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

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

Société d'Automatique, de Génie Industriel & de Productique

May 30, 2024

Agenda

Introduction

Problem Statement

Optimal Filtering for State Estimation of Unknown Systems

Results

Conclusion - Perspectives

Agenda

Introduction

Problem Statement

Optimal Filtering for State Estimation of Unknown Systems

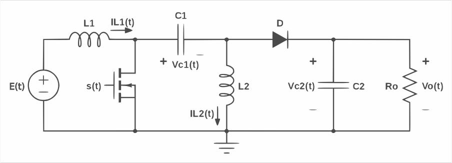
Results

Conclusion - Perspectives

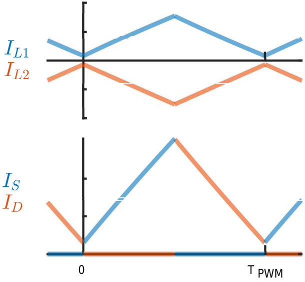
Continuous/Discontinuous Conduction Modes (CCM/DCM)

- PWM converters with two switches (e.g. MOSFET and diode).

SEPIC Converter



Continuous Conduction Mode

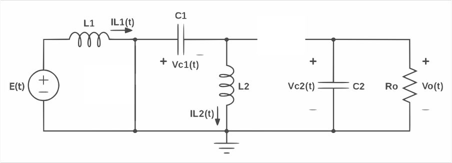


(a) CCM

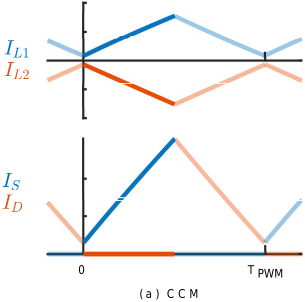
Continuous/Discontinuous Conduction Modes (CCM/DCM)

- PWM converters with two switches (e.g. MOSFET and diode).

SEPIC Converter



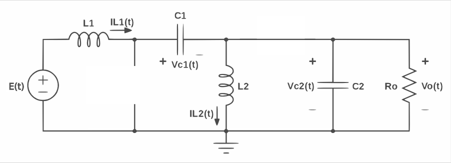
Continuous Conduction Mode



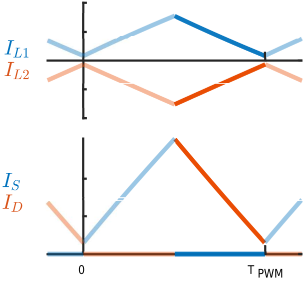
Continuous/Discontinuous Conduction Modes (CCM/DCM)

- PWM converters with two switches (e.g. MOSFET and diode).

SEPIC Converter



Continuous Conduction Mode

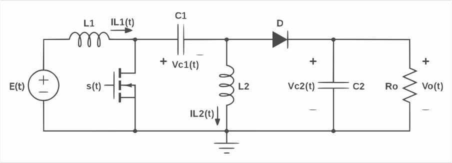


(a) CCM

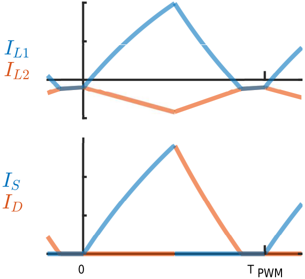
Continuous/Discontinuous Conduction Modes (CCM/DCM)

- PWM converters with two switches (e.g. MOSFET and diode).

SEPIC Converter



Discontinuous Conduction Mode

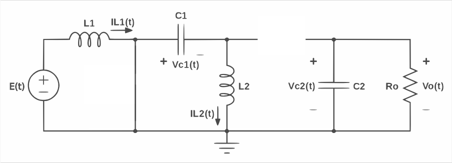


(b) DCM

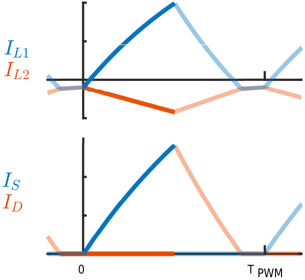
Continuous/Discontinuous Conduction Modes (CCM/DCM)

- PWM converters with two switches (e.g. MOSFET and diode).

SEPIC Converter



Discontinuous Conduction Mode

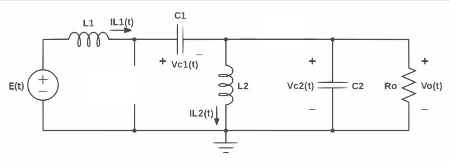


(b) DCM

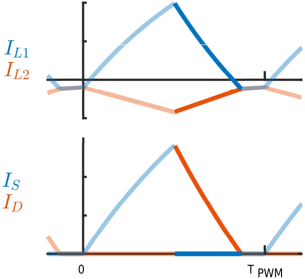
Continuous/Discontinuous Conduction Modes (CCM/DCM)

- PWM converters with two switches (e.g. MOSFET and diode).

SEPIC Converter



Discontinuous Conduction Mode

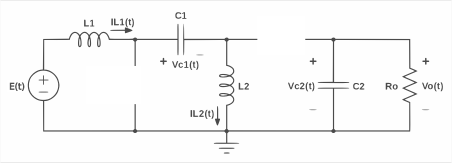


(b) DCM

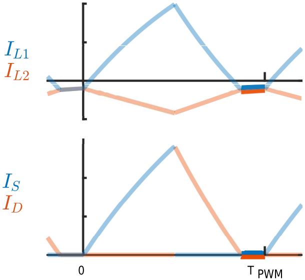
Continuous/Discontinuous Conduction Modes (CCM/DCM)

- PWM converters with two switches (e.g. MOSFET and diode).

SEPIC Converter



Discontinuous Conduction Mode

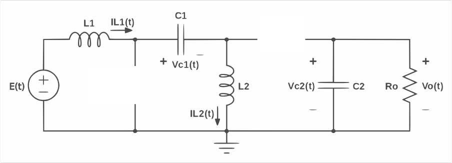


(b) DCM

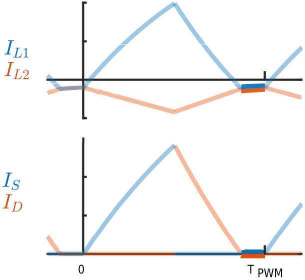
Continuous/Discontinuous Conduction Modes (CCM/DCM)

- PWM converters with two switches (e.g. MOSFET and diode).

SEPIC Converter



Discontinuous Conduction Mode



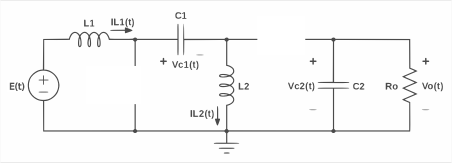
(b) DCM

- CCM-DCM transitions $\rightarrow$ change in dynamic properties.

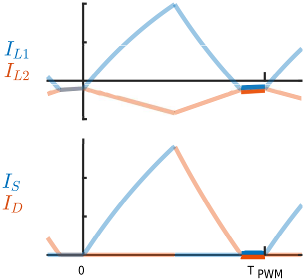
Continuous/Discontinuous Conduction Modes (CCM/DCM)

- PWM converters with two switches (e.g. MOSFET and diode).

SEPIC Converter



Discontinuous Conduction Mode



(b) DCM

- CCM-DCM transitions → change in dynamic properties.
- Hybrid behavior → complexity in observation problem.

Switched Power Converter - Hybrid Model

Société d'Automatique, de Génie Industriel & de Productique

- System of switched linear differential-algebraic equations.

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

- $\mathbf{x}(t) \in \mathbb{R}^{n_x}$: state, $\mathbf{u}(t) \in \mathbb{R}^{n_u}$: input $\mathbf{y}(t) \in \mathbb{R}^{n_y}$: output.
- $\mathbf{B}_x w(t)$: process noise , $\mathbf{B}_y w(t)$ measurement noise.
- $\mathbf{P}_\sigma$, $\mathbf{A}_\sigma$, $\mathbf{B}_\sigma$, $\mathbf{C}_\sigma$, $\mathbf{D}_\sigma$: selected by the system mode $\sigma(t) \in I$.

SEPIC Converter - Hybrid Model

- System matrices:

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

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

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

- $\mathbf{s}(t)$: Controlled switch state, $\boldsymbol{\delta}(t)$: Uncontrolled switch state

Agenda

Introduction

Problem Statement

Optimal Filtering for State Estimation of Unknown Systems

Results

Conclusion - Perspectives

Observation Problem

Problem

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

Société d'Automatique, de Génie Industriel & de Productique

Problem

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

Remarks:

- We assume no knowledge of the system model is available, but only measurements.

Observation Problem

Problem

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

Remarks:

- We assume no knowledge of the system model is available, but only measurements.
- We assume hybrid behavior (CCM & DCM operation) might be present in the system.

Contributions

- Data-based estimation for PWM power converters in a wide operation range (CCM-DCM).

Contributions

- Data-based estimation for PWM power converters in a wide operation range (CCM-DCM).
- Parallel implementation of the data-based estimation in a GPU.

Contributions

- Data-based estimation for PWM power converters in a wide operation range (CCM-DCM).
- Parallel implementation of the data-based estimation in a GPU.
- Use of principal component analysis (PCA) for dimensionality reduction of the datasets.

Optimal Filtering for State Estimation of Unknown Systems

Discrete-time nonlinear system

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

Causal estimator for state variable $v_k = x_{i,k}, i \in [1, \ldots, n_x]$:

$$
\begin{aligned}
\hat{v}_k = f(&\tilde{\mathbf{d}}_k, \tilde{\mathbf{y}}_k, \tilde{\mathbf{u}}_k, \tilde{\mathbf{d}}_{k-1}, \tilde{\mathbf{y}}_{k-1}, \tilde{\mathbf{u}}_{k-1}, \\
&\ldots, \tilde{\mathbf{d}}_{k-m+1}, \tilde{\mathbf{y}}_{k-m+1}, \tilde{\mathbf{u}}_{k-m+1}).
\end{aligned}
$$

Optimal Filtering for State Estimation of Unknown Systems

Discrete-time nonlinear system

$$\mathbf{x}_{k+1} = \mathbf{F}(\mathbf{x}_k, \mathbf{u}_k, \mathbf{d}_k, w_k)$$
$$\mathbf{y}_k = \mathbf{G}(\mathbf{x}_k, \mathbf{u}_k, \mathbf{d}_k, w_k)$$

Causal estimator for state variable $v_k = x_{i,k}, i \in [1, \dots, n_x]$:

$$\hat{v}_k = f(\tilde{\mathbf{d}}_k, \tilde{\mathbf{y}}_k, \tilde{\mathbf{u}}_k, \tilde{\mathbf{d}}_{k-1}, \tilde{\mathbf{y}}_{k-1}, \tilde{\mathbf{u}}_{k-1}, \dots, \tilde{\mathbf{d}}_{k-m+1}, \tilde{\mathbf{y}}_{k-m+1}, \tilde{\mathbf{u}}_{k-m+1}).$$

- **Objective:** Obtain a causal filter with small estimation error $v_k - \hat{v}_k$.

Optimal Filtering for State Estimation of Unknown Systems

Discrete-time nonlinear system

$$\mathbf{x}_{k+1} = \mathbf{F}(\mathbf{x}_k, \mathbf{u}_k, \mathbf{d}_k, w_k)$$
$$\mathbf{y}_k = \mathbf{G}(\mathbf{x}_k, \mathbf{u}_k, \mathbf{d}_k, w_k)$$

Causal estimator for state variable $v_k = x_{i,k}, i \in [1, \ldots, n_x]$:

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

- **Objective:** Obtain a causal filter with small estimation error $v_k - \hat{v}_k$.
- **Assumptions:** $\{F, G\}$ unknown, system is n-step observable [2], noise bounded in l_p-norm.

Optimal Filtering for State Estimation of Unknown Systems

Discrete-time nonlinear system

$$\mathbf{x}_{k+1} = \mathbf{F}(\mathbf{x}_k, \mathbf{u}_k, \mathbf{d}_k, w_k)$$
$$\mathbf{y}_k = \mathbf{G}(\mathbf{x}_k, \mathbf{u}_k, \mathbf{d}_k, w_k)$$

Causal estimator for state variable $v_k = x_{i,k}, i \in [1, \ldots, n_x]$:

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

- **Objective:** Obtain a causal filter with small estimation error $v_k - \hat{v}_k$.
- **Assumptions:** $\{F, G\}$ unknown, system is n-step observable [2], noise bounded in l_p-norm.
- **Approach:** Set Membership framework for system identification [4].

Agenda

Introduction

Problem Statement

Optimal Filtering for State Estimation of Unknown Systems

Results

Conclusion - Perspectives

Direct Filtering - Set Membership Framework [4]

Assume dataset of measurements $\mathcal{D} = \{(\tilde{\varphi}_i, \tilde{v}_i), \quad i = 1, 2, \ldots, N\}$ is available.
Prior assumptions on f_0:

$$f_0 \in \mathcal{F}(\gamma) := \left\{ f \in C^1 : \left\| f'(\varphi) \right\| \leq \gamma, \forall \varphi \in \mathbb{R}^{(n_d+n_y+n_u)m} \right\}$$

Prior assumptions on noise:

$$W \in \mathcal{W} = \{[w_1, w_2, \ldots, w_T] : |w_k| \leq \varepsilon, \forall k 1, 2, \ldots, T\}$$

Define the feasible filter set (FFS):

$$FFS \doteq \{f \in \mathcal{F}(\gamma) : |\tilde{v}_i - f(\tilde{\varphi}_i)| \leq \varepsilon, \quad i = 1, \ldots, N\}$$

Direct Filtering - Set Membership Framework [4]

Theorem

1. *A necessary condition for the FFS to be non-empty is:*

$$\overline{f}\left(\tilde{\varphi}_i\right) \geq \tilde{v}_i - \varepsilon, \quad i = 1, \ldots, N$$

2. *A sufficient condition for the FFS to be non-empty is:*

$$\overline{f}\left(\tilde{\varphi}_i\right) > \tilde{v}_i - \varepsilon, \quad i = 1, \ldots, N$$

Direct Filtering - Set Membership Framework

The worst-case bounds are defined as:

$$\overline{f}_c(\tilde{\varphi}_k) = \min_{i=1,\dots,N} \left(\tilde{v}_i + \varepsilon + \gamma \|\tilde{\varphi}_k - \tilde{\varphi}_i\| \right)$$

$$\underline{f}_c(\tilde{\varphi}_k) = \max_{i=1,\dots,N} \left(\tilde{v}_i - \varepsilon - \gamma \|\tilde{\varphi}_k - \tilde{\varphi}_i\| \right)$$

Direct Filtering - Set Membership Framework

The worst-case bounds are defined as:

$$\overline{f}_c(\tilde{\varphi}_k) = \min_{i=1,\dots,N} \left(\tilde{v}_i + \varepsilon + \gamma \left\|\tilde{\varphi}_k - \tilde{\varphi}_i\right\|\right)$$

$$\underline{f}_c(\tilde{\varphi}_k) = \max_{i=1,\dots,N} \left(\tilde{v}_i - \varepsilon - \gamma \left\|\tilde{\varphi}_k - \tilde{\varphi}_i\right\|\right)$$

Regressor error

Direct Filtering - Set Membership Framework

The worst-case bounds are defined as:

$$\overline{f}_c(\tilde{\varphi}_k) = \min_{i=1,\dots,N} \left(\tilde{v}_i + \varepsilon + \gamma \left\|\tilde{\varphi}_k - \tilde{\varphi}_i\right\|\right)$$

$$\underline{f}_c(\tilde{\varphi}_k) = \max_{i=1,\dots,N} \left(\tilde{v}_i - \varepsilon - \gamma \left\|\tilde{\varphi}_k - \tilde{\varphi}_i\right\|\right)$$

Regressor error

Maximum gradient

Direct Filtering - Set Membership Framework

The worst-case bounds are defined as:

$$\overline{f}_c(\tilde{\varphi}_k) = \min_{i=1,\ldots,N} \left(\tilde{v}_i + \varepsilon + \gamma \left\|\tilde{\varphi}_k - \tilde{\varphi}_i\right\|\right)$$

$$\underline{f}_c(\tilde{\varphi}_k) = \max_{i=1,\ldots,N} \left(\tilde{v}_i - \varepsilon - \gamma \left\|\tilde{\varphi}_k - \tilde{\varphi}_i\right\|\right)$$

Noise bounds

Regressor error

Maximum gradient

Direct Filtering - Set Membership Framework

The worst-case bounds are defined as:

$$\overline{f}_c(\tilde{\varphi}_k) = \min_{i=1,\dots,N} \left(\tilde{v}_i + \varepsilon + \gamma \left\|\tilde{\varphi}_k - \tilde{\varphi}_i\right\|\right)$$

$$\underline{f}_c(\tilde{\varphi}_k) = \max_{i=1,\dots,N} \left(\tilde{v}_i - \varepsilon - \gamma \left\|\tilde{\varphi}_k - \tilde{\varphi}_i\right\|\right)$$

Noise bounds

Regressor error

Maximum gradient

The **direct filter (DF)** is defined as:

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

Direct Filter Learning

Algorithm 1: Direct Filter learning for power converters (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 $\tilde{\mathbf{y}}(t)$, $\tilde{v}(t)$
3. $\bar{\mathbf{y}}(t) = \texttt{average}(\tilde{\mathbf{y}})$, $\bar{v}(t) = \texttt{average}(\tilde{v})$, $\texttt{average}(\cdot) :=$ non-causal filter
4. $\tilde{\mathbf{y}}_k = \texttt{resample}(\bar{\mathbf{y}}(t), T_s)$, $\tilde{v}_k = \texttt{resample}(\bar{v}(t), T_s)$, $\tilde{\mathbf{d}}_k = \texttt{resample}(\tilde{\mathbf{d}}(t), T_s)$, $\tilde{\mathbf{u}}_k = \texttt{resample}(\tilde{\mathbf{u}}(t), T_s)$,
5. Prepare dataset $\mathcal{D} = \{\tilde{\varphi}_i, \tilde{v}_i, i = 1, 2, \ldots, N\}$ using $\tilde{\mathbf{d}}_k$, $\tilde{\mathbf{y}}_k$, $\tilde{\mathbf{u}}_k$, $\tilde{v}_k$
6. Take $\varepsilon = \|\tilde{v}(t) - \bar{v}(t)\|_\infty$
7. Take $\gamma^* = \min \gamma$, subject to $\overline{f}(\tilde{\varphi}_i) > \tilde{v}_i - \varepsilon$, $i = 1, \ldots, N$ (sufficient condition in Theorem 1 [5])

Direct Filter Execution

Algorithm 2: Direct filtering estimation for power converters (online)

Data: $\mathcal{D}$, γ, ε

Result: $\hat{v}_k$

while *true* **do**

1. Measure $\tilde{d}_k$, $\tilde{y}_k$, $\tilde{u}_k$
2. $\tilde{\varphi}_k = [\tilde{\mathbf{d}}_k^m; \tilde{\mathbf{y}}_k^m; \tilde{\mathbf{u}}_k^m]$
3. $\overline{f}(\tilde{\varphi}_k) = \min_{i=1,\dots,N} \left(\tilde{v}_i + \varepsilon + \gamma \left\|\tilde{\varphi}_k - \tilde{\varphi}_i\right\|\right)$
   $\underline{f}(\tilde{\varphi}_k) = \max_{i=1,\dots,N} \left(\tilde{v}_i - \varepsilon - \gamma \left\|\tilde{\varphi}_k - \tilde{\varphi}_i\right\|\right)$
4. $\hat{v}_k = f_c\left(\tilde{\varphi}_k\right) = \frac{1}{2}\left[\overline{f}\left(\tilde{\varphi}_k\right) + \underline{f}\left(\tilde{\varphi}_k\right)\right]$

end

Direct Filtering - Advantages of the Approach

- Does not require an explicit system model.

Direct Filtering - Advantages of the Approach

- Does not require an explicit system model.
- Able to represent the dynamics in the complete operating range (CCM/DCM).

Direct Filtering - Advantages of the Approach

- Does not require an explicit system model.
- Able to represent the dynamics in the complete operating range (CCM/DCM).
- Provides a measure of uncertainty of the estimation.

Direct Filtering - Advantages of the Approach

- Does not require an explicit system model.
- Able to represent the dynamics in the complete operating range (CCM/DCM).
- Provides a measure of uncertainty of the estimation.
- Finite Impulse Response (FIR) structure: BIBO stable.

Real-time Implementation: Parallel Programming

- Distance computation $\|\tilde{\varphi}_k - \tilde{\varphi}_i\|$ in optimal tightest bounds $\rightarrow$ Expensive.

Real-time Implementation: Parallel Programming

- Distance computation $\|\tilde{\varphi}_k - \tilde{\varphi}_i\|$ in optimal tightest bounds $\rightarrow$ Expensive.
- **State-of-the-art:** Approximation of DF over a grid in regressor space [1] $\rightarrow$ Unfeasible for high-dimensional spaces.

Real-time Implementation: Parallel Programming

- Distance computation $\|\tilde{\varphi}_k - \tilde{\varphi}_i\|$ in optimal tightest bounds $\rightarrow$ Expensive.
- **State-of-the-art:** Approximation of DF over a grid in regressor space [1] $\rightarrow$ Unfeasible for high-dimensional spaces.
- Each term $\|\tilde{\varphi}_k - \tilde{\varphi}_i\|$, $i = 1, \ldots, N$ can be computed independently $\rightarrow$ parallelizable.

Real-time Implementation: Parallel Programming

- Distance computation $\|\tilde{\varphi}_k - \tilde{\varphi}_i\|$ in optimal tightest bounds $\rightarrow$ Expensive.
- **State-of-the-art:** Approximation of DF over a grid in regressor space [1] $\rightarrow$ Unfeasible for high-dimensional spaces.
- Each term $\|\tilde{\varphi}_k - \tilde{\varphi}_i\|$, $i = 1, \ldots, N$ can be computed independently $\rightarrow$ parallelizable.
- **Contribution:** Parallel programming implementation using Nvidia CUDA for improving computation performance:

Real-time Implementation: Parallel Programming

- Distance computation $\|\tilde{\varphi}_k - \tilde{\varphi}_i\|$ in optimal tightest bounds $\rightarrow$ Expensive.
- **State-of-the-art:** Approximation of DF over a grid in regressor space [1] $\rightarrow$ Unfeasible for high-dimensional spaces.
- Each term $\|\tilde{\varphi}_k - \tilde{\varphi}_i\|$, $i = 1, \ldots, N$ can be computed independently $\rightarrow$ parallelizable.
- **Contribution:** Parallel programming implementation using Nvidia CUDA for improving computation performance:

Kernel 1 : $\psi_i^j = (\tilde{\varphi}_k^j - \tilde{\varphi}_i^j)^2,\ i = 1, \ldots, N\ \ j = 1, \ldots, 3m.$

Real-time Implementation: Parallel Programming

- Distance computation $\|\tilde{\varphi}_k - \tilde{\varphi}_i\|$ in optimal tightest bounds $\rightarrow$ Expensive.
- **State-of-the-art:** Approximation of DF over a grid in regressor space [1] $\rightarrow$ Unfeasible for high-dimensional spaces.
- Each term $\|\tilde{\varphi}_k - \tilde{\varphi}_i\|, i = 1, \ldots, N$ can be computed independently $\rightarrow$ parallelizable.
- **Contribution:** Parallel programming implementation using Nvidia CUDA for improving computation performance:

Kernel 1 : $\psi_i^j = (\tilde{\varphi}_k^j - \tilde{\varphi}_i^j)^2, \; i = 1, \ldots, N \;\; j = 1, \ldots, 3m.$

Kernel 2 : $\Delta_i = \sqrt{\sum_{j=1}^{3m} \psi_i^j}, \;\; \overline{f}_i = \tilde{v}_i + \varepsilon + \gamma\Delta_i, \;\; \underline{f}_i = \tilde{v}_i - \varepsilon - \gamma\Delta_i.$

Real-time Implementation: Parallel Programming

- Distance computation $\|\tilde{\varphi}_k - \tilde{\varphi}_i\|$ in optimal tightest bounds $\rightarrow$ Expensive.
- **State-of-the-art:** Approximation of DF over a grid in regressor space [1] $\rightarrow$ Unfeasible for high-dimensional spaces.
- Each term $\|\tilde{\varphi}_k - \tilde{\varphi}_i\|, i = 1, \ldots, N$ can be computed independently $\rightarrow$ parallelizable.
- **Contribution:** Parallel programming implementation using Nvidia CUDA for improving computation performance:

Kernel 1 : $\psi_i^j = (\tilde{\varphi}_k^j - \tilde{\varphi}_i^j)^2,\ i = 1, \ldots, N\ \ j = 1, \ldots, 3m.$

Kernel 2 : $\Delta_i = \sqrt{\sum_{j=1}^{3m} \psi_i^j},\ \ \overline{f}_i = \tilde{v}_i + \varepsilon + \gamma\Delta_i,\ \ \underline{f}_i = \tilde{v}_i - \varepsilon - \gamma\Delta_i.$

Kernel 3 : $\overline{f} = \min_{i=1,\ldots,N}(\overline{f}_i),\ \ \underline{f} = \max_{i=1,\ldots,N}(\underline{f}_i).$

- **Contribution:** Principal component analysis (PCA) on regressor dataset to improve computation performance.

Agenda

Introduction

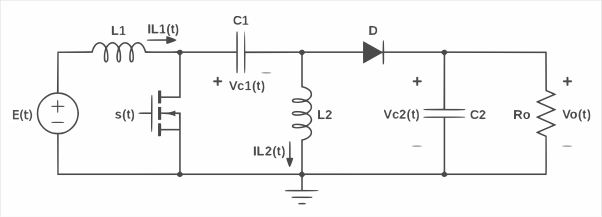
Problem Statement

Optimal Filtering for State Estimation of Unknown Systems

Results

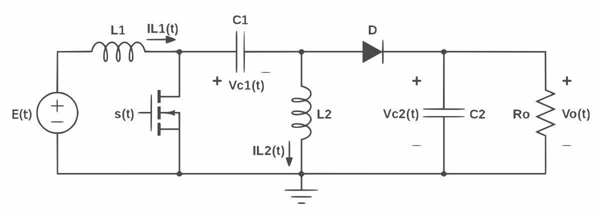
Conclusion - Perspectives

Example: SEPIC Converter



Observation problem: Compute estimates of current $I_{L1}(t)$ from measurements of input voltage $E(t)$, output voltage $V_o(t)$ and duty cycle $d(t)$ in PWM input $s(t)$ for the SEPIC converter operating in both CCM and DCM.

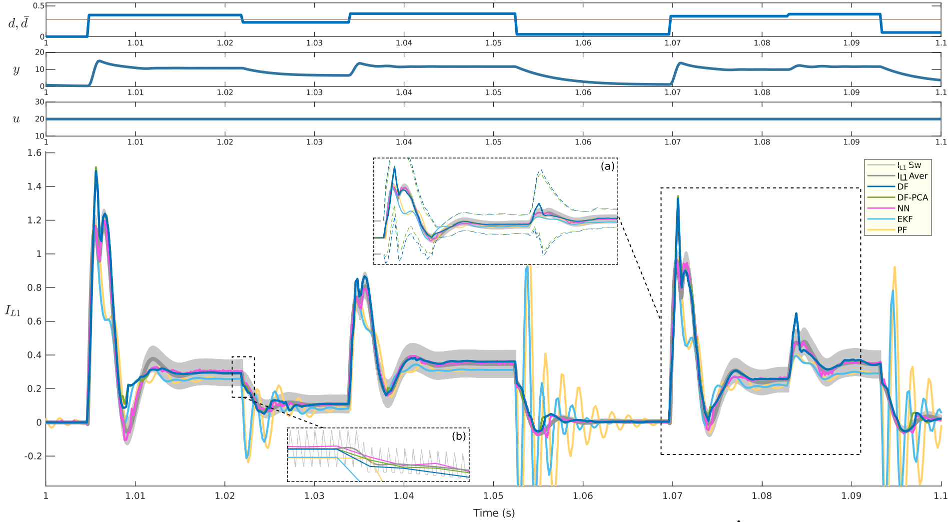
Example: SEPIC Converter



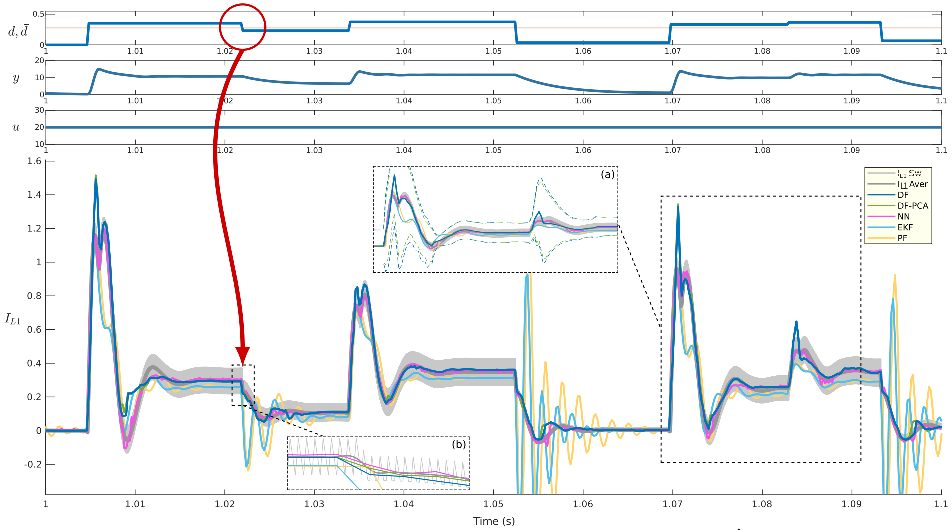
Observation problem: Compute estimates of current $I_{L1}(t)$ from measurements of input voltage $E(t)$, output voltage $V_o(t)$ and duty cycle $d(t)$ in PWM input $s(t)$ for the SEPIC converter operating in both CCM and DCM.

Comparison of direct filter (DF), direct filter with PCA-reduced dataset (DF+PCA), neural networks (NN), extended Kalman Filter (EKF) and particle filter (PF).

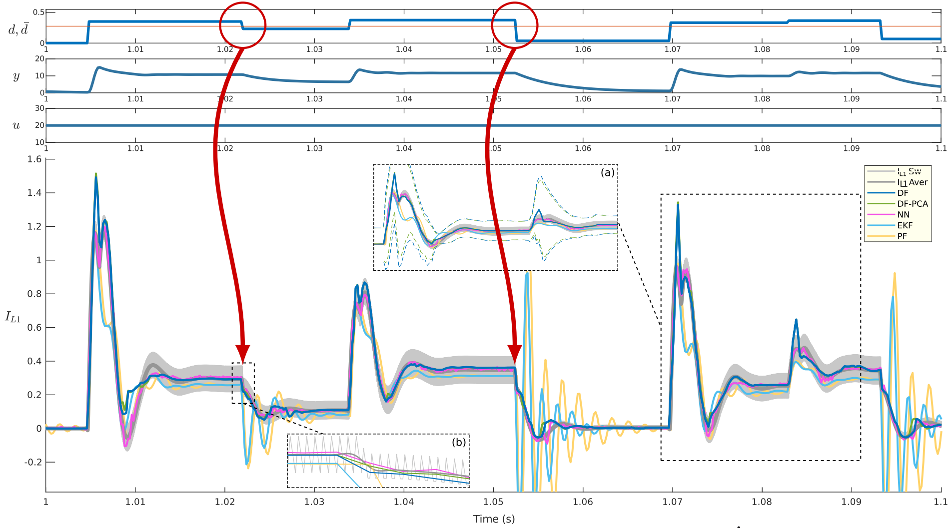
Example: SEPIC Converter - Simulation Results



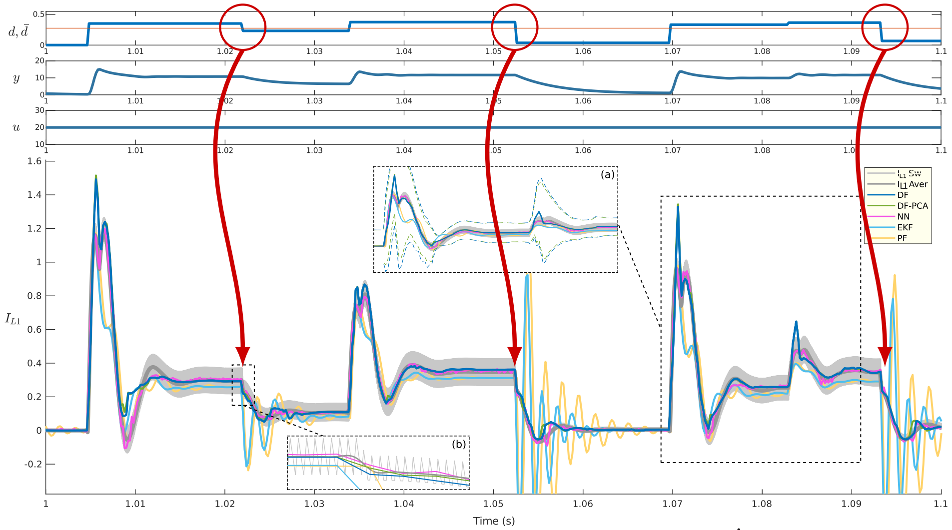
Example: SEPIC Converter - Simulation Results



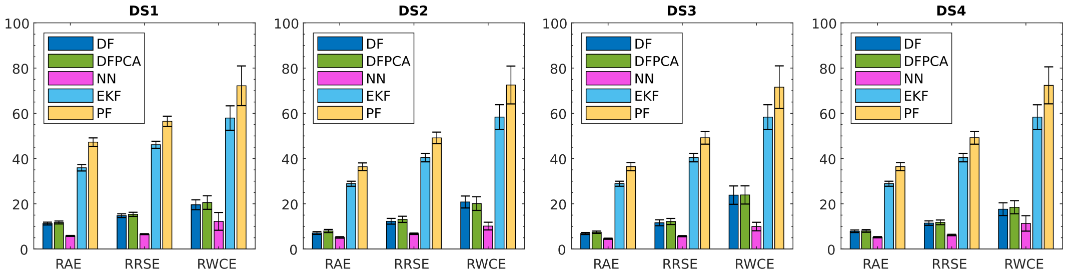
Example: SEPIC Converter - Simulation Results



Example: SEPIC Converter - Simulation Results



Example: SEPIC Converter - Simulation Results



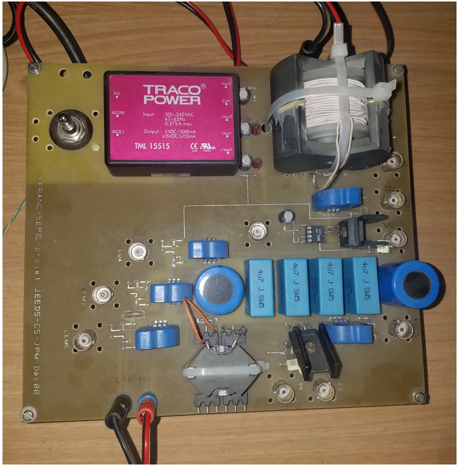
Performance measures:

Relative absolute error: $\text{RAE} = 100\, \|v - \hat{v}\|_1 \,/\, \|v - \bar{v}\|_1$

Root relative square error: $\text{RRSE} = 100\, \|v - \hat{v}\|_2 \,/\, \|v - \bar{v}\|_2$

Worst-case error: $\text{RWCE} = 100\, \|v - \hat{v}\|_\infty \,/\, \|v - \bar{v}\|_\infty$

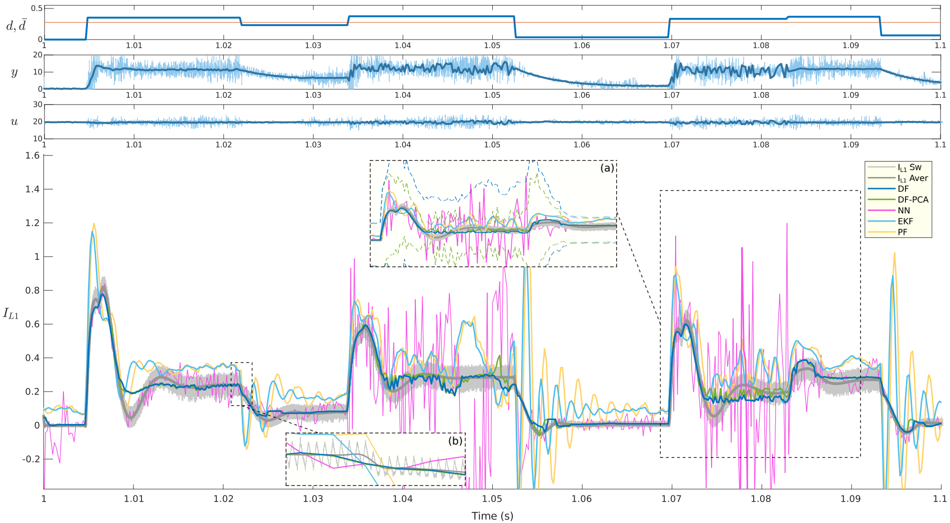
Example: SEPIC Converter - Experimental Results



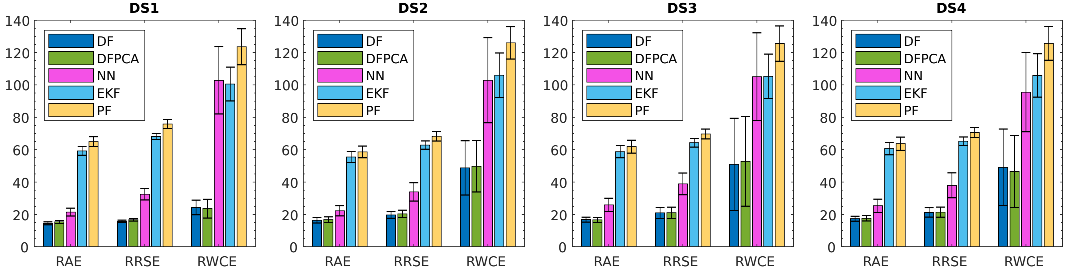
Converter	Parameters
SEPIC	$u(t) = 20$ V, $L_1 = 2.3$ mH, $C_1 = 190$ μF, $L_2 = 330$ μH, $C_2 = 190$ μF, $R_{L1} = 2.134$ Ω, $R_{L2} = 0.234$ Ω, $R_o = 22$ Ω.

SEPIC converter test bench available at Laboratoire Ampère

Example: SEPIC Converter - Experimental Results



Example: SEPIC Converter - Experimental Results



Performance measures:

Relative absolute error: $\mathrm{RAE} = 100 \, \|v - \hat{v}\|_1 \, / \, \|v - \bar{v}\|_1$

Root relative square error: $\mathrm{RRSE} = 100 \, \|v - \hat{v}\|_2 \, / \, \|v - \bar{v}\|_2$

Worst-case error: $\mathrm{RWCE} = 100 \, \|v - \hat{v}\|_\infty \, / \, \|v - \bar{v}\|_\infty$

Performance: CPU vs GPU vs GPU+PCA

Dataset	Mean performance loss (%)			Execution time (ms) / speedup w.r.t CPU		
	RAE	RRSE	RWCE	CPU	GPU	GPU+PCA
DS1	1.5446	1.6832	3.0170	1.6410	1.0141 / 1.6183X	0.2419 / 6.7847X
DS2	0.2641	0.8726	4.0489	1.6445	1.0157 / 1.6191X	0.2434 / 6.7577X
DS3	0.6703	0.1592	6.4868	1.6491	1.0172 / 1.6212X	0.2437 / 6.7660X
DS4	0.4993	0.7582	1.9521	1.6428	1.0176 / 1.6144X	0.2443 / 6.7235X

Performance: CPU vs GPU vs GPU+PCA

Dataset	Mean performance loss (%)			Execution time (ms) / speedup w.r.t CPU		
	RAE	RRSE	RWCE	CPU	GPU	GPU+PCA
DS1	1.5446	1.6832	3.0170	1.6410	1.0141 / 1.6183X	0.2419 / 6.7847X
DS2	0.2641	0.8726	4.0489	1.6445	1.0157 / 1.6191X	0.2434 / 6.7577X
DS3	0.6703	0.1592	6.4868	1.6491	1.0172 / 1.6212X	0.2437 / 6.7660X
DS4	0.4993	0.7582	1.9521	1.6428	1.0176 / 1.6144X	0.2443 / 6.7235X

Performance loss is small compared to gains in computation speed!

Agenda

Introduction

Problem Statement

Optimal Filtering for State Estimation of Unknown Systems

Results

Conclusion - Perspectives

Conclusion

With respect to the state-of-the-art, we have introduced:

- A practical approach to direct filtering in power converters using parallel programming and data compression.

Conclusion

With respect to the state-of-the-art, we have introduced:

- A practical approach to direct filtering in power converters using parallel programming and data compression.
- An estimation approach that works on a wide operation range, without requiring a complex system model.

Perspectives

- Implementation in an RTOS platform where GPU kernel execution can satisfy deterministic constraints.

Perspectives

- Implementation in an RTOS platform where GPU kernel execution can satisfy deterministic constraints.
- Investigate dependence of estimation performance on parameters m, N for different converter topologies.

Perspectives

- Implementation in an RTOS platform where GPU kernel execution can satisfy deterministic constraints.
- Investigate dependence of estimation performance on parameters m, N for different converter topologies.
- Investigate relation between performance loss and dimension of PCA transformation.

Perspectives

- Implementation in an RTOS platform where GPU kernel execution can satisfy deterministic constraints.
- Investigate dependence of estimation performance on parameters m, N for different converter topologies.
- Investigate relation between performance loss and dimension of PCA transformation.
- Investigate other dimensionality reduction approaches (Kernel PCA, linear discriminant analysis, generalized discriminant analysis, auto-encoders).

Thank you for your attention.

gerardo.becerra@estia.fr

Direct Filter - Observability [2]

Consider the following discrete-time linear system. Assume it is n-step observable.

$$x^{t+1} = A(\tilde{d}^t)x^t + B_u(\tilde{d}^t)\tilde{u}^t + B_w(\tilde{d}^t)w^t$$
$$\tilde{y}^t = C(\tilde{d}^t)x^t + D_u(\tilde{d}^t)\tilde{u}^t + D_w(\tilde{d}^t)w^t$$

Definition (n-step Observability)

$(A(\tilde{d}^t), C(\tilde{d}^t))$ is n-step observable if, for any time t and any sequence $\tilde{d}^t$, the state x^t can be uniquely determined by the corresponding zero-input response y^t for $k = t, \ldots, t+n-1$.

n-step observability matrix of the system:

$$\mathcal{O}_k^n = \begin{bmatrix} C^{t+n-1}\Phi^{t+n-1,t} \\ \vdots \\ C^{t+1}\Phi^{t+1,t} \\ C^t \end{bmatrix}$$

Transition matrix of the system:

$$\Phi^{t_2,t_1} = \begin{cases} A^{t_2-1}A^{t_2-2}\ldots A^{t_1}, & t_2 > t_1 \\ I, & t_2 = t_1 \end{cases}$$

Direct Filter - Observability [2]

- Since the system is assumed to be n-step observable, it follows that $rank(\mathcal{O}_n^t) = n$. Therefore, the inverse of $\mathcal{O}_n^t$ exists.
- In practice: Run the estimator assumming $(A(\tilde{d}^t), C(\tilde{d}^t))$ are known, and find minimum n such that $rank(\mathcal{O}_n^t) = n$.
- SEPIC converter in CCM: n = 20.

Direct Filter - Observability [2]

Evolution of $\tilde{y}$ from $t-n$ to $t-1$:

$$\mathbf{y}^{t-1,n} = \mathcal{O}_n^{t-n} x^{t-n} + \mathcal{T}_u^{t-n,n} \mathbf{u}^{t-1,n} + \mathcal{T}_w^{t-n,n} \mathbf{w}^{t-1,n}$$

$$\mathcal{T}_\alpha^{t-1,n} = \begin{bmatrix} D_\alpha^{t-1} & C^{t-1}\Phi^{t-1,t-1}B_\alpha^{t-2} & C^{t-1}\Phi^{t-1,t-2}B_\alpha^{t-3} & \dots & C^{t-1}\Phi^{t-1,t-n+1}B_\alpha^{t-n} \\ \mathbf{0} & D_\alpha^{t-2} & C^{t-2}\Phi^{t-2,t-2}B_\alpha^{t-3} & \dots & C^{t-2}\Phi^{t-2,t-n+1}B_\alpha^{t-n} \\ \mathbf{0} & \mathbf{0} & D_\alpha^{t-3} & \dots & C^{t-3}\Phi^{t-3,t-n+1}B_\alpha^{t-n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \ddots & D_\alpha^{t-n} \end{bmatrix}$$

References

[1] J.A. Castaño, F. Ruiz, and J. Régnier. "A Fast Approximation Algorithm for Set-Membership System Identification". In: *IFAC Proceedings Volumes* 44.1 (2011). 18th IFAC World Congress, pp. 4410–4415.

[2] C. Novara F. Ruiz and M. Milanese. "Direct design from data of optimal filters for LPV systems". In: *Systems and Control Letters* 59.1 (2010), pp. 1–8.

[3] I.T. Jolliffe. *Principal Component Analysis*. Springer Series in Statistics. Springer, 2002.

[4] M. Milanese and C. Novara. "Set Membership identification of nonlinear systems". In: *Automatica* 40.6 (2004), pp. 957–975.

[5] C. Novara, F. Ruiz, and M. Milanese. "Direct Filtering: A New Approach to Optimal Filter Design for Nonlinear Systems". In: *IEEE Transactions on Automatic Control* 58.1 (Jan. 2013), pp. 86–99.