



LABORATOIRE PLASMA ET  
CONVERSION D'ÉNERGIE

## IA pour le design en électronique de puissance

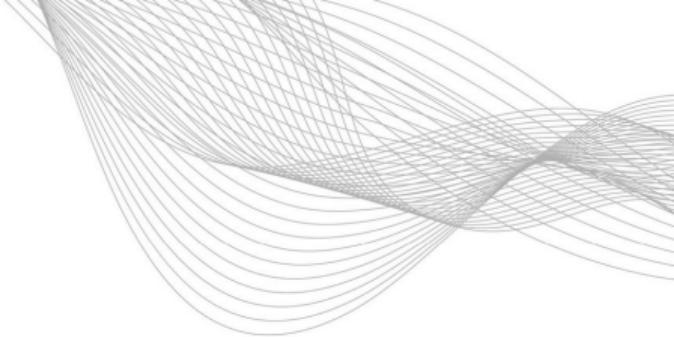
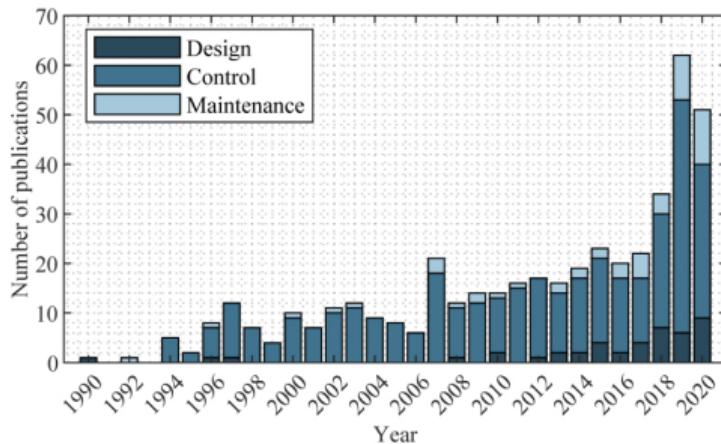
Journées de clôture PowerAlps

Pauline Kergus

10 Septembre 2025

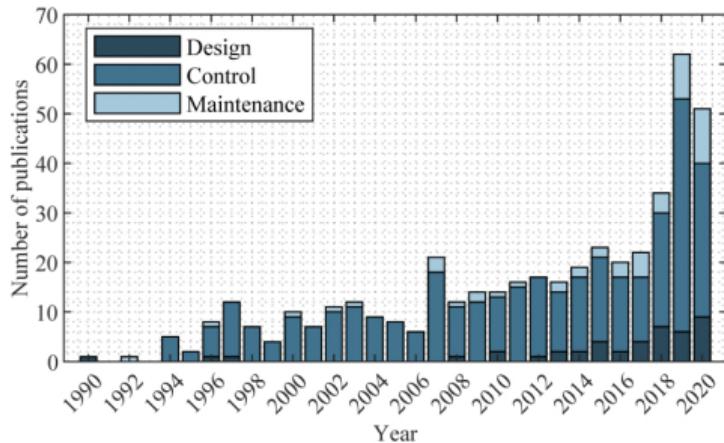


# Utilisation de l'IA en EP



An Overview of Artificial Intelligence Applications for Power Electronics, S. Zhao, F. Blaabjerg and H. Wang, *IEEE Transactions on Power Electronics*, 2021

# Utilisation de l'IA en EP

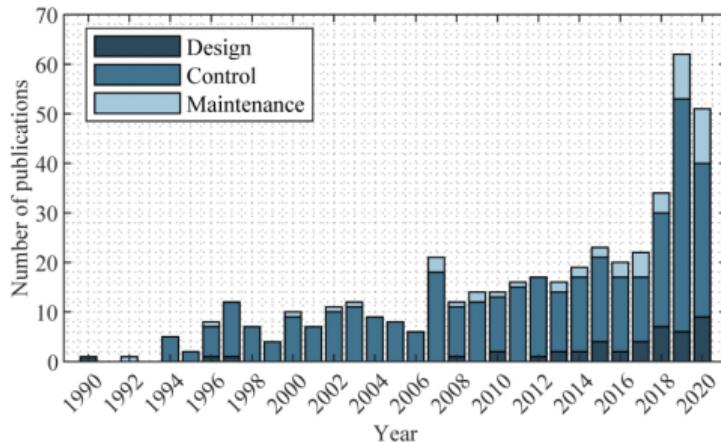


## Design

- ▶ Obj = Gain de temps
- ▶ Modèles de substitution
- ▶ Méta-heuristiques

*An Overview of Artificial Intelligence Applications for Power Electronics, S. Zhao, F. Blaabjerg and H. Wang, IEEE Transactions on Power Electronics, 2021*

# Utilisation de l'IA en EP



## Design

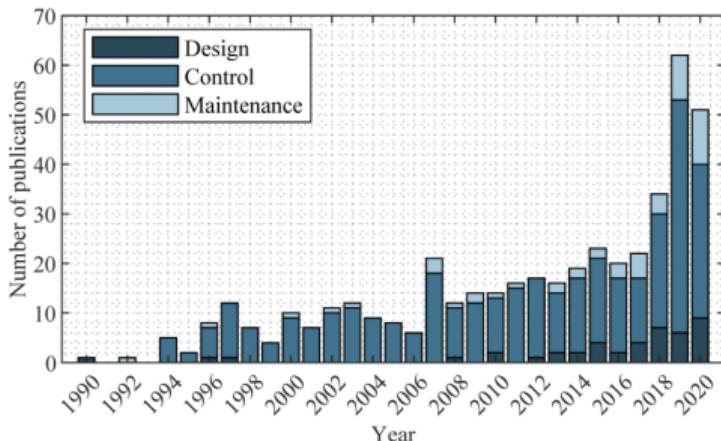
- ▶ Obj = Gain de temps
- ▶ Modèles de substitution
- ▶ Méta-heuristiques

## Commande

- ▶ Obj = Simplifier le design
- ▶ Logique floue
- ▶ Réseaux de neurones, RL

*An Overview of Artificial Intelligence Applications for Power Electronics, S. Zhao, F. Blaabjerg and H. Wang, IEEE Transactions on Power Electronics, 2021*

# Utilisation de l'IA en EP



## Design

- ▶ Obj = Gain de temps
- ▶ Modèles de substitution
- ▶ Méta-heuristiques

## Commande

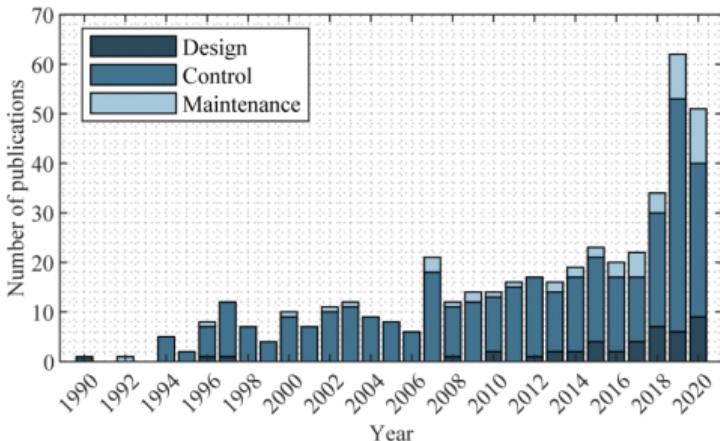
- ▶ Obj = Simplifier le design
- ▶ Logique floue
- ▶ Réseaux de neurones, RL

## Maintenance

- ▶ Obj = Modèles sur mesure
- ▶ Modèles (régression)
- ▶ Décision (classification)

An Overview of Artificial Intelligence Applications for Power Electronics, S. Zhao, F. Blaabjerg and H. Wang, *IEEE Transactions on Power Electronics*, 2021

# Utilisation de l'IA en EP



## Design

- ▶ Obj = Gain de temps
- ▶ Modèles de substitution
- ▶ Méta-heuristiques

## Commande

- ▶ Obj = Simplifier le design
- ▶ Logique floue
- ▶ Réseaux de neurones, RL

## Maintenance

- ▶ Obj = Modèles sur mesure
- ▶ Modèles (régression)
- ▶ Décision (classification)

TABLE VI  
REQUIREMENTS OF AI FOR EXEMPLARY APPLICATIONS IN DESIGN,  
CONTROL, AND MAINTENANCE

Requirements	Heatsink design (Design)	Intelligent controller (Control)	RUL prediction (Maintenance)
Computation Effort	+++	++	++
Algorithm Speed	+	+++	++
Accuracy	++	+++	+++
Dataset requirement	+	+	+++
Interpretability	+	+++	+++

High: +++, moderate: ++, low: +.

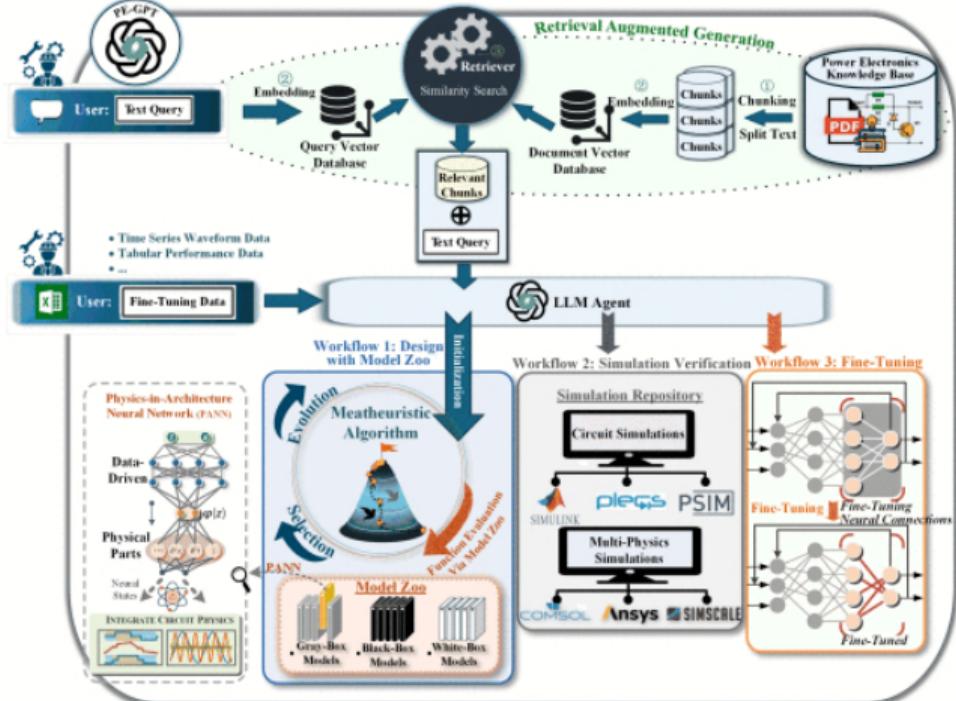
An Overview of Artificial Intelligence Applications for Power Electronics, S. Zhao, F. Blaabjerg and H. Wang, IEEE Transactions on Power Electronics, 2021

# Focus sur PE-GPT

Design automatisé de convertisseurs en 2 niveaux :

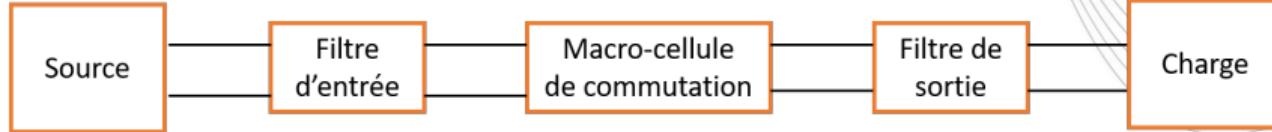
- Hétérogénéité des données et des tâches à effectuer
- Un LLM (GPT4) + RAG pour la formalisation du cahier des charges
- Des workflows complémentaires
  - Design
  - Simulation
  - Fine tuning à partir de données

PE-GPT: A New Paradigm for Power Electronics Design, F. Lin et al., IEEE Transactions on Industrial Electronics, April 2025



# Vers un design automatisé

## ► Calcul analytique



### Avantages :

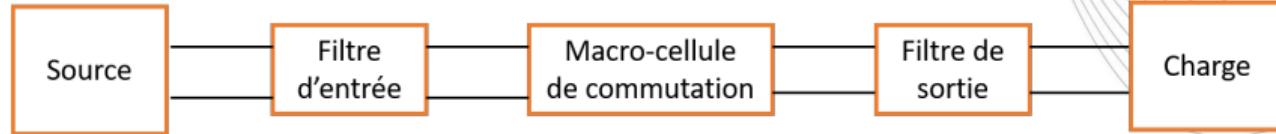
- Modularité / variété d'architectures
- Simulation incluse

### Limitations :

- En régime permanent et en BO
- Temps de calcul

# Vers un design automatisé

## ► Calcul analytique



### Avantages :

- Modularité / variété d'architectures
- Simulation incluse

### Limitations :

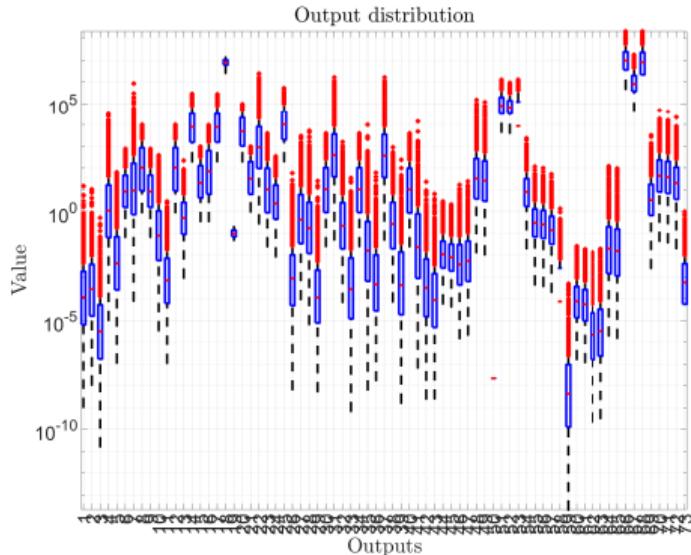
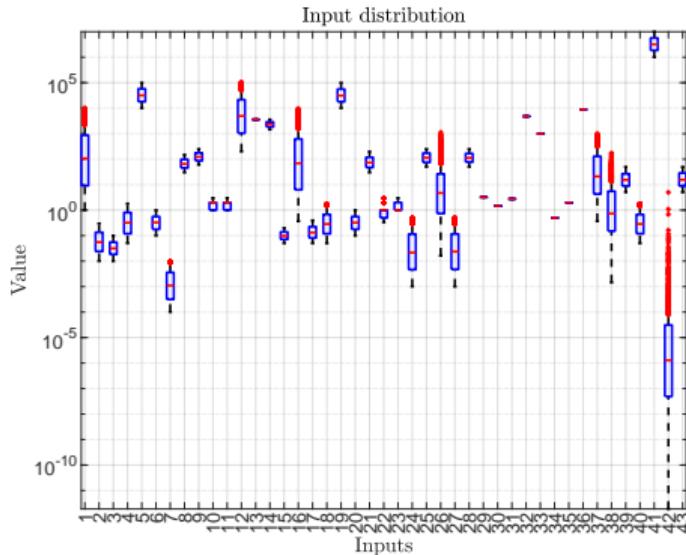
- En régime permanent et en BO
- Temps de calcul

## ► Modèle de design



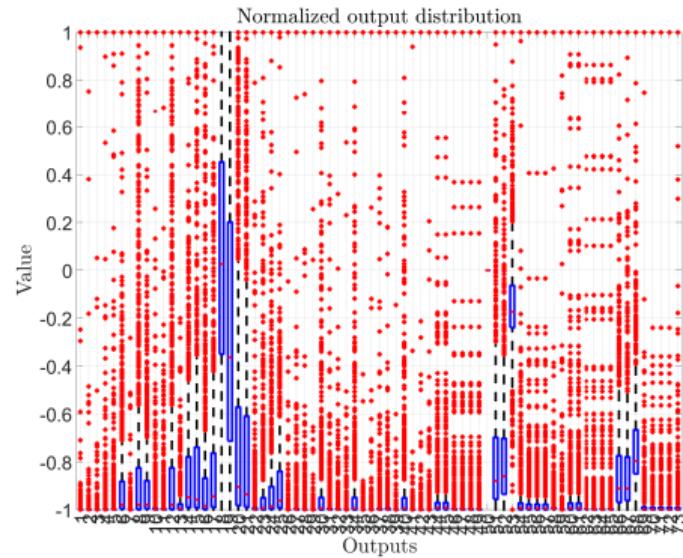
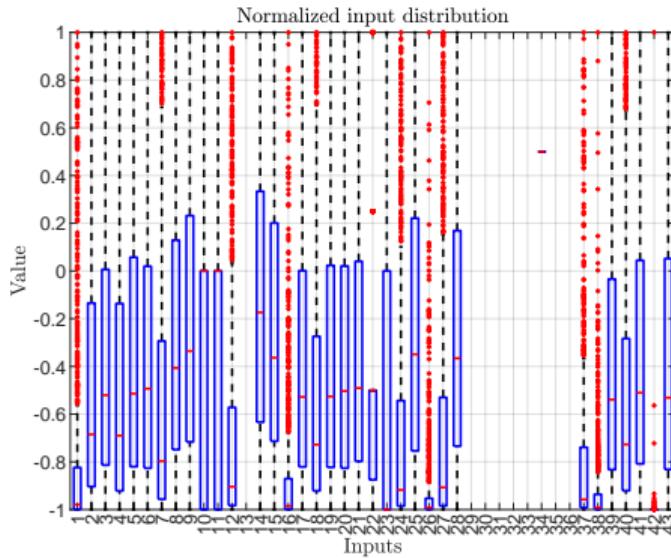
# Construction d'une base de donnée

- ▶ Génération de 1000 designs aléatoires
  - ▶ 43 entrées : spécifications + *design drivers* (DC-DC uniquement)
  - ▶ 73 sorties : valeurs des paramètres correspondants récupérés sur PLECS

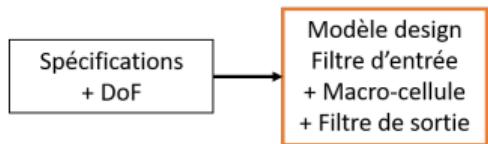


# Construction d'une base de donnée

- Génération de 1000 designs aléatoires
  - 43 entrées : spécifications + *design drivers* (DC-DC uniquement)
  - 73 sorties : valeurs des paramètres correspondants récupérés sur PLECS

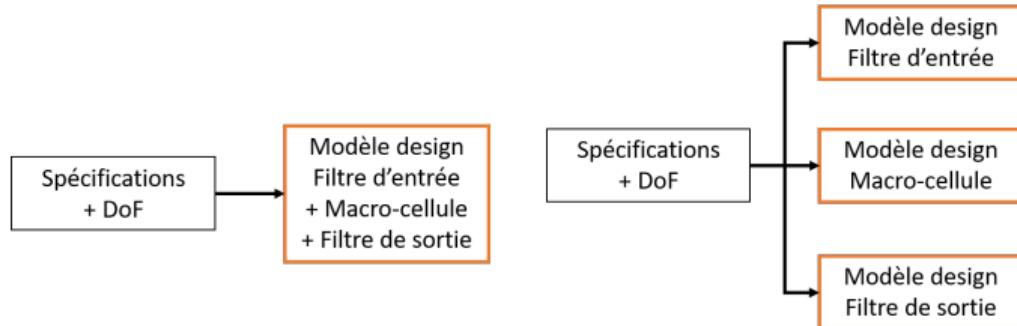


# Structure du modèle de design



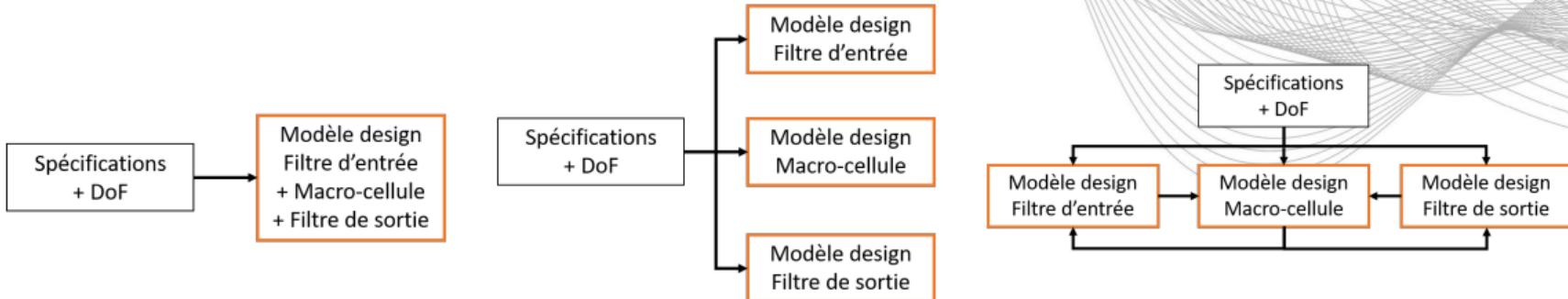
- ▶ Pas de modularité

# Structure du modèle de design



- ▶ Pas de modularité
- ▶ Modularité préservée
- ▶ Pas de connexions entre les différents éléments

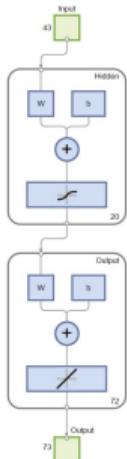
# Structure du modèle de design



- ▶ Pas de modularité
- ▶ Modularité préservée
- ▶ Pas de connexions entre les différents éléments
- ▶ Réseaux de neurones interconnectés : boucle algébrique très complexe!!
- ▶ Ou version itérative (problèmes de convergence et de temps de calcul)

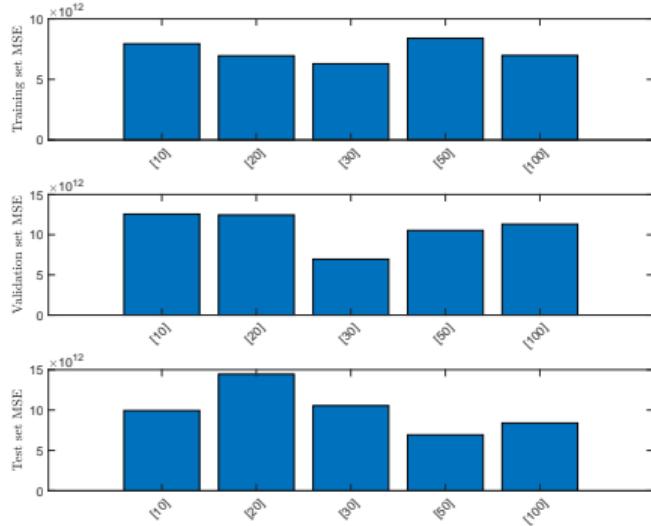
# Premiers résultats

## Modèle de design global



Une couche (*hidden layer*):

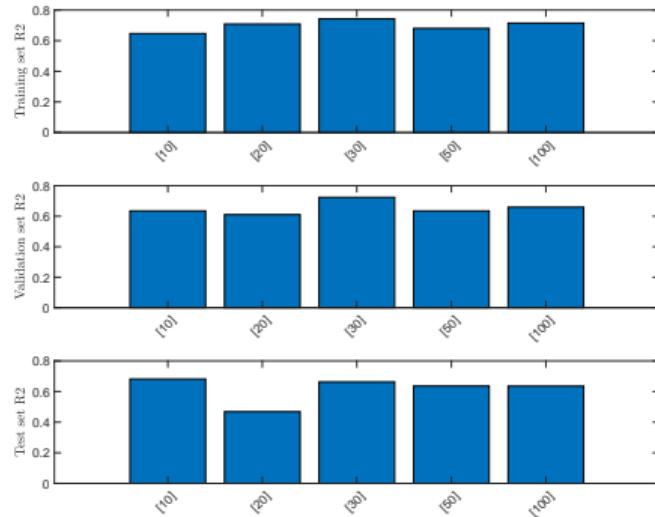
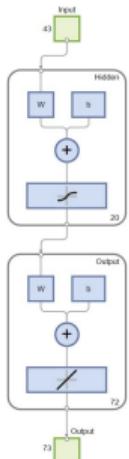
- ▶ Tansig (fct d'activation)
- ▶ 10, 20, 30, 50, 100 neurones



$$\text{MSE} = \frac{1}{N n_y} \sum_{i=1}^N \sum_{j=1}^{n_y} (y_{ij} - \hat{y}_{ij})^2$$

# Premiers résultats

## Modèle de design global



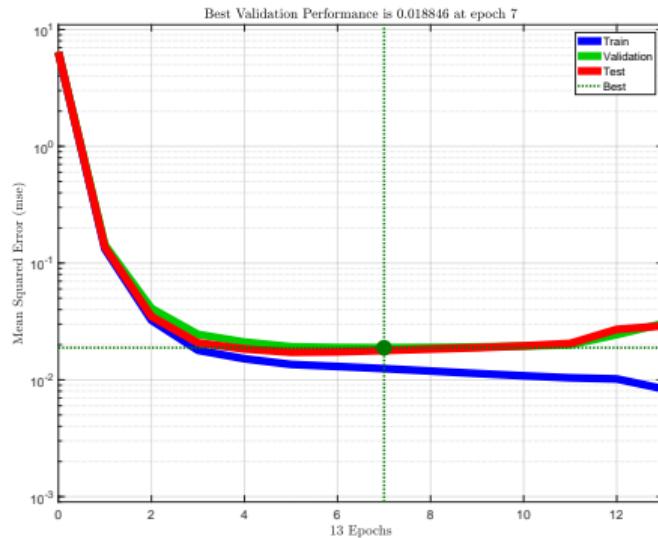
Une couche (*hidden layer*):

- ▶ Tansig (fct d'activation)
- ▶ 10, 20, 30, 50, 100 neurones

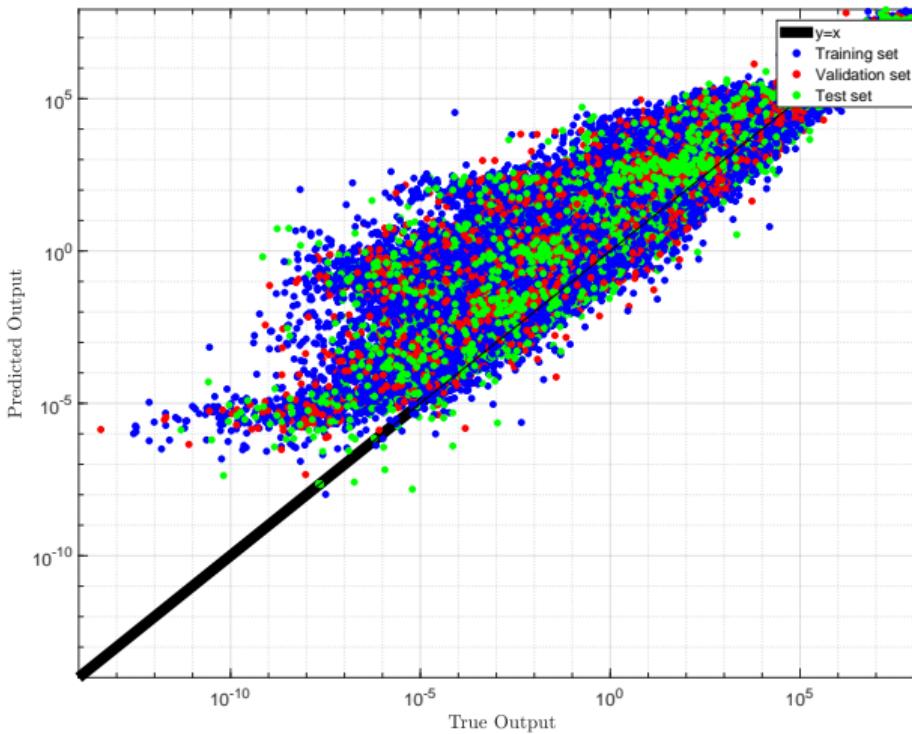
$$R^2 = 1 - \frac{\sum_{i=1}^N \sum_{j=1}^{n_y} (y_{ij} - \hat{y}_{ij})^2}{\sum_{i=1}^N \sum_{j=1}^{n_y} (y_{ij} - \bar{y}_j)^2}$$

# Entraînement du modèle global de design

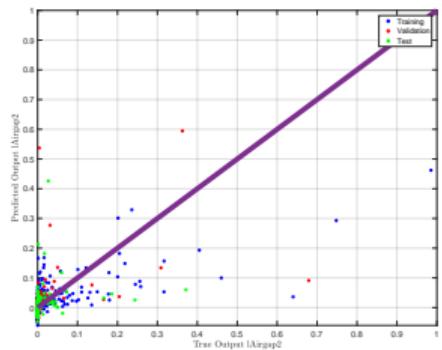
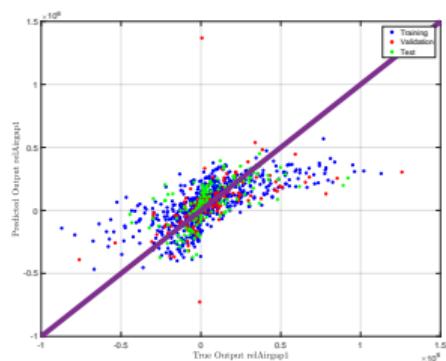
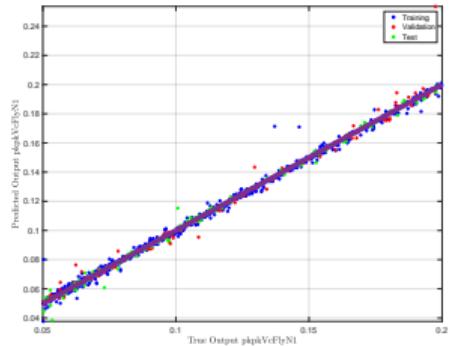
- ▶ Focus sur le réseau 1 couche 30 neurones
- ▶ Arrêt au bout de 13 epochs (9 minutes)



# Performances du modèle global de design



# Performances du modèle global de design



## Et avec un modèle plus complexe?

Couches		1 layer					2 layers		
Neurones		10	20	30	50	100	10	20	30
MSE	train	7.94e12	6.95e12	6.29e12	8.40e12	6.98e12	1.56e13	7.37e12	8.19e12
	val	1.26e13	1.25e13	6.95e12	1.05e13	1.13e13	1.06e13	1.14e13	1.10e13
	test	9.94e12	1.44e13	1.05e13	6.92e12	8.39e12	1.66e13	5.63e12	8.22e12
R2	train	0.65	0.71	0.74	0.68	0.71	0.41	0.70	0.67
	val	0.64	0.61	0.72	0.63	0.66	0.47	0.67	0.63
	test	0.68	0.47	0.66	0.64	0.64	0.39	0.72	0.64

## Et avec un modèle plus complexe?

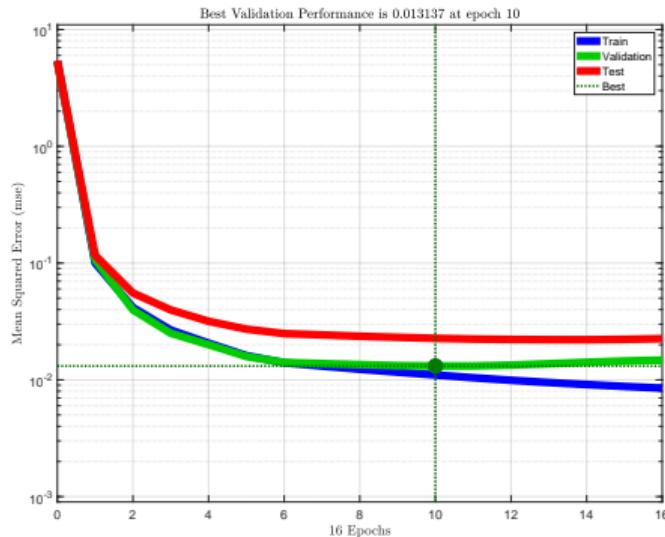
Couches		1 layer					2 layers		
Neurones		10	20	30	50	100	10	20	30
MSE	train	7.94e12	6.95e12	6.29e12	8.40e12	6.98e12	1.56e13	7.37e12	8.19e12
	val	1.26e13	1.25e13	6.95e12	1.05e13	1.13e13	1.06e13	1.14e13	1.10e13
	test	9.94e12	1.44e13	1.05e13	6.92e12	8.39e12	1.66e13	5.63e12	8.22e12
R2	train	0.65	0.71	0.74	0.68	0.71	0.41	0.70	0.67
	val	0.64	0.61	0.72	0.63	0.66	0.47	0.67	0.63
	test	0.68	0.47	0.66	0.64	0.64	0.39	0.72	0.64

$$\text{MSE} = \frac{1}{N n_y} \sum_{i=1}^N \sum_{j=1}^{n_y} (y_{ij} - \hat{y}_{ij})^2$$

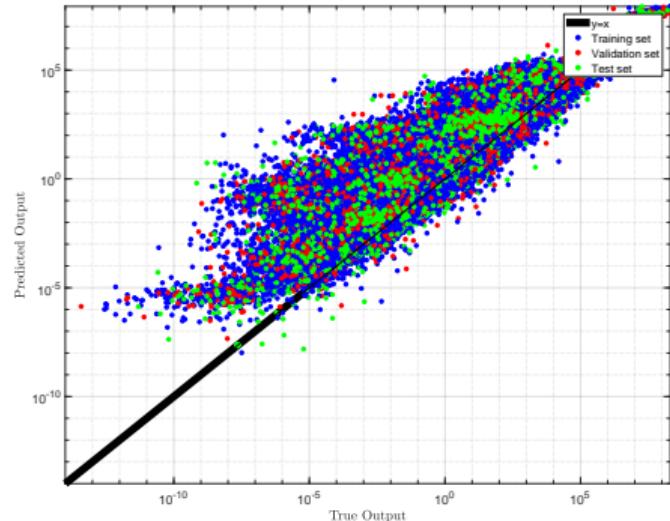
- ▶ Entraînement sur les entrées-sorties normalisées
- ▶ Performances calculées sur les prédictions dénormalisées!

# Entraînement du modèle global de design

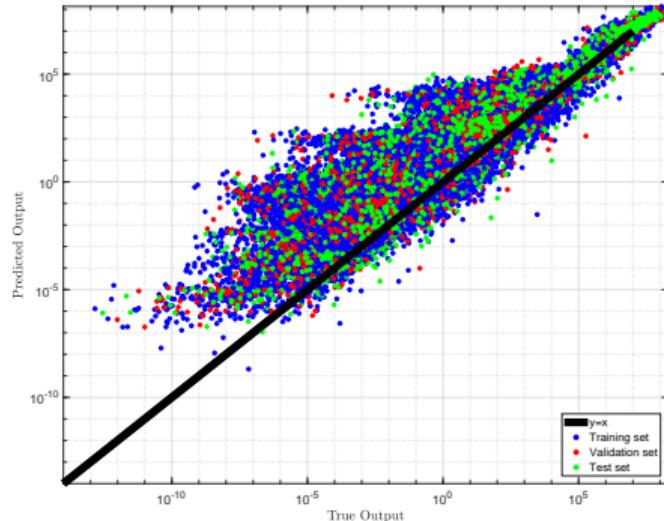
- ▶ Focus sur le réseau 2 couches de 30 neurones
- ▶ Arrêt au bout de 16 epochs (10 minutes)



# Performances du modèle global de design



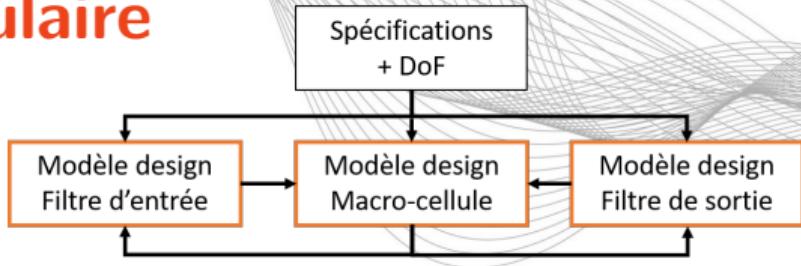
1 couche à 30 neurones



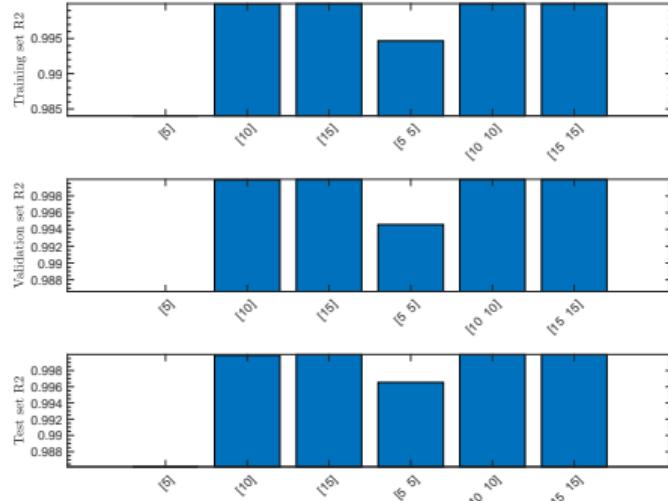
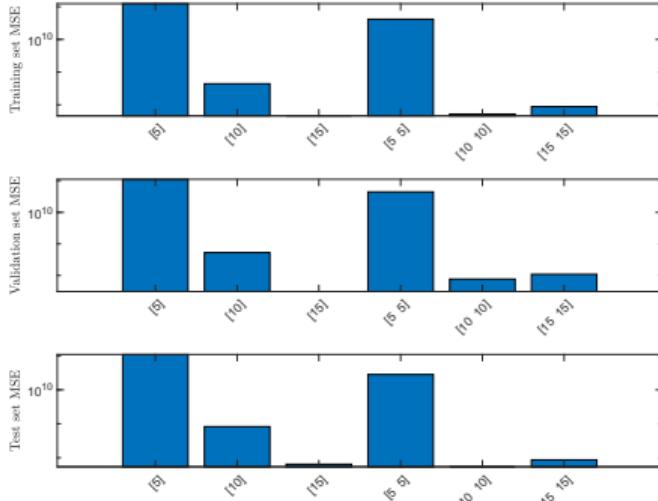
2 couches à 20 neurones

# Vers une modélisation modulaire

## Exemple de la macro-cellule

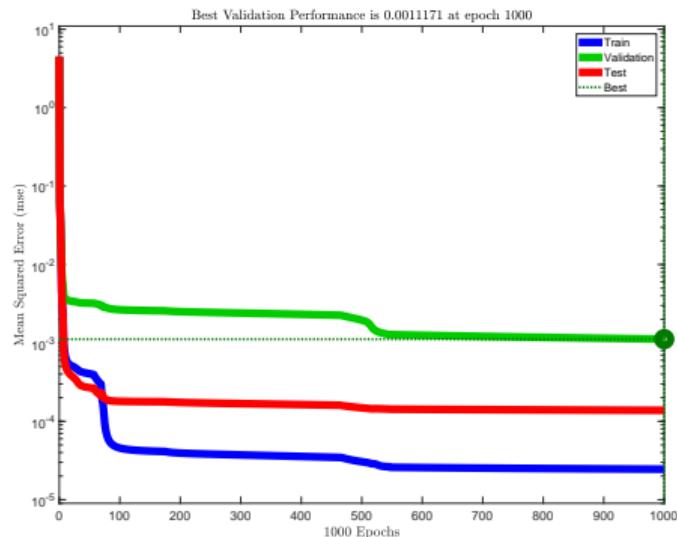


- ▶ 12 entrées et 8 sorties



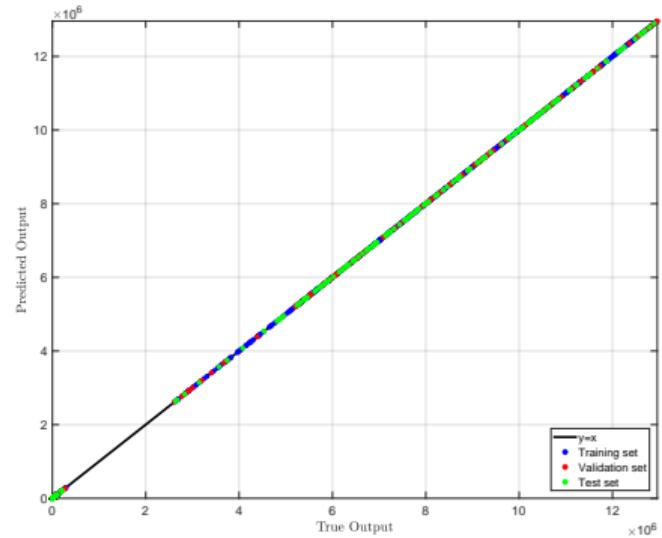
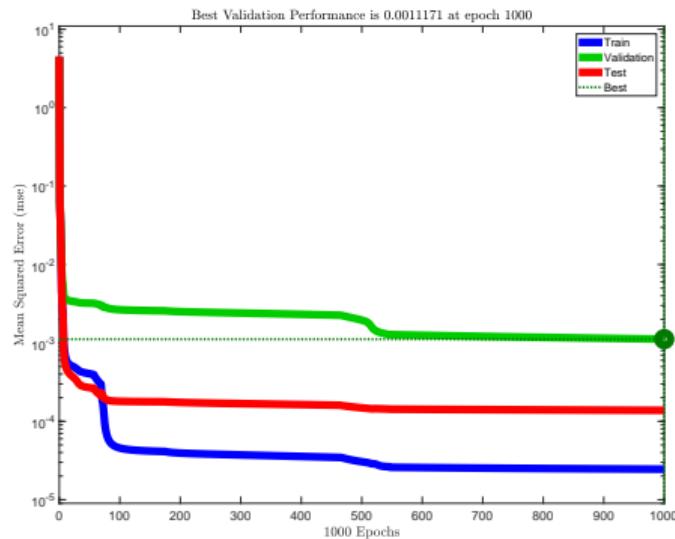
# Entraînement du modèle de design de la macro-cellule

- ▶ Focus sur le réseau 1 couche 15 neurones
- ▶ Stop à 1000 epochs après 45,7 secondes seconds



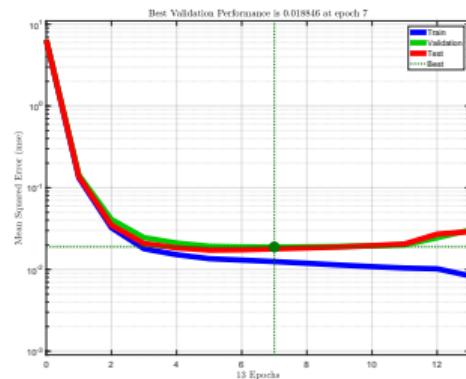
# Entraînement du modèle de design de la macro-cellule

- ▶ Focus sur le réseau 1 couche 15 neurones
- ▶ Stop à 1000 epochs après 45,7 secondes seconds

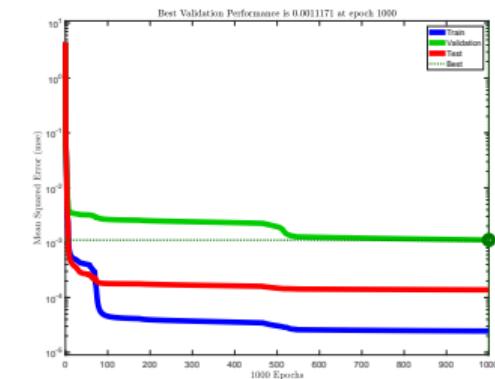


# Discussions

- ▶ Optimisation des hyperparamètres
- ▶ Tester la capacité de généralisation au delà de la base de données
- ▶ Autres structures à explorer pour le modèle de design
  - ▶ Autres types de réseaux de neurones ou utilisation d'arbres de régression (gradient boosting)
  - ▶ Explorer la voie des 3 modèles indépendants
  - ▶ Intégrer de la physique



Entraînement du modèle de design global



Entraînement du modèle de design de  
macro-cellule  Laplace

# Discussions

- ▶ Modèle à intégrer dans une boucle d'optimisation (méta-heuristique)

# Discussions

- ▶ Modèle à intégrer dans une boucle d'optimisation (méta-heuristique)
- ▶ Utiliser des méthodes de quantification d'incertitudes

# Discussions

- ▶ Modèle à intégrer dans une boucle d'optimisation (méta-heuristique)
- ▶ Utiliser des méthodes de quantification d'incertitudes
- ▶ Inclure le type de commande en entrée et ses paramètres en sortie

# Discussions

- ▶ Modèle à intégrer dans une boucle d'optimisation (méta-heuristique)
- ▶ Utiliser des méthodes de quantification d'incertitudes
- ▶ Inclure le type de commande en entrée et ses paramètres en sortie
- ▶ Est ce que l'apprentissage permet vraiment d'aller plus vite?
  - ▶ Formaliser et utiliser la résolution analytique sans PLECS
  - ▶ Interpolation pour le modèle de design

# Discussions

- ▶ Modèle à intégrer dans une boucle d'optimisation (méta-heuristique)
- ▶ Utiliser des méthodes de quantification d'incertitudes
- ▶ Inclure le type de commande en entrée et ses paramètres en sortie
- ▶ Est ce que l'apprentissage permet vraiment d'aller plus vite?
  - ▶ Formaliser et utiliser la résolution analytique sans PLECS
  - ▶ Interpolation pour le modèle de design

IAPUCA : IA pour l'électronique de PUissance et la Commande  
Associée

**Groupe de travail :** T. Meynard, G. Fontes, P. Kergus, R. Ruelland,  
M. Fadel, G. LeGoff, A. Picot, J. Regnier, A. Llor, G. Gateau, F.  
Messine, D. Bonnafous, E. Monmasson



shutterstock.com - 1031749363