2023 IEEE PES ISGT-Europe

October 23rd-26th, 2023, Grenoble, France

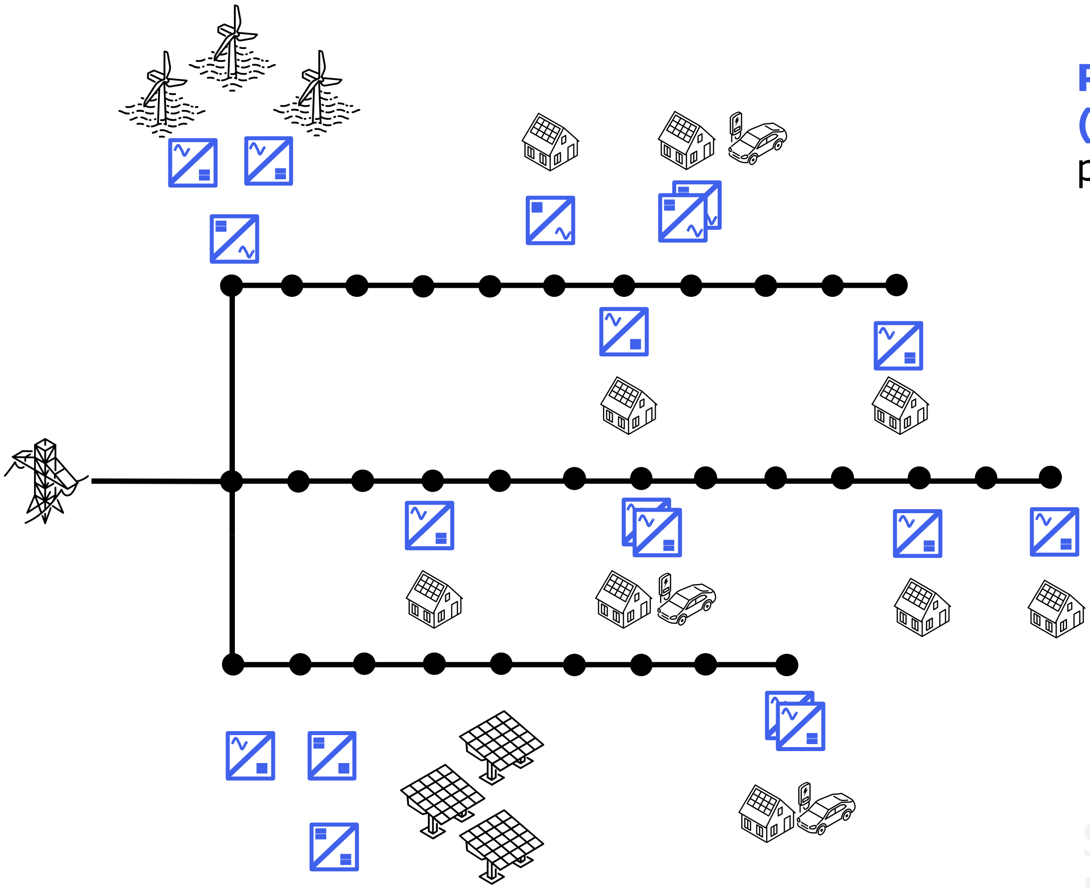
A C-HIL based data-driven DC-DC power electronics converter model for system-level studies

A. Colot, **B. Ewbank**, B. Cornélusse, M. Glavic

Montefiore Institute – University of Liège (BE)

Paper №: A8784BE

Introduction



Power Electronics Converters (PEC) are **ubiquitous** in modern power systems

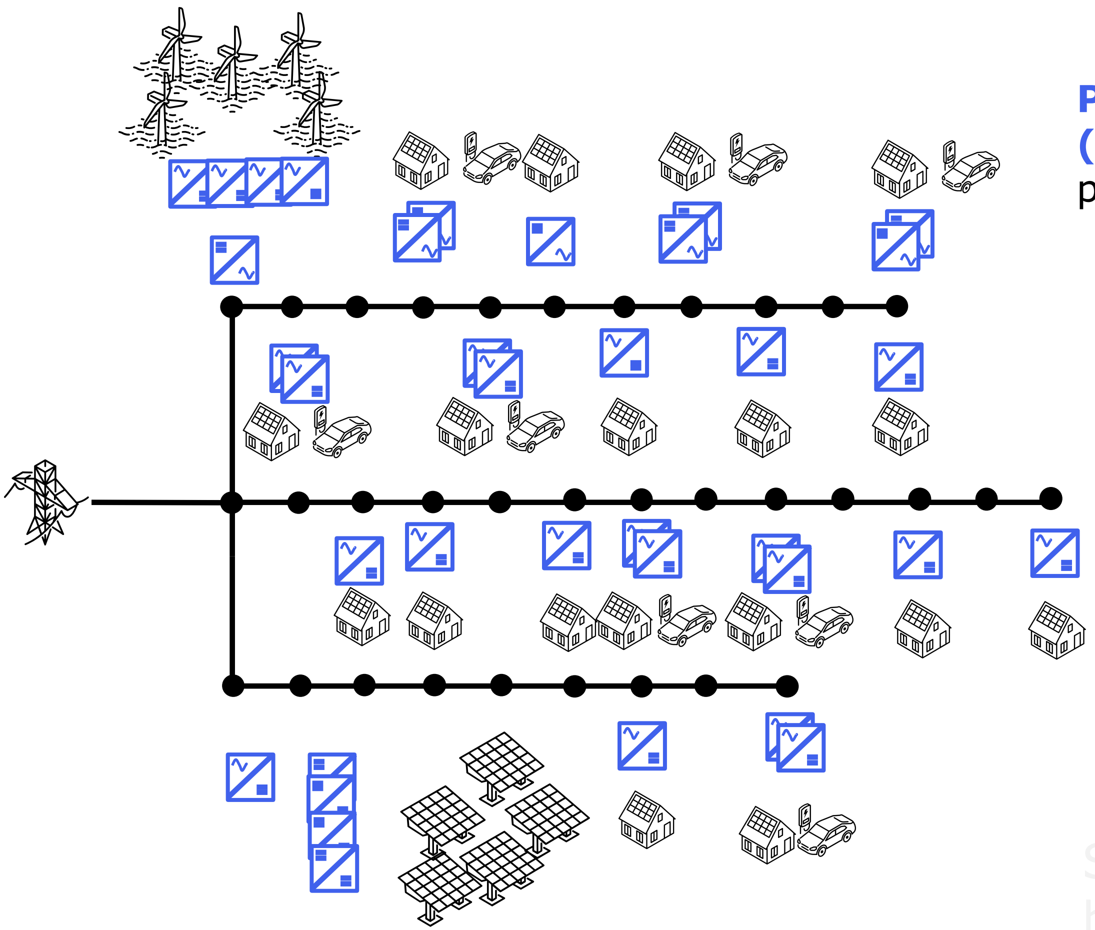
→ Need for **system-level studies**

Challenges :

- **Modeling,**
  PECs are Non-LTI system
- **Privacy,**
  Manufacturers don't want to disclose their secrets

System identification and black box modeling.

Introduction



Power Electronics Converters (PEC) are **ubiquitous** in modern power systems

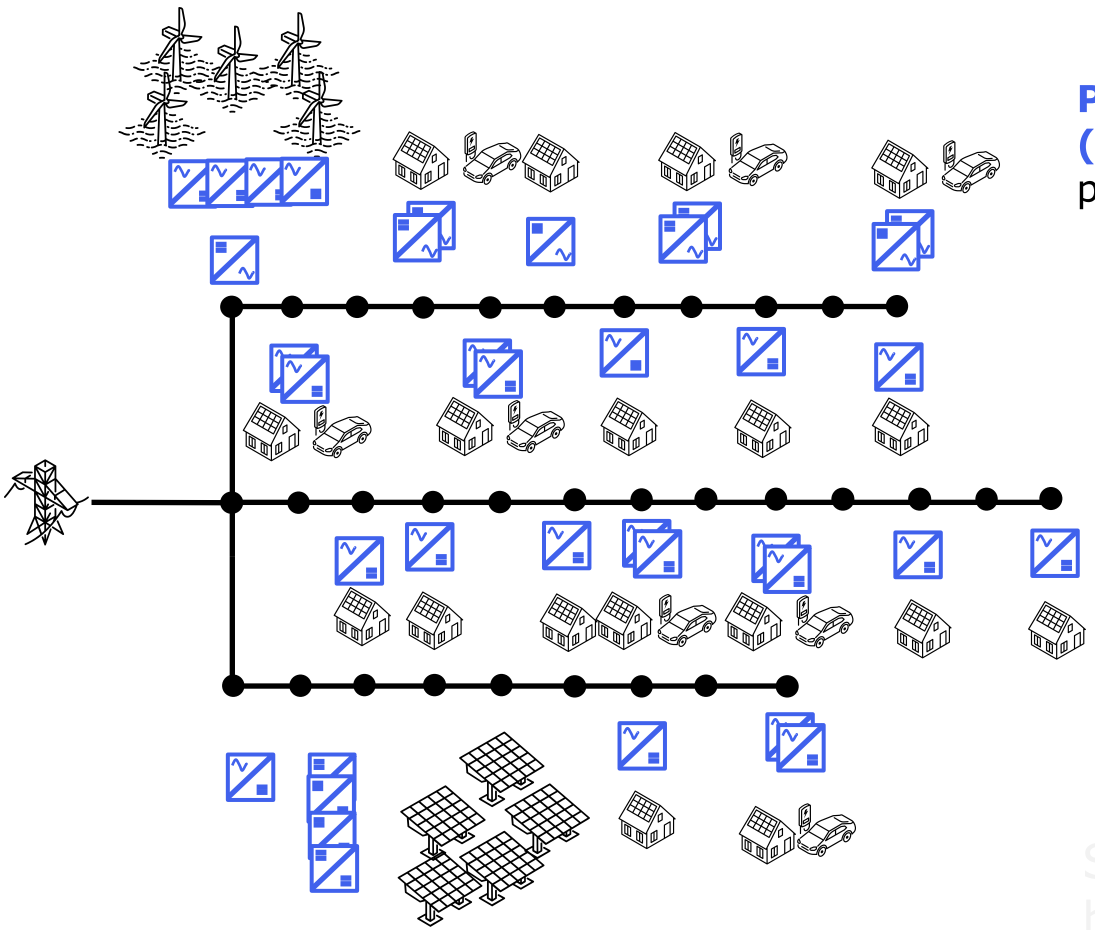
→ Need for **system-level studies**

Challenges :

- **Modeling,**
  PECs are Non-LTI system
- **Privacy,**
  Manufacturers don't want to disclose their secrets

System identification and black box modeling.

Introduction



Power Electronics Converters (PEC) are **ubiquitous** in modern power systems

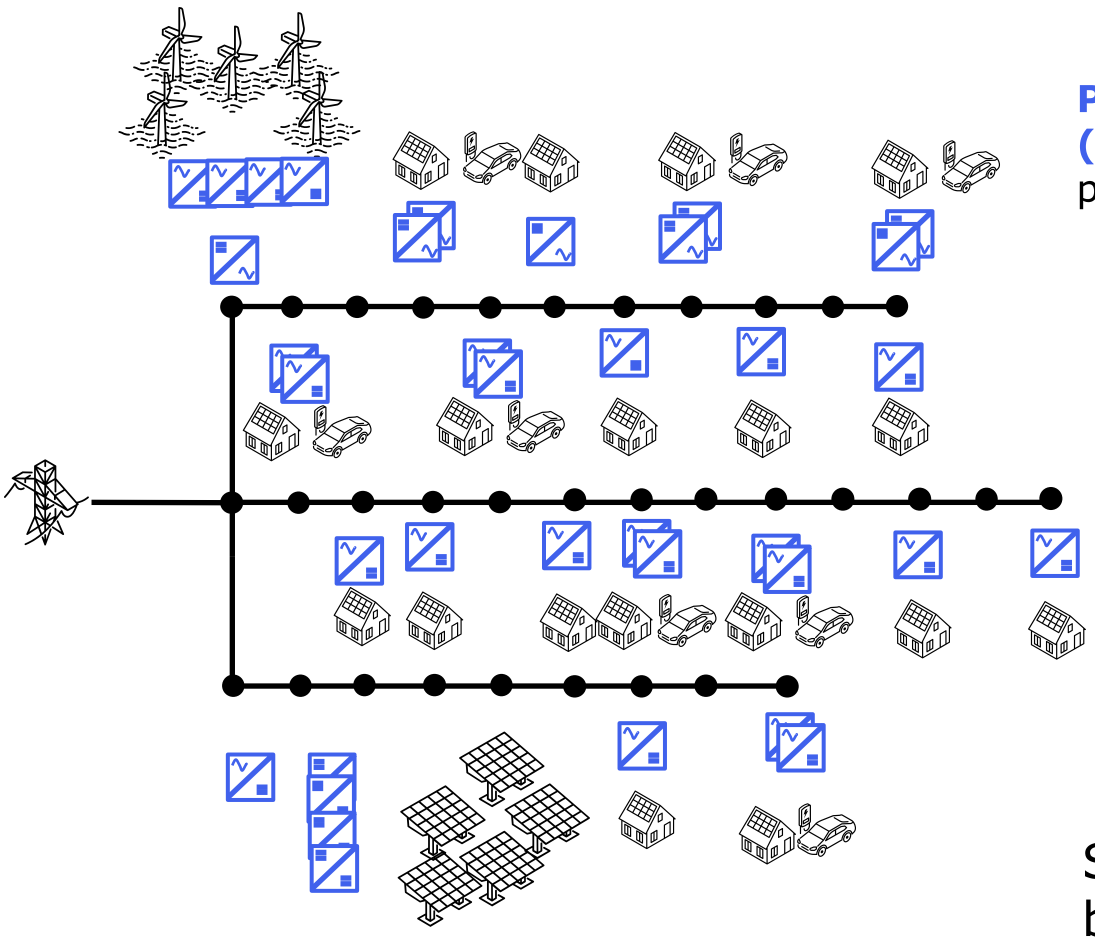
→ Need for **system-level studies**

Challenges :

- **Modeling,**
  PECs are Non-LTI system
- **Privacy,**
  Manufacturers do not want to disclose their secrets

System identification and black box modeling.

Introduction



Power Electronics Converters (PEC) are **ubiquitous** in modern power systems

→ Need for **system-level studies**

Challenges :

- **Modeling,**
  PECs are Non-LTI system
- **Privacy,**
  Manufacturers do not want to disclose their secrets

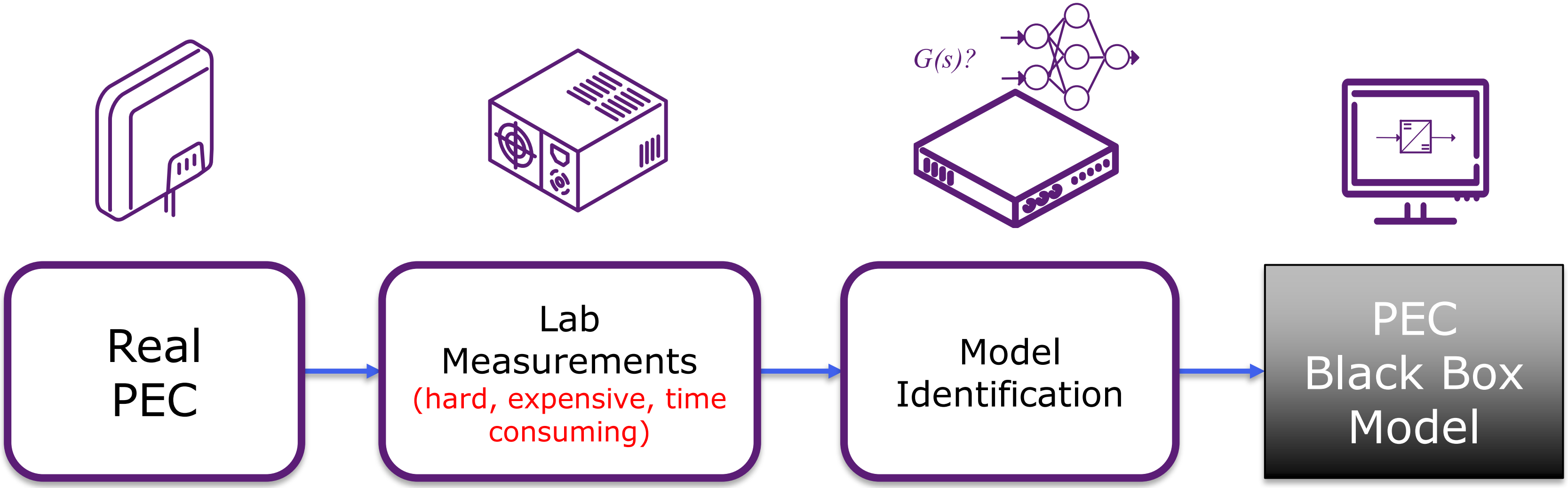
System identification and black box modeling

Black Box Model Generation

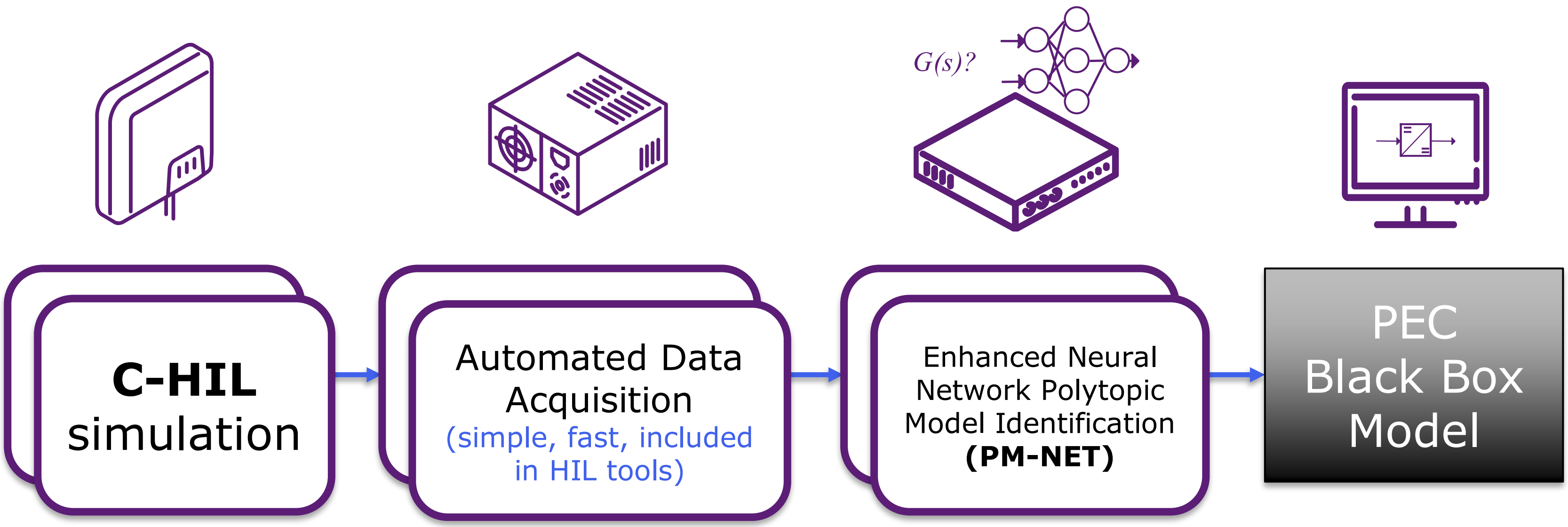
Real PEC

PEC Black Box Model

Black Box Model Generation



Black Box Model Generation



Polytopic models

Combination of N weighted linear models

Procedure :

1) Identify linear models around different operating points : $g_i(x)$

2) Design the input-depent weighting function : $\omega_i(x) \in [0\,;1]$

3) Output of polytopic model : $\hat{y} = \sum_{i=1}^{N} \omega_i(x)\, g_i(x)$

Main challenges : 1) and 2)

Which order and parameters of linear models ?

How to partition the operating space ?

How to design the weighting function ?

Proposed approach : **PM-NET**

Polytopic models

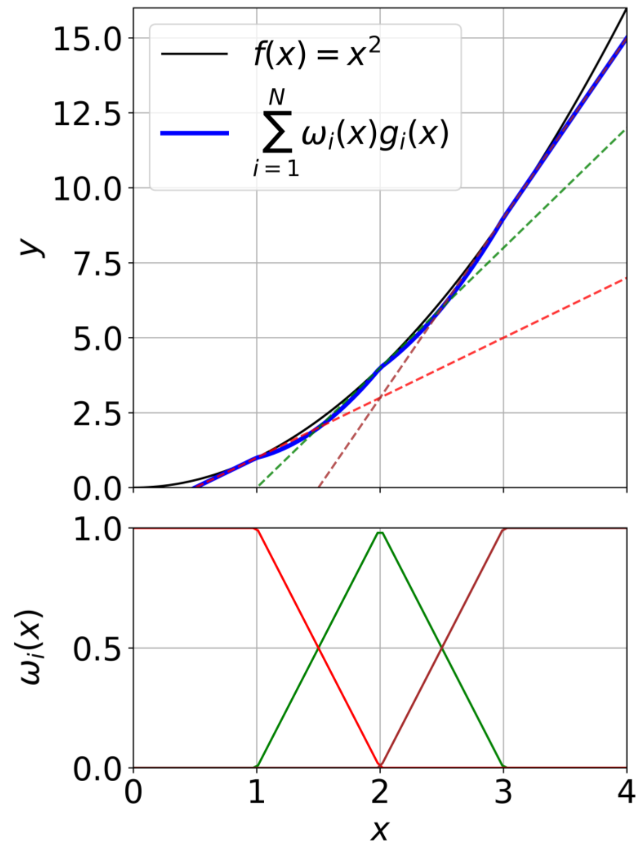
Combination of N weighted linear models

<u>Procedure:</u>

1) Identify linear models around different operating points: $g_i(x)$

2) Design the input-depent weighting function: $\omega_i(x) \in [0\,;1]$

3) Output of polytopic model: $\hat{y} = \sum_{i=1}^{N} \omega_i(x)\, g_i(x)$



Main challenges : 1) and 2)

Which order and parameters of linear models ?

How to partition the operating space ?

How to design the weighting function ?

Proposed approach : **PM-NET**

Polytopic models

Combination of N weighted linear models

Procedure:

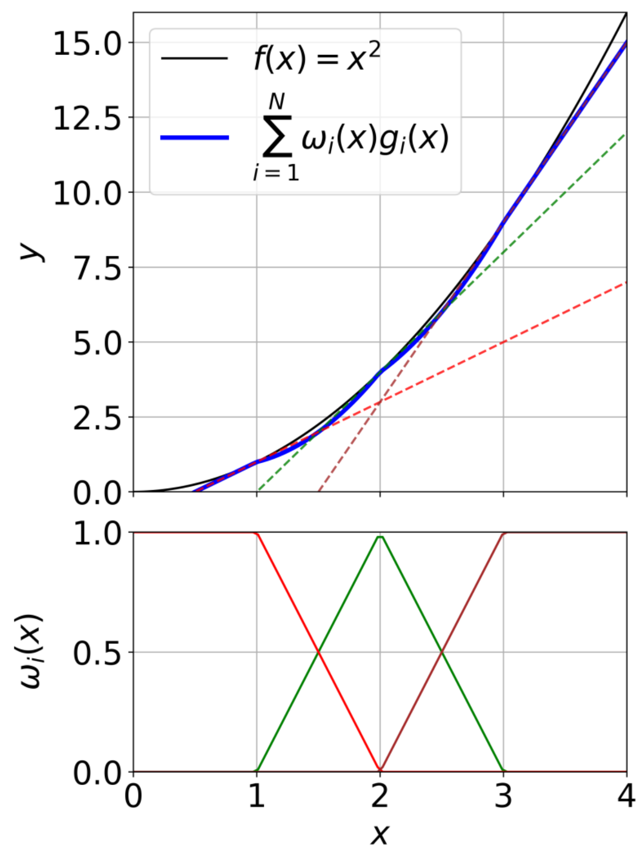
1) Identify linear models around different operating points: $g_i(x)$

2) Design the input-depent weighting function: $\omega_i(x) \in [0\,;1]$

3) Output of polytopic model: $\hat{y} = \sum_{i=1}^{N} \omega_i(x)\, g_i(x)$

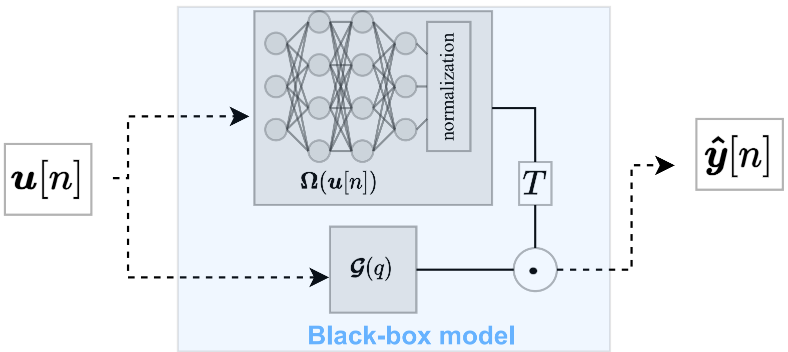
Main challenges: 1) and 2)

Which order and parameters of linear models ?
How to partition the operating space ?
How to design the weighting function ?



Proposed approach : **PM-NET**

PM-NET Identification



Init:

1) Data harvesting among the entire operating space

2) Orthotope-based partitioning along the input operating space

Iterative procedure :

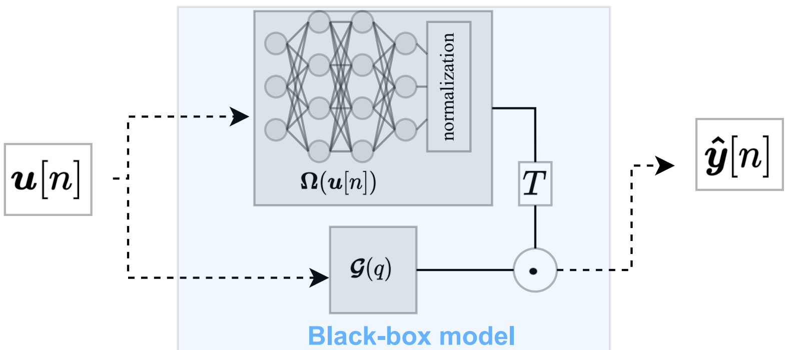
1) Data harvesting at selected operating points

2) $\boldsymbol{G}(q)$ LTI **models fitting**

3) Train MLP that yields the weighting function : $\boldsymbol{\Omega}(\mathbf{u}[\mathrm{n}])$

4) Analyze weights**, remove useless** $\boldsymbol{G}(\boldsymbol{q})$ and compute gradients of $\boldsymbol{\Omega}(\mathbf{u}[\mathrm{n}])$ for **selecting new op-point**

PM-NET Identification



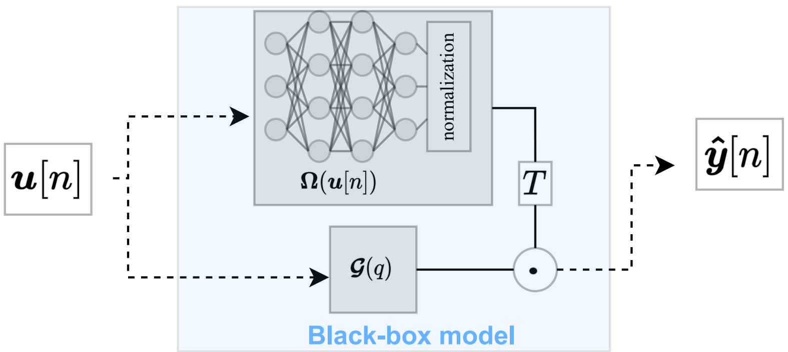
Init:

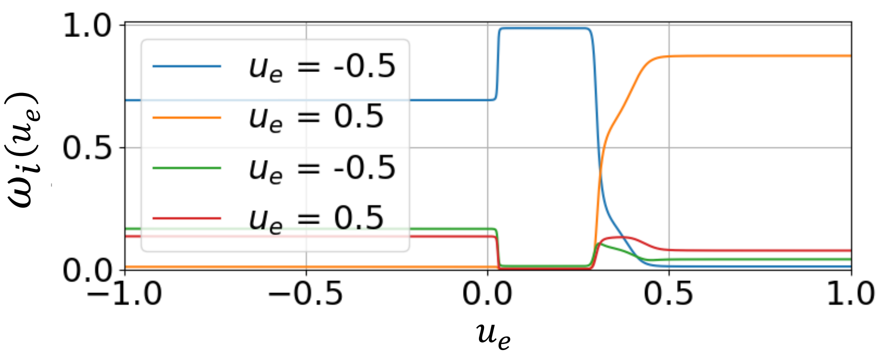
1) Data harvesting among the entire operating space
2) Orthotope-based partitioning along the input operating space

Iterative procedure:

1) Data harvesting at selected operating points
2) $\boldsymbol{G}(q)$ LTI **models fitting**
3) Train MLP that yields the weighting function : $\boldsymbol{\Omega}(\mathbf{u}[n])$
4) Analyze weights, **remove useless** $\boldsymbol{G}(q)$ and compute gradients of $\boldsymbol{\Omega}(\mathbf{u}[n])$ for **selecting new op-point**

PM-NET Identification





Init:

1) Data harvesting among the entire operating space
2) Orthotope-based partitioning along the input operating space

Iterative procedure:

1) Data harvesting at selected operating points
2) $\boldsymbol{G}(q)$ LTI **models fitting**
3) Train MLP that yields the weighting function : $\boldsymbol{\Omega}(\mathbf{u}[\mathrm{n}])$
4) Analyze weights**, remove useless $\boldsymbol{G}(\boldsymbol{q})$** and compute gradients of $\boldsymbol{\Omega}(\mathbf{u}[\mathrm{n}])$ for **selecting new op-point**

C-HIL simulation

Why C-HIL data are suitable for black box modeling:

C-HIL …

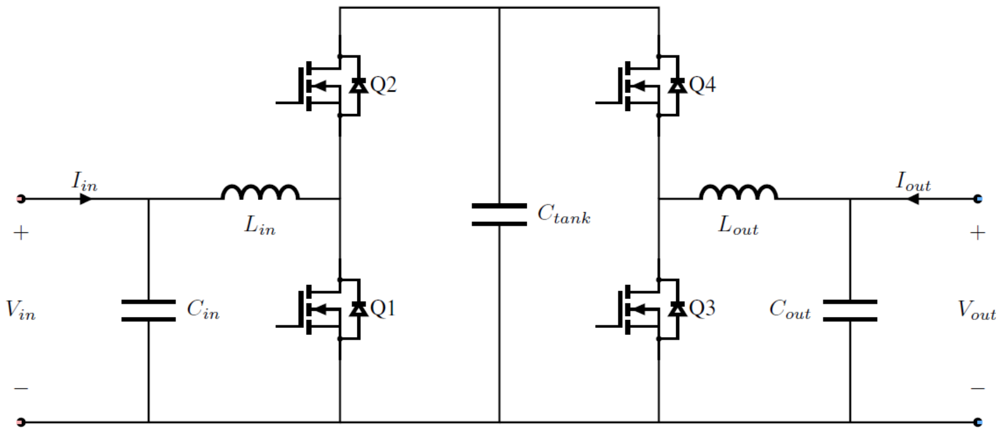
…models PECs with ideal switches and R,L,C components at ~1MHz bandwidth
→ **Accurate model for low frequencies**

…includes the true controller
→ **Closed loop system data**
(steady-state, start up, turn off, ramp…)

… is still partly a simulation
→ No technical constraint on **operating points browsing** and system perturbation

…can script the data acquisition
→ **Ease of use**

PEC under test



Bidirectionnal DC/DC droop controlled converter from an industrial partner

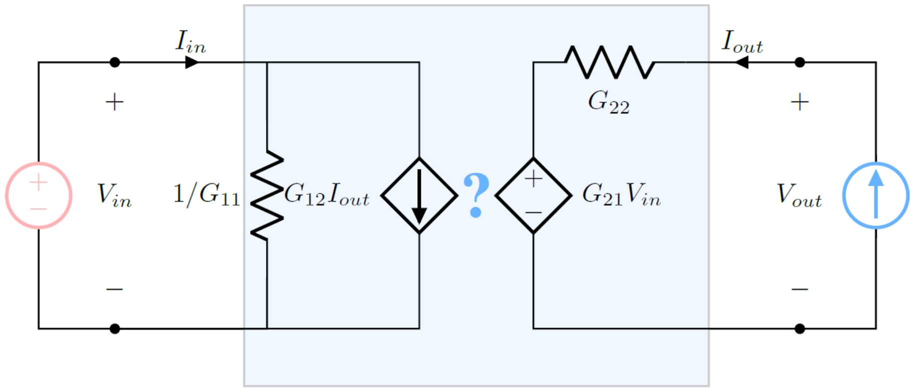
Specify I/O for model identification
(~Inverse hybrid parameters of a voltage amplifier)

Outputs **Inputs**

$$\begin{pmatrix} I_{in}[n] \\ V_{out}[n] \end{pmatrix} = \begin{pmatrix} G_{11}(q) & G_{12}(q) \\ G_{21}(q) & G_{22}(q) \end{pmatrix} \begin{pmatrix} V_{in}[n] \\ I_{out}[n] \end{pmatrix}$$

$$\boldsymbol{y}[n] = \boldsymbol{G}(q)\ \boldsymbol{u}[n]$$

PEC under test



Bidirectionnal DC/DC droop controlled converter from an industrial partner

Specify I/O for model identification
(~Inverse hybrid parameters of a voltage amplifier)

Outputs **Inputs**

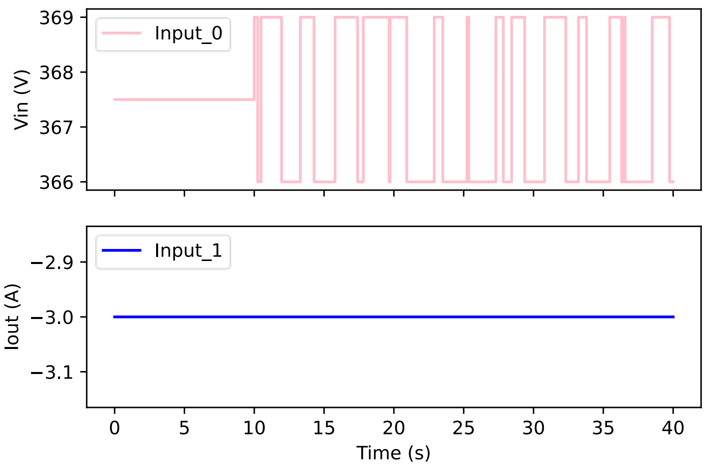
$$\begin{pmatrix} I_{in}[n] \\ V_{out}[n] \end{pmatrix} = \begin{pmatrix} G_{11}(q) & G_{12}(q) \\ G_{21}(q) & G_{22}(q) \end{pmatrix} \begin{pmatrix} V_{in}[n] \\ I_{out}[n] \end{pmatrix}$$

$$\boldsymbol{y}[n] = \boldsymbol{G}(q)\ \boldsymbol{u}[n]$$

C-HIL data acquisition

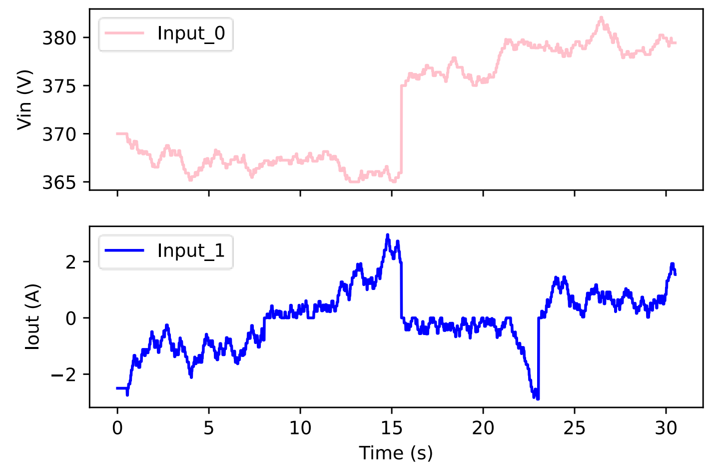
PRBS inputs

($\boldsymbol{G}(q)$ LTI models fitting)

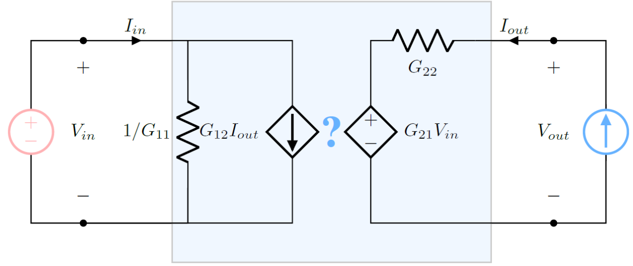


Browsing operating space

weighting function $\boldsymbol{\Omega}(\mathbf{u}[\mathrm{n}])$



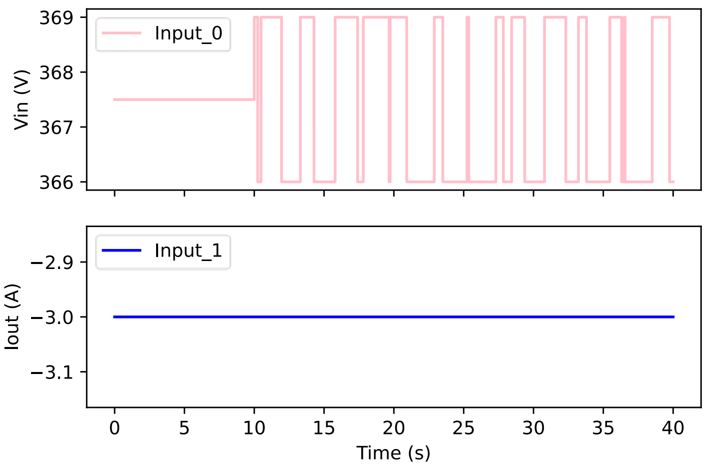
→ Fmax of PRBS:
100 Hz



C-HIL data acquisition

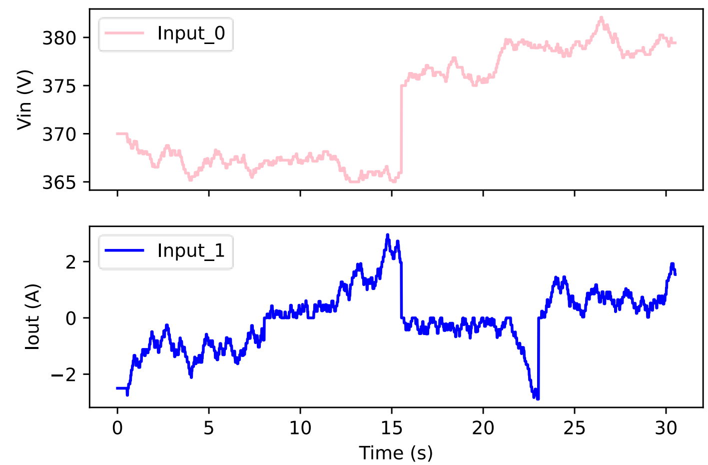
PRBS inputs

($\boldsymbol{G}(q)$ LTI models fitting)

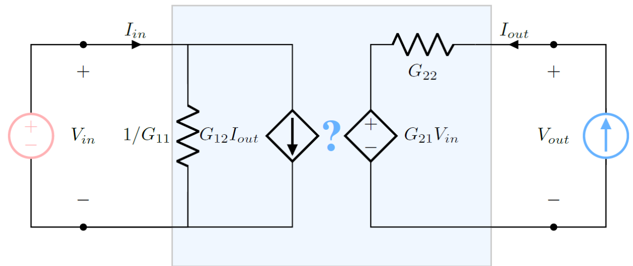


Browsing operating space

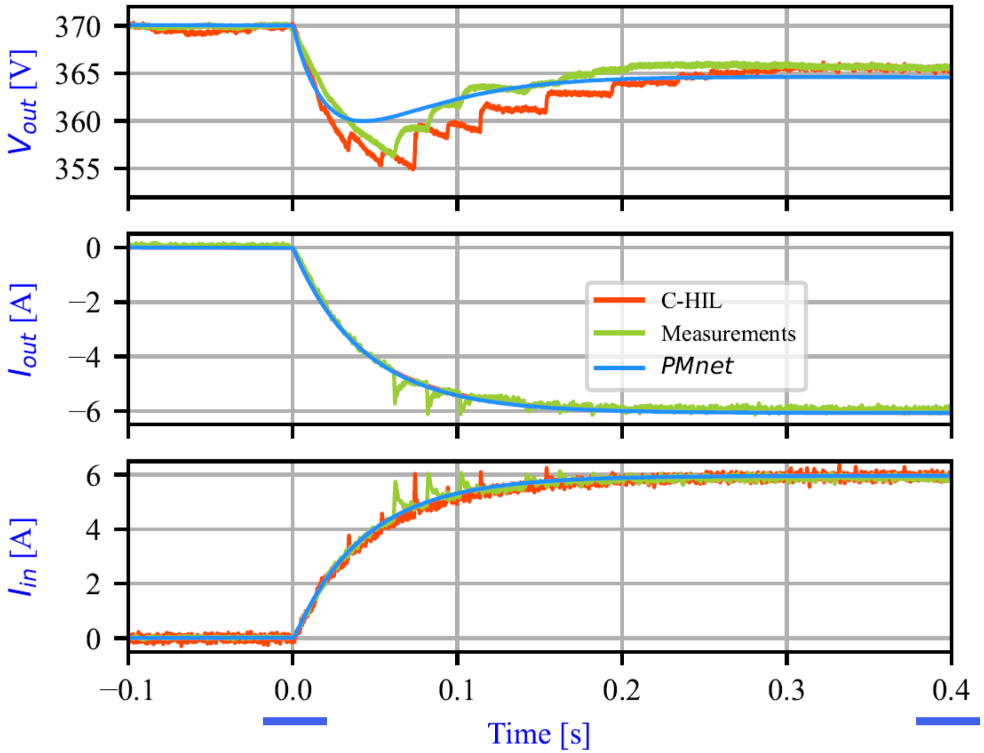
weighting function $\boldsymbol{\Omega}(\mathbf{u}[\mathrm{n}])$



→ Fmax of PRBS:
100 Hz



Results analysis : R-L load connection



Black box PM-net model fits well both real measurements and C-HIL simulation data.

- Steady state values v
- Lower voltage drop ~
- Slow dynamics v
- Fast transients x

Results analysis : small DC system

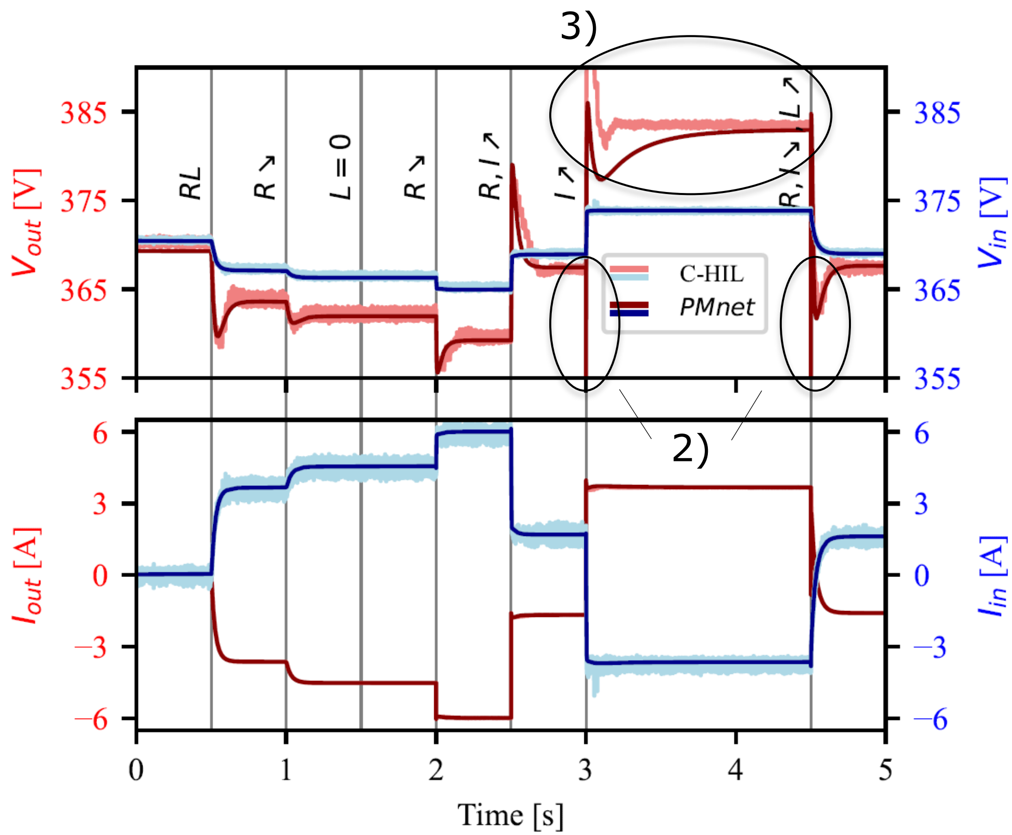
Testing single black box model with a battery and a variable load

Noticeable results :

1) **Model fits well while the battery was not present for the training**

2) An abrupt transition from one $G(q)$ to another can lead to an overshoot

3) Dead-band control operating points can be improved



Conclusion

Advantages of our approach :

- The PM-net black box models accuratly represent slow dynamics of PECs
- C-HIL simulation is viable for model identification

Further work :

- Applying state estimation for non-active LTI systems to prevent transients when sudden change in operating point
- Realize complete system-level studies

PM-NET operating space partion

Analyze if some weights are constantly below a threshold α:

- Remove the associated linear model and train again the neural network. *(pruning)*

- Compute numerically the gradients of the weighting function to identify along which direction to partition the operating space. *(segregation)*