $x = [I_{L_1}, V_{C_1}, I_{L_2}, V_{C_2}]^\intercal$. The switched linear model of the SEPIC converter in Eq. (1) has the following matrices:

Algorithm 3: PCA Dataset Preparation

Data: $\mathcal{D}$

Result: Z, $\mathcal{O}$

1. Organize the regressors $\tilde{\boldsymbol{\varphi}}_i$ in dataset $\mathcal{D}$ into matrix $\tilde{\Phi} \in \mathbb{R}^{3m \times N}$.
2. Compute the empirical mean $u \in \mathbb{R}^{3m \times 1}$ with $u_j = \frac{1}{N} \sum_{i=1}^{N} \tilde{\Phi}_{ij}$.
3. Subtract the deviations from the mean: $B = \tilde{\Phi} - uh$, where $h \in \mathbb{R}^{1 \times N} = [1, 1, \ldots, 1]$.
4. Compute the covariance matrix $C = \frac{1}{N-1} BB^{\intercal}$.
5. Compute eigenvectors $V \in \mathbb{R}^{3m \times 3m}$ and eigenvalues $D \in \mathbb{R}^{3m \times 1}$ of covariance matrix C.
6. Sort the eigenvectors V in decreasing order of associated eigenvalues and save them in matrix V_s.
7. Compute the cumulative eigenvalues $g_j = \sum_{i=1}^{j} D_i$ for $j = 1, \ldots, 3m$.
8. Choose l as the smallest integer such that $g_l / g_{3m} >= \zeta$.
9. Select the first l columns of V_s and save them as matrix Z.
10. Compute the matrix of reduced-size regressors as $\tilde{\Omega} = Z^{\intercal} \tilde{\Phi}$.
11. Save the columns of matrix $\tilde{\Omega} = [\tilde{\boldsymbol{\omega}}_1, \tilde{\boldsymbol{\omega}}_2, \ldots, \tilde{\boldsymbol{\omega}}_N]$ in the new dataset $\mathcal{O} = \{(\tilde{\boldsymbol{\omega}}_i, \tilde{v}_i), i = 1, 2, \ldots, N\}$.

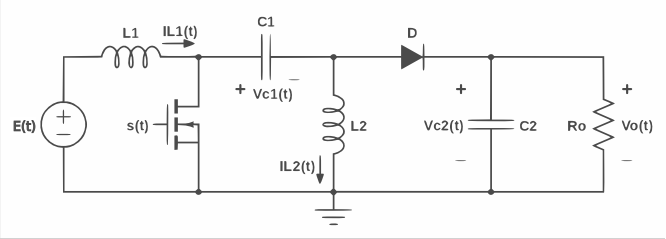

Figure 1: Schematic diagram of the SEPIC converter.

$$\mathbf{A}_{s(t),\delta(t)} = \begin{bmatrix} -R_{L1} - \beta R_{L2} & s-1 & 0 & s\delta - \delta \\ 1-s & s\delta & s - s\delta & s\delta \\ \beta & -s & \beta - R_{L2}(s+\delta-s\delta) & \delta - s\delta \\ \delta - s\delta & 0 & -\delta & -1/R_o \end{bmatrix},$$

$$\mathbf{B}_{s(t),\delta(t)} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \mathbf{P}_{s(t),\delta(t)} = \begin{bmatrix} L_1 & 0 & \beta L_1 & 0 \\ 0 & (1-s\delta)C_1 & 0 & 0 \\ 0 & 0 & (s+\delta-s\delta)L_2 & 0 \\ 0 & s\delta C_1 & 0 & C_2 \end{bmatrix},$$

$$\mathbf{C}_{s(t),\delta(t)} = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}, \mathbf{D}_{s(t),\delta(t)} = 0,$$

where $s(t) \in \{0,1\}$ is the binary signal that controls the switching of the MOSFET, $\delta(t) \in \{0,1\}$ is a binary variable that represents the uncontrolled diode state and $\beta = (1-s)(1-\delta)$. R_{L1}, R_{L2} are the equivalent internal inductor resistances and R_o is the load resistance. Table 1 shows the parameters used for the SEPIC converter in the simulation and experimental tests.

In CCM operation, the MOSFET and the diode are activated complementarily (i.e. $s = 1, \delta = 0$ and $s = 0, \delta = 1$)). In this case, the off-diagonal elements in matrix $\mathbf{P}_{s(t),\delta(t)}$ become zero, and the diagonal corresponds to the storage element values. The resulting model in Eq. (1) corresponds to a system of linear differential equations.

In DCM operation the MOSFET and diode simultaneously turn on or off, depending on the type of discontinuity (i.e. $s = 0, \delta = 0$ or $s = 1, \delta = 1$). In this case, the matrix $\mathcal{P}_{s(t),\boldsymbol{\delta}(t)}$ has nonzero elements in the off-diagonal, and one of the differential equations becomes an algebraic equation. For example, when the SEPIC converter is in discontinuous inductor current mode (DICM), the MOSFET and diode are turned off ($s = 0, \delta = 0$). In this mode, both inductors are connected in series, forcing them to have the same current value. The equivalent inductance determines the dynamics throughout the discontinuous mode, and the system has three state variables: $I_{L_{eq}}$, V_{C_1} and V_{C_2}. In discontinuous capacitor voltage mode (DCVM) the MOSFET and diode are turned on at the same time ($s = 1, \delta = 1$), forcing both capacitors in a parallel configuration, yielding a system with three state variables: $V_{C_{eq}}$, I_{L_1}, I_{L_2}. Using the averaging principle, $s(t)$ and $\delta(t)$ can be redefined as continuous variables in the interval $[0,1]$ representing the duty cycle of PWM signals with a sufficiently high constant frequency f_{PWM}. These duty cycles will represent the period fraction that each switching device

Table 1: *Parameters used in the simulation and experimental tests.*

Converter	Parameters
SEPIC	$E = 20$ V, $L_1 = 2.3$ mH, $C_1 = 190$ μF, $L_2 = 330$ μH, $C_2 = 190$ μF, $R_{L1} = 2.134$ Ω, $R_{L2} = 0.234$ Ω, $R_o = 22$ Ω, $f_{pwm} = 20$ kHz

is active. In this model, we assume there is a way to measure or estimate the uncontrolled state of the diode.

The design of an observer for estimating the state variables in this hybrid model is not an easy task. Therefore, we use the approach introduced in section 4 to design direct filters for estimating the current $I_{L_1 k}$ from sampled noisy measurements of the input voltage $u_k = E_k$, the output voltage $y_k = V_{C_2 k}$ and from knowledge of the controlled PWM duty cycle d_k. A dataset $\mathcal{D}$ is obtained by applying an amplitude-modulated pseudo-random binary signal (APRBS) to the duty cycle input of the SEPIC converter in open loop and measuring the output voltage while maintaining a constant input voltage. The APRB signal maintains a random constant value throughout a given dwell time T_d. We randomly generate the signals by defining minimum and maximum dwell times $T_{d,\min}$, $T_{d,\max}$ for both the training and test datasets. Assuming a constant load resistance R_o, the duty cycle boundary value $\bar{d}$ separating the CCM and DCM operating regions can be computed by static analysis [17], using Eq. (9). The minimum and maximum levels of the duty cycle signal applied to the converter are selected in a wide range to force operation in both CCM and DCM regions.

$$\bar{d} = 1 - \frac{2 f_{pwm} L_1 L_2}{R_o (L_1 + L_2)} = 0.2756 \tag{9}$$

To obtain the results presented in the following subsections, we first acquired the detailed converter current and voltage waveforms for the designed APRB signals. These signals are then averaged and resampled at a sufficiently high rate f_{rs} to capture the low-frequency behavior of the converter. We applied a scaling scheme to the regressor dataset as described in [10] to improve the estimation performance. Additionally, we used principal component analysis (PCA) to map the regressors to a lower dimensional space, reducing the computational cost.

5.1. Simulation Results

A detailed noiseless simulation of the SEPIC converter is run and all the variables are sampled at $f_s = 1$ MSa/s, and then averaged and resampled at $f_{rs} = 5$ kSa/s. Multiple direct filters are designed and tested for different regressor lengths m and dataset sizes N.

We developed two different direct filter implementations. The base case direct filter estimator described by Eq. (8) is DF. The direct filter with PCA-reduced dataset (DFPCA) implements the same estimator, but uses a dataset formed by regressors with reduced dimension, obtained using the procedure described in Section 4.3. We compared the performance of DF and DFPCA to other estimation approaches: extended Kalman filter (EKF), particle filter (PF), and neural networks (NN). For the model-based approaches (EKF, PF), we used the nonlinear average model of the SEPIC converter. The PF computes the evolution of 50 particles using Euler's method for 10 subintervals on each sample step. For the data-based approaches (DF, DFPCA, NN) a dataset with $N = 10000$ regressors and total regressor length $3m = 60$ was obtained from simulation. We used this dataset to learn the parameters (ε, γ) for the DF and to train a feed-forward neural network with $3m$ inputs and 1 output. After some tests, we achieved the best performance with 30 neurons in the hidden layer.

We prepared four different datasets (DS1, DS2, DS3, DS4) using several minimum and maximum dwell times for the APRB signal, specified in Table 2. For each dataset, we prepared 20 different test runs. Fig. 2 shows the performance of the estimators for these datasets, using as error measures the relative absolute error (RAE), root relative square error (RRSE), and relative worst-case error defined as:

$$\begin{aligned} \mathrm{RAE} &= 100 \left\| x - \hat{x} \right\|_1 / \left\| x - \bar{x} \right\|_1 , \\ \mathrm{RRSE} &= 100 \left\| x - \hat{x} \right\|_2 / \left\| x - \bar{x} \right\|_2 , \\ \mathrm{RWCE} &= 100 \left\| x - \hat{x} \right\|_\infty / \left\| x - \bar{x} \right\|_\infty , \end{aligned}$$

where x corresponds to the measured signal, $\bar{x}$ the average, and $\hat{x}$ the estimate. These error measures quantify several properties of each estimator: RAE indicates the mean deviation, RRSE penalizes the outliers, and RWCE indicates the maximum deviation.

For all datasets, the model-based estimators have worse performance than the data-based ones. The cause is their dependence on the nonlinear average model that doesn't provide a good approximation of the switched linear

Table 2: Constant duty cycle periods in the training and test APRB signals for different datasets.

	Training		Test	
Dataset	$T_{d,\min}$	$T_{d,\max}$	$T_{d,\min}$	$T_{d,\max}$
DS1	5 ms	5 ms	10 ms	10 ms
DS2	10 ms	10 ms	20 ms	20 ms
DS3	5 ms	10 ms	10 ms	20 ms
DS4	5 ms	10 ms	10 ms	30 ms

system, particularly during transients and DCM operation. In the case of the data-based estimators, NN has better performance than DF and DFPCA in all datasets because the parameter γ in the DF represents global worst-case bounds on partial derivatives of the approximated function, while the NN locally approximates them. Notice that in most cases, the DFPCA estimate is slightly worse than the DF because the regressor mapping into a lower-dimensional space implies a certain amount of information is lost.

Fig. 3 shows a small section of a particular test run, comparing the estimates for all the implemented methods with the switching input current I_{L1} and its average, shown in light and dark gray, respectively. Notice that the model-based approaches (EKF, PF) exhibit noticeable oscillations during transitions from CCM to DCM. These oscillations cause the high 2-norm and ∞-norm errors for these estimators. Estimates from EKF and PF also have an offset during steady-state operation. Fig. 3(a) shows the upper and lower bounds ($\overline{f}$, $\underline{f}$) for DF and DFPCA estimates, using dashed lines. Notice that the distance between these bounds grows during transients, which indicates an increase in estimation uncertainty under these conditions. Fig. 3(b) shows the detail of a CCM to DCM transition, where the switched current I_{L1} exhibits the typical discontinuous waveform. The DF and DFPCA estimators correctly track the change in the average current during the transition from the symmetric to the asymmetric waveform.

5.2. Experimental Results

We used the experimental SEPIC converter test bench shown in Fig. 4 for testing the proposed estimation approach. We executed multiple runs by applying different APRB duty cycle signals in open loop and measuring all the variables with an oscilloscope at $f_s = 2$ MSa/s. Then we resampled the acquired signals at $f_{rs} = 4$ KSa/s. We used algorithm 1 to prepare four different regressor datasets with the $T_{d,\min}$, $T_{d,\max}$ values presented in Table 2. For each dataset, we prepared $N = 10000$ training regressors with a total

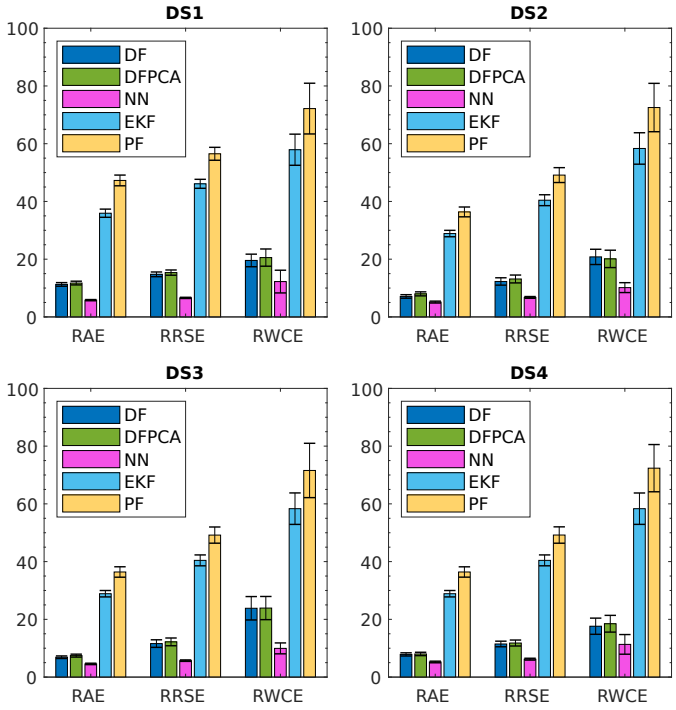

Figure 2: Simulation results: estimation error (mean ± stdev, 20 test runs of 1.5 s each) for direct filter (DF), direct filter with PCA dataset (DFPCA), neural networks (NN), extended Kalman filter (EKF) and particle filter (PF) with 4 different datasets ($m = 20$, $N = 10000$, $\varepsilon = 0.1292$).

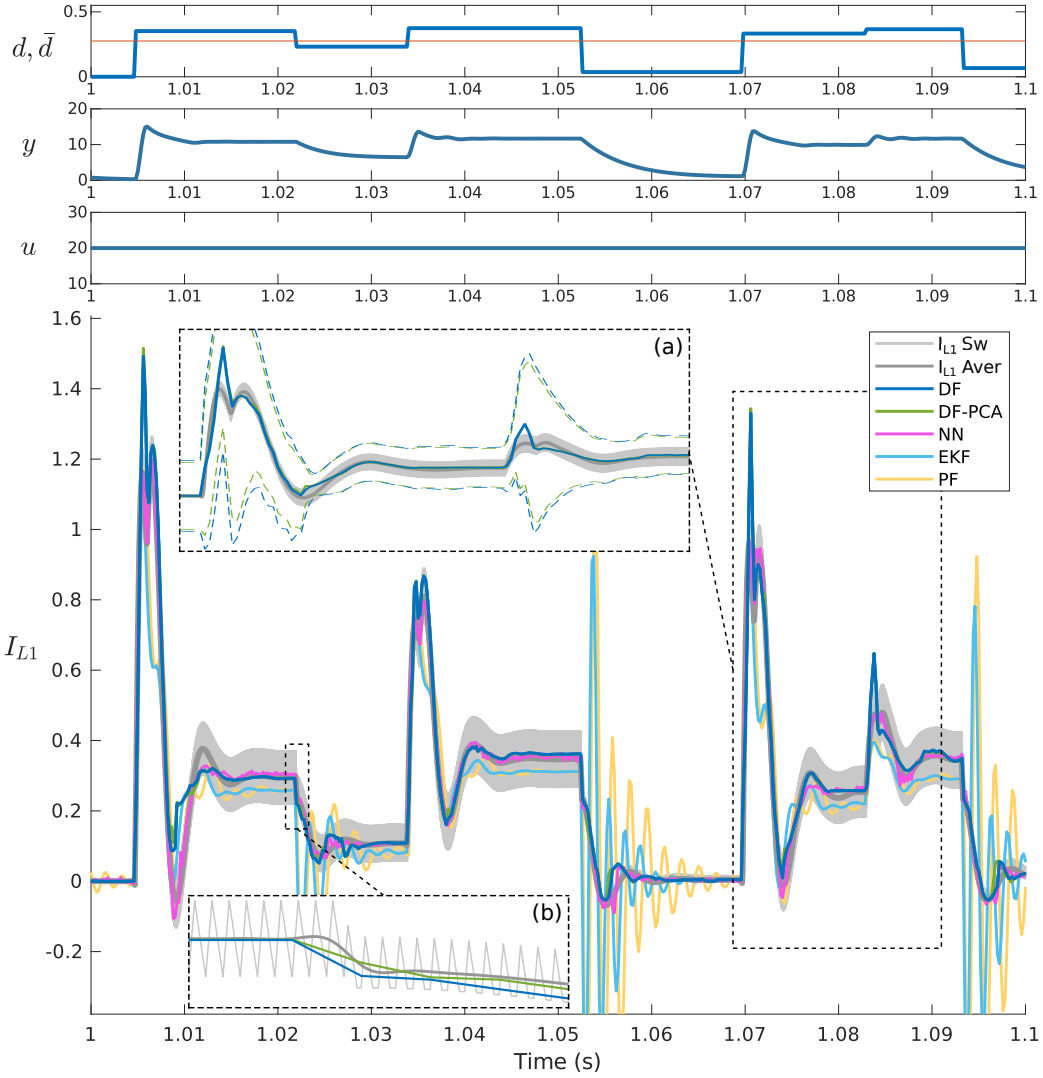

Figure 3: Simulation results (top to bottom): Duty cycle d and DCM boundary $\bar{d}$. Output voltage $y = V_{C2}$. Input voltage $u = 20$ V. Input current I_{L1} estimate with direct filter (DF), direct filter with PCA dataset (DFPCA), neural networks (NN), extended Kalman filter (EKF), and particle filter (PF).

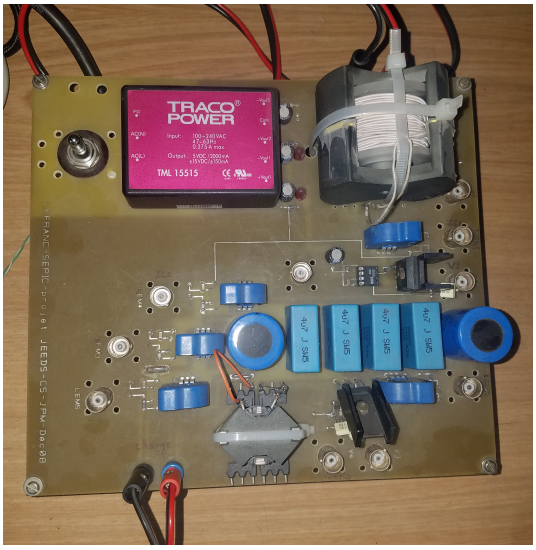

Figure 4: SEPIC converter test bench.

length of $3m = 60$ using five 500 ms captures. We also prepared 20 test signals using three 500 ms captures for each. We designed all the estimators using the same criteria discussed in the previous subsection.

Fig. 5 shows the performance comparison for all the implemented estimators. Notice that the data-based estimators (DF, DFPCA, NN) perform better than the model-based ones (EKF, PF). However, the performance of the NN estimator is worse than the DF and DFPCA estimators. In particular, the RWCE measure for NN is considerably worse than the others, indicating the presence of high amplitude outliers. Fig. 6 shows large amounts of noise for the NN estimate. The DF and DFPCA estimators show better noise rejection than all the other estimators. Fig. 6 (a) shows the detail of the upper and lower bounds ($\overline{f}_c$, $\underline{f}_c$) for DF and DFPCA. Notice that the noise increases the estimation uncertainty, but DF and DFPCA give better estimates of the average current. Fig. 6 (b) shows the DF and DFPCA estimates tracking the average current during a transition from CCM to DCM.

Let us remark that the DF in this example takes as the only uncontrolled input $u^t = E$ corresponding to the input voltage. If load disturbances were present, the estimation performance would decrease. However, we could define the load resistance as another uncontrolled input $u^t = [E, R_0]$, allowing the DF to give estimates under input and load disturbances. This scenario

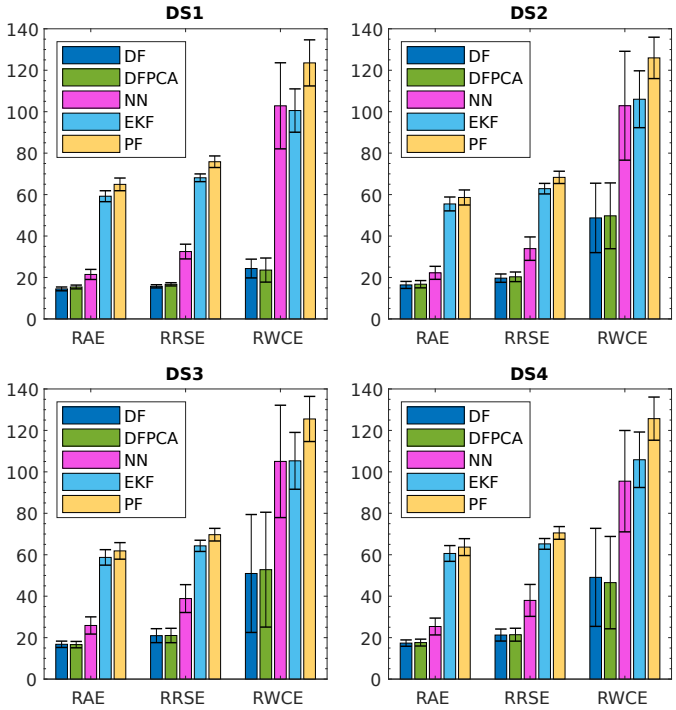

Figure 5: Experimental results: estimation error (mean ± stdev, 20 test runs of 1.5 s each) for direct filter (DF), direct filter with PCA dataset (DFPCA), neural networks (NN), extended Kalman filter (EKF), and particle filter (PF) with 4 different datasets ($m = 20$, $N = 10000$, $\varepsilon = 0.1292$).

requires including in the dataset $\mathcal{D}$ the typical load disturbance scenarios the system could be subject to.

5.3. Estimation Performance for the Parallel Implementation

Table 3 shows the estimation performance results of the DF and DFPCA estimators using the four experimental datasets available. We show the mean estimation performance loss of the DFPCA compared to the DF for the three error measures. The table shows the average execution time of one iteration of algorithm 2 for two different implementations: CPU running the DF estimator sequentially and GPU running the parallel implementation of the DFPCA estimator. The hardware used for these tests is the Nvidia Jetson TX2, using one of the four available ARM Cortex-A57 CPU cores running at 2.0 GHz and the Pascal GPU running at 1.12 GHz. The speedup of the GPU+PCA implementation compared to the base CPU case is also shown.

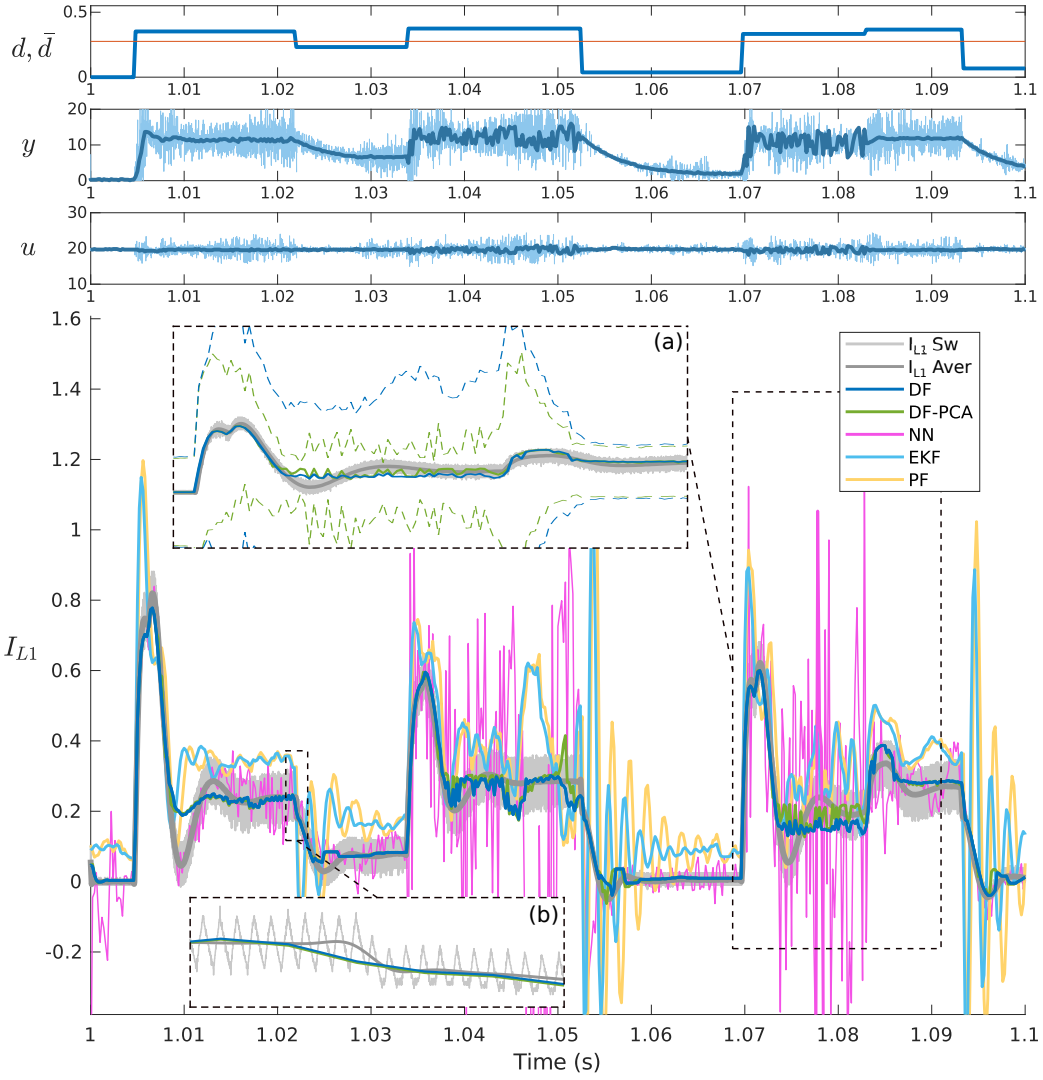

Figure 6: Experimental results (top to bottom): Duty cycle d and CCM-DCM boundary $\bar{d}$. Output voltage $y = V_{C2}$. Input current I_{L1} estimate with direct filter (DF), direct filter with PCA dataset (DFPCA), neural networks (NN), extended Kalman Filter (EKF) and particle filter (PF).

Table 3: Comparison of computational performance between DF and DFPCA implementations in CPU and GPU using PCA reduced dataset. $N = 10000$, regressor length in original dataset $3m = 60$, regressor length in PCA reduced dataset = 13.

Dataset	Mean estimation perf. loss (%)			Aver. exec. time (ms) / speedup	
	RAE	RRSE	RWCE	CPU	GPU+PCA
DS1	1.5446	1.6832	3.0170	1.6410	0.2419 / 6.7847X
DS2	0.2641	0.8726	4.0489	1.6445	0.2434 / 6.7577X
DS3	0.6703	0.1592	6.4868	1.6491	0.2437 / 6.7660X
DS4	0.4993	0.7582	1.9521	1.6428	0.2443 / 6.7235X

Notice that when we applied PCA to all datasets, the estimation performance loss was less than 2% of the absolute error. On the other hand, the algorithm implementation in the GPU using PCA allowed us to increase the computation speed more than six times. The estimation rate achieved was close to 4 kHz, corresponding to the sample time of the experimental dataset. These results show that using the reduced dimension datasets in GPU provides a computation speed gain that outweighs the estimation losses and provides a feasible implementation of the DF algorithm for real-time applications. However, the experiments performed didn't use a real-time operating system (RTOS) but a general-purpose operating system (GPOS). Consequently, we observed some jitter in the estimates produced by the DF algorithm. This behavior may render the proposed method infeasible in applications where the satisfaction of hard deterministic timing constraints is required. Development of real-time GPU applications for control systems requires additional considerations [18].

The source code implementing the DF algorithm is open and available in an online repository [19]. It includes several datasets to test the algorithm with different regressor lengths and dataset sizes.

6. Conclusion and Perspectives

We presented a method for the design of estimators without exact knowledge of the system model. We used this approach in the context of current estimation in power converters working in both CCM and DCM. Because of drastic dynamic changes in the system behavior when traversing the boundary between CCM and DCM, the average model does not provide a good approximation. This method can be used in PWM power converters and is demonstrated in the case of the SEPIC DC-DC converter. Simulation and experimental results have been presented, showing that this approach

provides good estimation results compared to other model-based and data-based methods. Also, we show that a real-time implementation of the direct filter algorithm is feasible on an embedded system by exploiting its parallel structure to run it on a GPU and applying PCA to reduce the size of the datasets.

Future extensions of this work involve improving the computational performance of the GPU implementation with additional configurations on the Jetson embedded system such as power model, clock frequencies, CPU core isolation, memory locks, and also applying a real-time patch to the Linux kernel (PREEMPT-RT) to improve the determinism compared to the stock Linux kernel. Additional improvements can be obtained with the implementation of persistent CUDA kernels to reduce the overhead of kernel launch at each sample time [20]. Regarding the power converter, we are interested in analyzing the dependence of the estimation performance on the parameter space (m, N) for different converter topologies where DCM is present. Finally, a more detailed investigation of the estimation performance loss caused by the mapping to a lower-dimensional regressor space would provide a better understanding of the advantage of applying the PCA technique in this context.

References

[1] M. Biglarbegian, S. J. Nibir, H. Jafarian, B. Parkhideh, Development of current measurement techniques for high frequency power converters, in: 2016 IEEE International Telecommunications Energy Conference (INTELEC), 2016, pp. 1–7. doi:10.1109/INTLEC.2016.7749133.

[2] Y. S. Lee, C. J. Hsu, High accuracy CMOS current sensing circuit for current mode control buck converter, Proceedings of the International Conference on Power Electronics and Drive Systems (2007) 44–48doi:10.1109/PEDS.2007.4487675.

[3] A. Alessandri, P. Coletta, Switching observers for continuous-time and discrete-time linear systems, in: Proceedings of the 2001 American Control Conference. (Cat. No.01CH37148), Vol. 3, 2001, pp. 2516–2521 vol.3. doi:10.1109/ACC.2001.946132.

[4] M. Pahlevani, S. Pan, S. Eren, A. Bakhshai, P. Jain, An adaptive nonlinear current observer for boost pfc ac/dc converters, IEEE

Transactions on Industrial Electronics 61 (12) (2014) 6720–6729. doi:10.1109/TIE.2014.2316216.

[5] M. Djemai, N. Manamanni, J. P. Barbot, Nonlinear Observer for Autonomous Switching Systems with Jumps, Springer International Publishing, Cham, 2015, pp. 103–128. doi:10.1016/j.nahs.2006.01.001.

[6] G. Cimini, G. Ippoliti, G. Orlando, M. Pirro, Current sensorless solution for pfc boost converter operating both in dcm and ccm, in: 21st Mediterranean Conference on Control and Automation, 2013, pp. 137–142. doi:10.1109/MED.2013.6608711.

[7] C. H. van der Broeck, R. W. D. Doncker, S. A. Richter, J. v. Bloh, Unified control of a buck converter for wide-load-range applications, IEEE Transactions on Industry Applications 51 (5) (2015) 4061–4071. doi:10.1109/TIA.2015.2431994.

[8] V. Spinu, M. Dam, M. Lazar, Observer design for dc/dc power converters with bilinear averaged model, IFAC Proceedings Volumes 45 (9) (2012) 204 – 209, 4th IFAC Conference on Analysis and Design of Hybrid Systems. doi:10.3182/20120606-3-NL-3011.00091.

[9] A. Alawieh, R. Ortega, H. Pillai, A. Astolfi, E. Berthelot, Voltage regulation of a boost converter in discontinuous conduction mode: A simple robust adaptive feedback controller, IEEE Control Systems Magazine 33 (3) (2013) 55–65. doi:10.1109/MCS.2013.2249431.

[10] M. Milanese, C. Novara, Set membership identification of nonlinear systems, Automatica 40 (6) (2004) 957 – 975. doi:10.1016/j.automatica.2004.02.002.

[11] C. Novara, F. Ruiz, M. Milanese, Direct filtering: A new approach to optimal filter design for nonlinear systems, IEEE Transactions on Automatic Control 58 (1) (2013) 86–99. doi:10.1109/TAC.2012.2204160.

[12] F. Ruiz, C. Novara, M. Milanese, Direct design from data of optimal filters for lpv systems, Systems and Control Letters 59 (1) (2010) 1 – 8. doi:10.1016/j.sysconle.2009.10.008.

[13] L. Ljung, System Identification: Theory for the User, Prentice Hall information and system sciences series, Prentice Hall PTR, 1999. doi:10.1016/S0005-1098(01)00214-X.

[14] J. Castano, F. Ruiz, J. Regnier, A fast approximation algorithm for set-membership system identification, IFAC Proceedings Volumes 44 (1) (2011) 4410 – 4415, 18th IFAC World Congress. doi:10.3182/20110828-6-IT-1002.03212.

[15] D. Storti, M. Yurtoglu, CUDA for Engineers: An Introduction to High-performance Parallel Computing, Always learning, Addison-Wesley, 2015.
URL https://books.google.com.co/books?id=kA1wrgEACAAJ

[16] I. Jolliffe, Springer-Verlag, Principal Component Analysis, Springer Series in Statistics, Springer, 2002.
URL https://books.google.com.co/books?id=_olByCrhjwIC

[17] N. Li, Digital control strategies for DC/DC SEPIC converters towards integration, Theses, INSA de Lyon (May 2012).
URL https://tel.archives-ouvertes.fr/tel-00760064

[18] G. A. Elliott, J. H. Anderson, Real-world constraints of gpus in real-time systems, in: 2011 IEEE 17th International Conference on Embedded and Real-Time Computing Systems and Applications, Vol. 2, 2011, pp. 48–54. doi:10.1109/RTCSA.2011.46.

[19] G. Becerra, gjbecerra/directfiltercuda: Initial release (May 2021). doi:10.5281/zenodo.4760811.

[20] A. J. Calderón Torres, Real-time high-performance computing for embedded control systems, Ph.D. thesis, UPC, Departament d'Arquitectura de Computadors (Jul 2022).
URL http://hdl.handle.net/2117/371621