

Aide au dimensionnement des inductances de puissance par l'utilisation de réseaux de neurones

Cyrille GAUTIER

Safran Tech, Safran, 1, rue des jeunes bois, 78772 Magny-les-Hameaux

RESUMÉ L'objectif du travail présenté dans cet article est de montrer comment des réseaux de neurones peuvent être utilisés pour déterminer le comportement magnétique et thermique d'un composant magnétique. Les résultats d'un grand nombre de simulations par éléments finis sont utilisés pour entraîner deux réseaux de neurones, un pour le dimensionnement magnétique et l'autre pour le comportement thermique. Après validation, ces réseaux de neurones permettent d'obtenir rapidement le comportement de composants magnétiques en fonction de contraintes géométriques, électriques et thermiques. La génération des données, la définition et la validation des réseaux de neurones utilisés et les résultats obtenus sont présentés. Ces résultats pourront ensuite être utilisés dans un processus d'optimisation appliqué au dimensionnement global d'un convertisseur statique de puissance [1].

Mots-clés – Modélisation, composant magnétique, réseau de neurones.

1. INTRODUCTION

L'électrification avancée des aéronefs (M.E.A.) et l'électrification de la propulsion (M.E.P.) sont une des composantes sur lesquelles travaille l'industrie pour réduire les émissions de CO₂ liées à l'activité aéronautique [2]. L'utilisation de convertisseurs électroniques de puissance permet d'apporter des solutions performantes pour le pilotage des machines électriques, et permet également d'envisager des architectures électriques de bus aéronautiques flexibles, souvent élaborées autour de la distribution d'un bus DC [3]. Pour rendre ces solutions acceptables, il faut relever les deux défis majeurs que constituent la fiabilité et la masse des équipements électriques à embarquer. Dans le cas des convertisseurs statiques, les composants magnétiques représentent pour la majorité des topologies une part importante de la masse totale. Il est donc important de pouvoir optimiser leur dimensionnement et celui du convertisseur dans lequel ils sont utilisés [4].

Des travaux précédents sur les stratégies d'optimisation du dimensionnement des composants magnétiques ont été menés en utilisant un pré-dimensionnement basé sur une formulation analytique et l'utilisation d'une modélisation par éléments finis 2D associée à une optimisation multi-objectifs [5]. Cette méthodologie permet d'obtenir des résultats pertinents mais reste assez complexe à mettre en œuvre, et surtout elle se place dans l'hypothèse où les contraintes magnétiques sont prédominantes en tenant compte d'un niveau maximum de pertes prédéfini.

L'utilisation de logiciels de calcul par éléments finis 2D ou 3D permet d'obtenir des résultats précis, au détriment du temps de calcul nécessaire à la modélisation et à la résolution du problème. Cependant, même avec ces logiciels, des approximations sont faites qui limitent la précision des résultats. La modélisation du comportement des matériaux magnétiques reste perfectible, et la prise en compte de l'évolution de la température, qui influe sur les paramètres, demande l'utilisation de simulations multi-physique. Les simulations électromagnétiques et thermiques peuvent cependant être réalisées à partir de la même géométrie. Dans l'article [6], les auteurs analysent le potentiel des réseaux neuronaux artificiels (ANN) pour la modélisation et l'optimisation inductances. Ces chercheurs proposent l'ou-

til *AI-Mag* sous ©Matlab pour aider à la conception des inductances. Ils mettent à la disposition de la communauté les données, issues de leurs simulations par éléments finis 3D, qu'ils ont utilisés pour entraîner l'ANN qui est au cœur de leur outil, soit 18522 simulations thermique et 18444 simulations magnétiques. Ce travail de référence démontre les performances des réseaux de neurones appliqués au pré-dimensionnement des inductances de puissance. Deux parties sont à considérer :

— La génération des données

— L'entraînement et la validation des réseaux de neurones.

Un des objectifs de ce travail est d'enrichir les données utilisées pour entraîner les réseaux de neurones en introduisant une diversité de matériaux magnétiques et en réalisant les simulations sur une plage de fréquence permettant la prise en compte précise des effets de peau et de proximité. Le second objectif est l'utilisation et la maîtrise d'outils actuels et performants pour la partie sur les réseaux de neurones. Enfin, une contrainte est d'utiliser des outils libres et disponibles pour permettre un partage simple des résultats. Le choix a été fait de travailler avec le logiciel FEMM pour la partie simulation par éléments finis [7], et en python pour la partie réseaux de neurones (bibliothèques Keras et Tensorflow), avec un support GPU.

2. GÉNÉRATION DES DONNÉES

L'élément fondamental pour travailler avec des réseaux de neurones est d'avoir un ensemble de données ("*Dataset*") suffisamment riche et couvrant l'ensemble du domaine défini par les paramètres utilisés en entrée. L'idéal serait de disposer de données issues de réalisation et de caractérisation de composants magnétiques. A défaut, Nous avons d'abord utilisé les données mises à disposition dans *AI-Mag* et nous avons ensuite choisi de générer d'autres données en utilisant la simulation par éléments finis 2D (logiciel FEMM). Le script permettant l'automatisation des simulation a été initialement écrit sous Octave avant d'être transcrit en python en utilisant la librairie *pyFEMM*. Toute la partie relative aux réseaux de neurones étant également codée en Python, cela permet d'avoir un environnement de développement unique.

2.1. Modélisation

Pour la modélisation par éléments finis, nous avons opté pour une simulation axisymétrique. Une table de géométries basées sur les noyaux E, RM, P et PM a été utilisée pour déterminer les dimensions. La paramétrisation du noyau est représenté sur la figure 1 : le cas d'un entrefer centré est représenté, mais la possibilité de modéliser un entrefer complet entre les deux parties du noyau ou un entrefer réparti sur la jambe centrale est également implémenté dans le code de génération de la géométrie.

Dans l'avancement actuel des travaux, seul le matériau ferrite N87 (Epcos/TDK) avec une prise en compte de sa courbe B(H) a été utilisé pour la génération des données. Un des objectifs des travaux futurs et d'ajouter d'autres matériaux de type Ferrite mais également des poudres de fer type MPP ou XFlux (Magnetics) pour la génération de données. La prise en compte de matériaux nanocristallins demande de redéfinir la géométrie du

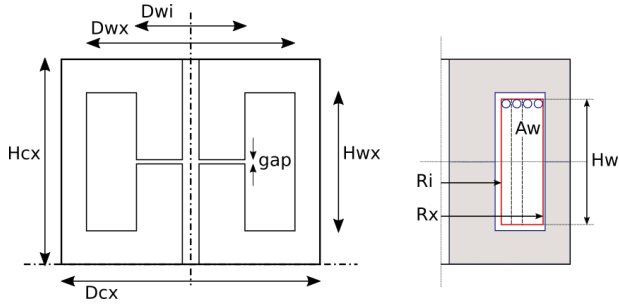


FIG. 1. Paramétrisation du noyau magnétique (noyau PM)

noyau et le mode de simulation. Pour le bobinage, nous utilisons le modèle homogénéisé défini dans *FEMM* [7]. L'utilisation d'un bobinage homogénéisé permet de représenter chaque bobinage par la surface qu'il occupe sans représenter chaque conducteur, donc de simplifier le modèle géométrique et le maillage. Le gain en temps de modélisation et de simulation est conséquent, tout en gardant la prise en compte des effets de peau et de proximité. Pour le calcul du coefficient de remplissage, la surface du bobinage est décomposée en hexagones afin d'être cohérent avec la définition des bobinages homogénéisés (2).

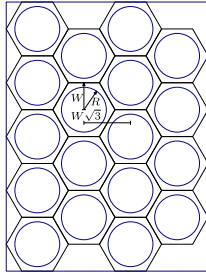


FIG. 2. Remplissage de la fenêtre de bobinage A_w par un motif hexagonal

N représente le nombre de spires. R représente le rayon du conducteur et W la demi-hauteur de l'hexagone entourant les conducteurs. Le ratio k_{fill} entre la surface occupée par les hexagones et celle occupée par les conducteurs est donnée par l'équation 1. Une façon simple de caractériser le coefficient de remplissage du bobinage est de regarder le rapport $c = \frac{R}{W}$. Ce rapport est utilisé pour la génération des données en simulation.

$$k_{fill} = \frac{\pi R^2}{2\sqrt{3}W^2} \quad (1)$$

Le coefficient de remplissage k_w utilisé comme donnée pour les réseaux de neurones correspond au rapport entre la surface de cuivre A_{cu} et la surface de bobinage.

$$k_w = \frac{N(\pi R^2)}{A_w} \quad (2)$$

Pour la modélisation thermique, un modèle homogénéisé du bobinage est également utilisé, les pertes du noyau et du bobinage sont introduites comme source de chaleur volumique pour les deux blocs. L'échange thermique du composant magnétique avec l'air ambiant est défini par le coefficient de convection thermique h_{conv} . On se trouve confronté ici à une limite forte de la modélisation 2D, puisque l'on ne peut pas représenter simplement en même temps la surface d'échange du bobinage et celle du noyau, contrairement à une modélisation en 3D. [8]. Pour chaque jeu de paramètres géométriques et de contraintes magnétiques, un ensemble de simulation en fréquence est effectué et les données de sortie sont calculées. De même un ensemble de

simulations thermiques est réalisé et les élévations de température calculées sont enregistrées. Les données géométriques utilisées en entrée des deux réseaux de neurones sont listées dans le tableau 1.

TABLEAU 1. Données géométriques

| Données | Commentaires | Unité |
|--------------------------|----------------------------|----------------|
| D_{cx}, D_{wi}, D_{wx} | Dimensions du noyau | m |
| H_{cx}, H_{wx} | Dimensions du noyau | m |
| A_e | Surface équivalente de fer | m ² |
| A_w | Surface bobineable | m ² |
| d_{gap} | entrefer total | m |
| d_{iso} | épaisseur isolant | m |
| k_w | coefficient de remplissage | p.u. |

2.2. Données magnétiques et thermiques utilisées en entrée

En plus des données géométriques, les données utilisées en entrée pour l'entraînement des réseaux de neurones magnétiques (tableau 2), et thermiques (tableau 3) sont définies. Les données géométriques et magnétiques sont utilisées pour entraîner le réseau de neurones magnétique, et les données géométriques et thermiques sont utilisées pour entraîner le réseau de neurones thermique.

TABLEAU 2. Données magnétiques

| Données | Commentaires | Unité |
|-----------------|-------------------------------|-------|
| I_w | Courant dans le bobinage | A |
| L_{norm} | inductance normalisée L/N^2 | H |
| β | coefficient de Steinmetz | p.u. |
| μ | perméabilité relative | p.u. |
| f | fréquence | Hz |
| R_{AC}/R_{DC} | ratio résistance AC sur DC | p.u. |

TABLEAU 3. Données thermiques

| Données | Commentaires | Unité |
|---------------|---------------------------|----------------------|
| T_{amb} | Température ambiante | °C |
| h_{conv} | coefficient de convection | W/(m ² K) |
| P_{core} | pertes magnétiques | W |
| $P_{winding}$ | pertes cuivre | W |

Une normalisation, décrite dans la suite, est appliquée à ces données avant de les utiliser en entrée des réseaux de neurones pour l'entraînement. Cette normalisation est fondamentale pour l'entraînement des réseaux de neurones afin que leurs coefficients soient répartis avec des poids équivalents pour tous les résultats de simulation.

2.3. Données en sorties

Les données utilisées en sortie (tableau 4) sont normalisées par rapport au nombre de spires N et au courant pic \hat{I} . Le choix de ces variables est important car il permet de se limiter aux paramètres de champs, d'énergie et de température moyennés

sur le volume. Une fonction de coût de type erreur quadratique moyenne (*mse*) est calculée sur chacune des sorties et moyennée pour chacun des réseaux, avec la possibilité d'affecter des coefficients de pondération.

TABLEAU 4. Données de sortie FEM/ANN

| Var. | Définition | Unité |
|--------------------------|--|-------------------|
| BH_{int} | $\frac{1}{N^2 \bar{f}^2} \left(\iiint_{all} \hat{B} \hat{H} dV \right)$ | H |
| B_{int} | $\frac{1}{N \bar{f}} \left(\frac{1}{V_c} \iiint_c B^\beta dV \right)^{1/\beta}$ | T |
| J_{int} | $\frac{1}{N \bar{f}} \left(\frac{1}{V_w} \iiint_w \hat{J}^2 dV \right)^{\frac{1}{2}}$ | A m ⁻² |
| H_{int} | $\frac{1}{N \bar{f}} \left(\frac{1}{V_w} \iiint_w \hat{H}^2 dV \right)^{\frac{1}{2}}$ | A m ⁻¹ |
| $\Delta T_c, \Delta T_w$ | Delta température max (noyau, bobinage) | K |

3. RÉSEAUX DE NEURONES

Le framework *Keras* a été choisi pour l'implémentation en python de la partie deep learning (apprentissage profond) du projet. C'est une librairie de haut-niveau qui propose des blocs à assembler pour implémenter les structures de réseaux de neurones et développer le modèle d'apprentissage profond. Ce framework s'appuie sur le backend Tensorflow pour la manipulation et la différenciation bas niveau des données organisées sous forme de tenseurs [9]. D'un point de vue matériel, nous avons utilisé la possibilité d'utiliser un GPU (Nvidia RTX 3060 mobile) pour accélérer les calculs sans avoir à modifier le code.

3.1. Normalisation des données

La première partie du travail a consisté à étudier les données pour évaluer leur répartition. Afin d'améliorer la précision sur la prédiction des données, nous avons appliqué une transformation logarithmique à certaines données en fonction de leur distribution. Une normalisation de type min max a ensuite été appliquée à l'ensemble des données, en utilisant la fonction *MinMaxScaler* de la librairie *sklearn*.

$$y = \frac{(x - \min)}{(\max - \min)} \quad (3)$$

Le réglage de l'opérateur *mag_scaler* de *MinMaxScaler()* se fait sur les données d'entraînement (fonction *fit()*), et la transformations (fonction *transform()*) peut ensuite être appliquée aux données d'entraînement, de test ou aux données qui seront utilisées en entrée du réseau lors de son utilisation. Cette étape de préparation des données est essentielle pour permettre par la suite un apprentissage efficace et une bonne performance des prédictions.

3.2. Structure du réseau de neurones

Dans la seconde partie, nous avons implémenté un réseau de neurones de type régression pour la modélisation magnétique. Les données utilisées sont issues de résultats de simulation et ne sont pas bruitées. Dans ces conditions, il existe un risque que le réseau apprenne précisément les données d'entraînement mais qu'il ne soit pas en mesure d'extrapoler efficacement le résultat associé à de nouvelles données d'entrées. Nous avons utilisé une couche *GaussianNoise* en entrée afin d'ajouter artificiellement du bruit et de limiter ce risque. Après plusieurs essais, nous avons choisi d'utiliser pour la partie principale du réseau deux couches profondes, de 64 et de 32 neurones, avec une fonction d'activation *RELU* (figure 3). Classiquement, nous avons utilisé l'erreur quadratique moyenne (*mse*) comme fonction de coût, et *Adam* comme fonction d'optimisation. L'algorithme *Adam* (adaptive moment estimation) est une extension de la méthode

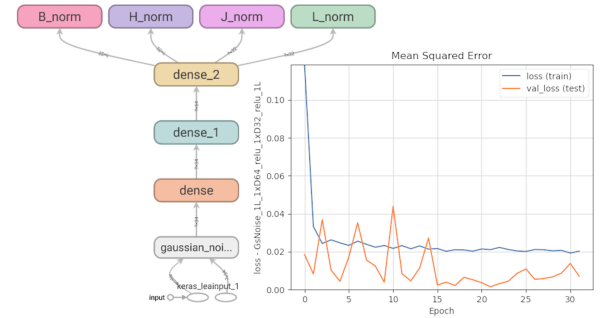


FIG. 3. Magnétique, réseau deux couches profondes, données mises à l'échelle

de descente de gradient stochastique basée sur une estimation adaptative des moments de premier et de second ordre.

Le modèle thermique est plus simple que le modèle magnétique, sans comportement non-linéaire. Après avoir testé un réseau avec une structure équivalente au réseau utilisé pour la modélisation magnétique, nous avons décidé d'utiliser un réseau de neurones plus simple en implémentant une seule couche profonde de 32 neurones et une fonction d'activation de type *sigmoid* (figure 4).

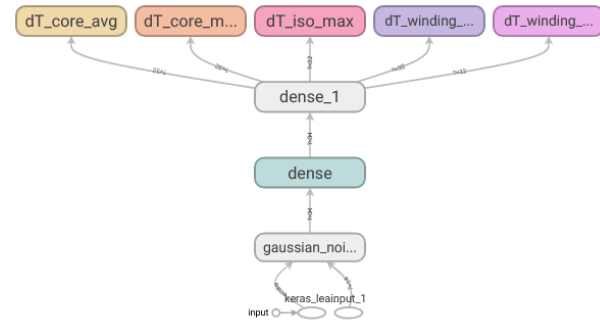


FIG. 4. Structure du réseau simplifié pour le modèle thermique

La convergence se fait rapidement pour ce réseau et le fait d'avoir enlevé une couche profonde ne dégrade pas l'apprentissage (figure 5).

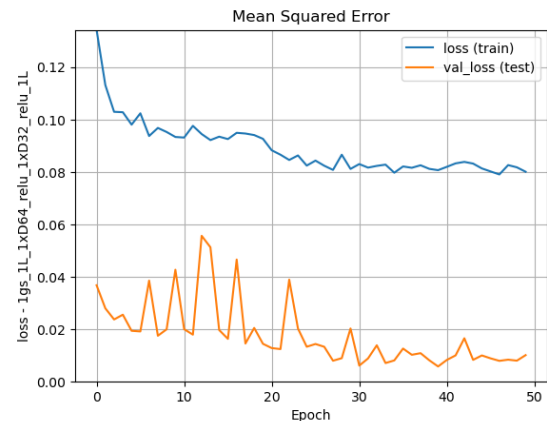


FIG. 5. convergence du réseau simplifié pour le modèle thermique

3.3. Entraînement

Les données sont séparées en données d'entraînement *train* et en données de test *test* dans un rapport 80% / 20%. Seules les

données d'entraînement sont utilisées lors de la phase d'apprentissage, les données de test étant utilisées à la fin de la phase d'apprentissage pour évaluer la performance du réseau entraîné. Cette phase d'apprentissage est structurée en époques (*epoch*), et dans chaque époque les données sont traitées par lot (*batch*). Il y a une itération pour chaque lot, qui comprend le traitement des données du lot, le calcul de la fonction de coût (*loss*) ainsi que la modification et la rétro-propagation des facteurs de poids du réseau. Une époque correspond à un cycle pendant lequel l'ensemble des données a été utilisé. Afin d'empêcher le sur-apprentissage, nous avons implémenté la fonction *earlystopping* pour que l'apprentissage s'arrête lorsque la mesure de performance ne s'améliore plus. Plus précisément, nous avons défini comme critère que l'erreur quadratique moyenne sur les données de test, représentée par *val_loss*, ne diminue plus pendant 10 époques (*patience* = 10). Une fois la structure des réseaux de neurones fixée, nous avons utilisé une validation croisée *KFold* en cinq sous-groupes pour estimer la performance des deux réseaux de neurones.

3.4. Validation

3.4.1. Validation à partir des données de test

Les prédictions des données de test pour chacune des sorties sont représentées figure 6. La corrélation entre données réelles et données prédites est très bonne pour les quatre grandeurs ce qui valide que les prédictions sont de bonne qualité. De même,

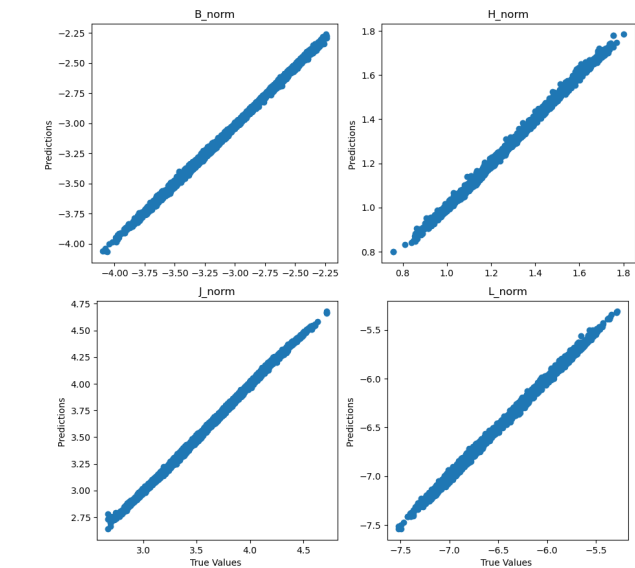


FIG. 6. Magnétique, réseau deux couches profondes, prédiction des données

les dispersions sur les erreurs, figure 7 montrent que les erreurs de prédiction sont bien réparties. Ces indicateurs permettent de maitriser la performance de ces réseaux pour la prédiction et pour leur exploitation.

3.4.2. Validation croisée K-Fold

La validation croisée est utilisée pour estimer la performance d'un modèle de réseau de neurones sur les données non utilisées pendant l'entraînement. Cette approche consiste à subdiviser de façon aléatoire l'ensemble des données d'entraînement en *k* sous-ensembles de même dimension. Ensuite, le processus d'entraînement est répété *k* fois en gardant à chaque fois un sous-ensemble différent comme jeu de validation, et en entraînant le modèle avec les (*k*-1) sous-ensembles restant (figure 8). A chaque fois, le score du modèle est calculé, et le modèle est réinitialisé. La performance du modèles est établie à partir de l'ensemble des scores intermédiaires calculés. Lors de la

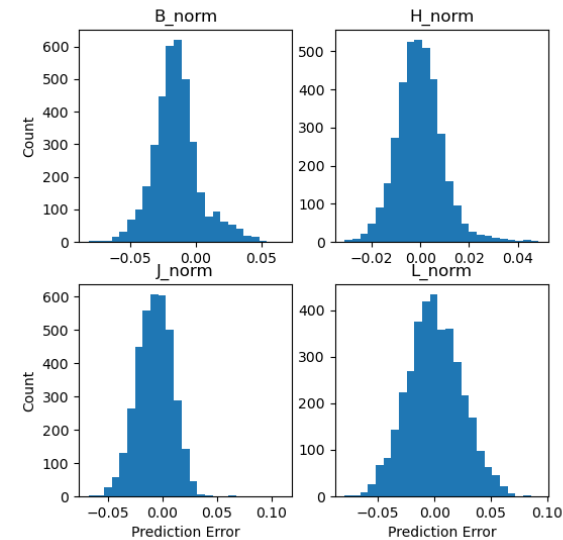


FIG. 7. Magnétique, distribution de l'erreur, réseau deux couches profondes

mise en œuvre de la validation croisée, la méthode *KFold* de la librairie *scikit-learn* a été utilisée pour la séparation en sous-ensemble.



FIG. 8. K-Fold : répartition des données en *k* sous ensembles

La méthode a été implémentée avec *k* = 10, les dix entraînements ont donné des résultats cohérents. La synthèse de la fonction de perte calculée pendant chaque entraînement est donnée table 5. Afin de garder des conditions de calcul équivalente, le nombre d'époques a été fixé de manière identique, à 40, pour tous les entraînements.

TABEAU 5. Validation k-Fold : Score par Fold (*k*=10)

| Fold | Loss |
|---------|---------------|
| 1 | 1.780885e-03 |
| 2 | 5.221064e-03 |
| 3 | 1.486160e-03 |
| 4 | 0.4686831e-03 |
| 5 | 0.6180170e-03 |
| 6 | 0.8955433e-03 |
| 7 | 0.9510671e-03 |
| 8 | 1.0016299e-03 |
| 9 | 9.2838658e-03 |
| 10 | 0.8318156e-03 |
| Average | 2.2538730e-03 |

4. UTILISATION

4.1. Paramètres

Les deux réseaux de neurones, entraînés sur les données magnétiques et sur les données thermiques sont utilisés conjointement pour déterminer le comportement d'un composant magnétique à partir de ses dimensions et des contraintes électriques et thermiques (figure 9). En tenant compte des contraintes élec-

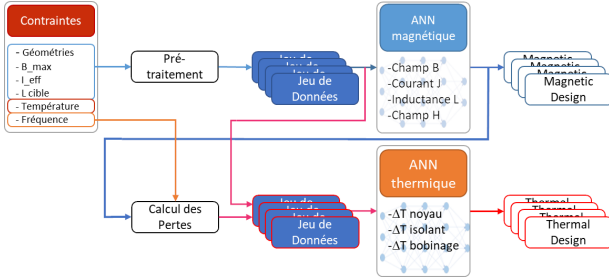


FIG. 9. Utilisation des réseaux de neurones pour déterminer les caractéristiques magnétiques et thermiques des dimensionnements

triques et des objectifs sur les valeurs magnétiques à atteindre, un certain nombre de géométries sont sélectionnées et les plages de dimensions, d'entrefer et de nombre de spires sont définies. La combinatoire des paramètres donne un nombre important de configurations, qui vont servir d'ensemble de données en entrée des réseaux de neurones.

TABLEAU 6. Paramètres et exemple de plages de variation

| nom | min | max | unité | commentaire |
|-----------|-----|------|--------------|--------------------|
| Noyaux | E5 | E100 | | Dimensions |
| B_{max} | 0.1 | 0.4 | T | Champ magnétique |
| J | 5 | 15 | $A\ mm^{-2}$ | Densité de courant |
| gap | 100 | 1000 | μm | Entrefer |
| k_w | 0.3 | 0.6 | p.u. | Coef. bobinage |

Pour chaque jeux de paramètre, une prédiction du comportement magnétique est déterminée en utilisant le réseau de neurones magnétique. Les pertes du noyau et du bobinages sont calculées pour être ajoutées aux données en entrée du réseau de neurones thermique. A l'issue du calcul, l'ensemble de la modélisation magnétique et thermique est réalisée. Un premier filtrage sur les températures et sur B_{max} va limiter le nombre de solutions à comparer. Une des forces de l'utilisation des réseaux de neurones est la rapidité de calcul en prédiction sur un grand nombre de données en entrée. Ainsi, pour un ensemble de 50 000 jeux de données, le calcul demande une trentaine de secondes sur un ordinateur de type PC classique sans utiliser de GPU (Intel Core i5 de 11^e génération, 16Go de ram).

4.2. Résultats

Nous présentons deux cas d'étude : le dimensionnement d'une inductance de 20 μH , 20 A à 20 kHz, avec un objectif de B_{max} de 300 mT, et le dimensionnement d'une inductance de 100 μH , 40 A à la fréquence de 800 Hz avec un champ B limité à 400 mT.

4.2.1. inductance de 20 μH

La fréquence de fonctionnement choisie correspond à une fréquence de découpage usuelle en électronique de puissance, qui impacte directement les pertes dans le noyau et dans le bobinage et peut entraîner des variations de température importantes lorsque l'on cherche à augmenter la compacité. L'ensemble des résultats calculés pour ce premier cas sont présentés figure 10.

Chaque point correspond à un dimensionnement (géométrie, entrefer, L , B_{max} , pertes fer, pertes cuivre, ΔT). En limitant la plage de température et celle de la valeur d'inductance, le nombre de solutions est fortement réduit (figure 11). On peut également faire un zoom sur la valeur de l'inductance (figure 12) pour ensuite aller sélectionner une solution particulière et étudier le résultat de dimensionnement associé. La tableau 7 regroupe les exemples de résultats pour les deux cas.

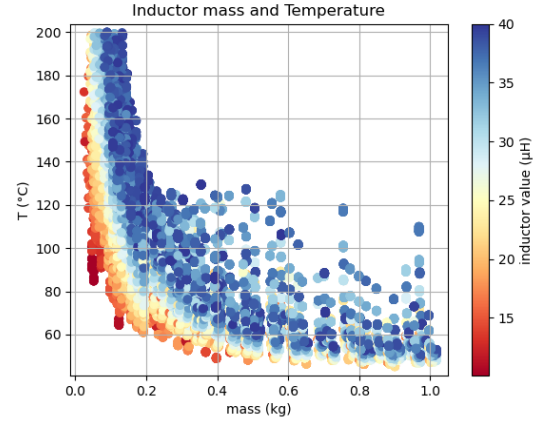


FIG. 10. Résultat des prédictions pour l'ensemble du jeux de paramètres

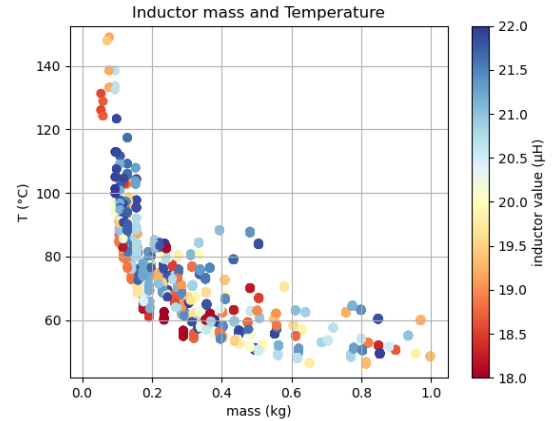


FIG. 11. Résultat des prédictions après filtrage

4.2.2. inductance de 100 μH

Ce cas correspond plus au dimensionnement d'une inductance de filtrage réseau pour une application aéronautique. La fréquence de fonctionnement de 800 Hz ne va pas entraîner de pertes importantes, mais la valeur du courant efficace plus élevé va influencer directement sur la dimension des noyaux la masse des solutions (figure 13). Cela apparaît clairement lorsque l'on effectue un zoom sur la valeur d'inductance (figure 14). Nous avons sélectionné deux dimensionnements pour les comparer dans le tableau 7 :

- noyau E70 : la contrainte d'un entrefer inférieur à 500 μm impose un noyau plus volumineux et entraîne une température de fonctionnement plus élevée que dans la seconde solution
- noyau E56 : en autorisant un entrefer plus grand, ce dimensionnement donne de meilleurs résultats en termes de pertes et de masse.

TABLEAU 7. Sélection de résultats

| Core | N spires | entrefer (mm) | kw | L (μ H) | mass (kg) | T ($^{\circ}$ C) |
|------|----------|------------------|-----|-----------------|--------------|----------------------|
| E47 | 6 | 0.364 | 0.3 | 20.0 | 0.243 | 75 |
| E70 | 5 | 0.462 | 0.3 | 99.97 | 4.01 | 77 |
| E56 | 10 | 1.18 | 0.6 | 100.4 | 1.24 | 57 |

un ensemble de données pour supprimer les données corrompues. Ensuite, la difficulté tiens à la définition des réseaux utilisés (nombre de couches, nombre de neurones par couches, fonctions d'activation), à leurs entraînement et à la validation de la qualité des entraînements. Il faut essayer différentes structures de réseaux de neurones pour déterminer laquelle est la mieux adaptée au problème, mais cela nécessite à chaque fois de réaliser un apprentissage qui demande du temps et des ressources machines importantes. La possibilité de disposer de machines de calcul équipées de GPU pendant cette phase peut l'accélérer de façon très efficace. Une fois le ou les réseaux définis, entraînés et validés, la dernière limite vient de l'obligation de rester dans l'espace défini par les données utilisées pour faire les *prédictions*. Il est nécessaire de bien maîtriser l'ensemble du processus pour obtenir des résultats cohérents.

6. CONCLUSIONS

L'ensemble du processus de modélisation par réseaux de neurones a été réalisé et nous sommes en mesure grâce à leur mise en œuvre d'étudier très rapidement un nombre très important d'échantillons pour aller cibler précisément et rapidement une solution pertinente. Cela nous permet d'optimiser la masse d'un composant magnétique lors des pré-dimensionnements, en tenant compte des contraintes magnétiques et thermiques. Les données d'entrée doivent encore être enrichies par d'autres simulations pour élargir la base des matériaux magnétiques disponibles pour des plages de températures étendues. Enfin, la possibilité d'utiliser un seul réseau de neurones pour modéliser le comportement magnétique et thermique est à étudier comme une évolution logique de ces travaux.

RÉFÉRENCES

- [1] Adrien Voldoire, Jean Luc Schanen, Jean Paul Ferrieux, Alexis Derbey, and Cyrille Gautier. Three-phase PWM voltage-source-inverter weight optimization for aircraft application using deterministic algorithm. *MDPI Electronics*, 2020.
- [2] Jean-Brice Dumont (Airbus), Rolf Henke (DLR), and Bart de Vries (KLM). Fly the green deal. Technical Report JUNE, ACARE, 2022.
- [3] Ashkan Barzkar and Mona Ghassemi. Electric Power Systems in More and All Electric Aircraft : A Review. *IEEE Access*, 2020.
- [4] Timothe Delaforge. Tutorial at epe 2020 : Electronic design automation and optimization algorithms for the next generation of optimal power converters. In *EPE 2020 Tutorial*, Lyon, France, 2020.
- [5] Guillaume Devos, Maya Hage-Hassan, Philippe Dessante, Cyrille Gautier, Adrien Mercier, and Eric Laboure. Optimization strategy for the sizing of passive magnetic components. *2020 22nd European Conference on Power Electronics and Applications, EPE 2020 ECCE Europe*, 2020.
- [6] Thomas Guillod, Panteleimon Papamanolis, and Johann W. Kolar. Artificial Neural Network (ANN) Based Fast and Accurate Inductor Modeling and Design. *IEEE Open Journal of Power Electronics*, June, 2020.
- [7] D. Meeker. Continuum representation of wound coils via an equivalent foil approach. 2006.
- [8] Guillermo Salinas López, Alberto Delgado Exposito, Javier Munoz-Anton, Jesus Angel Oliver Ramirez, and Roberto Prieto Lopez. Fast and Accurate Thermal Modeling of Magnetic Components by FEA-Based Homogenization. *IEEE Transactions on Power Electronics*, 35, 2020.
- [9] François Chollet. *DEEP LEARNING with PYTHON*. Manning edition, 2018.

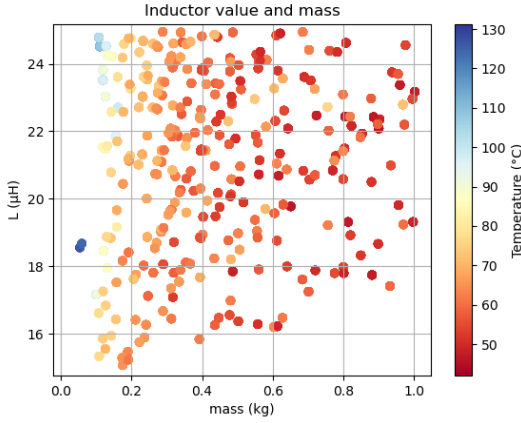


FIG. 12. Résultat des prédictions après filtrage - zoom sur la valeur de L

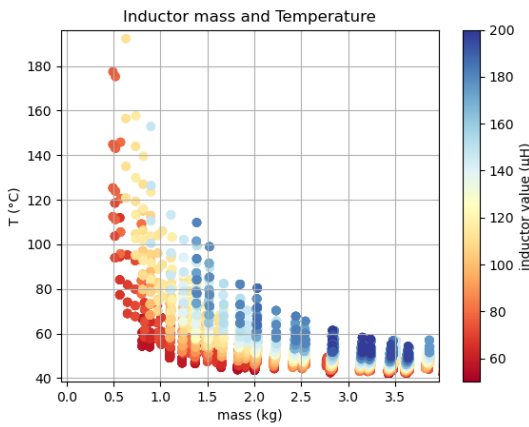


FIG. 13. Résultat des prédictions pour l'ensemble du jeu de paramètres

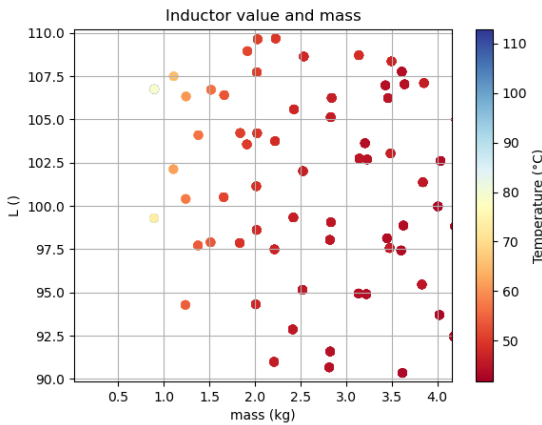


FIG. 14. Résultat des prédictions pour l'ensemble du jeu de paramètres

5. CONTRAINTES ET LIMITES DE LA MÉTHODE

Les réseaux de neurones sont très efficaces pour réaliser rapidement des interpolations sur un ensemble de données important, avec un nombre de paramètres élevé, quand ils sont bien entraînés. La première contrainte vient donc de la nécessité de disposer d'un nombre de données important, couvrant un périmètre suffisant par rapport à l'application envisagée, et de bonnes qualités. Il peut souvent y avoir un travail important à réaliser sur