MITSUBISHI ELECTRIC RESEARCH LABORATORIES
https://www.merl.com

Tandem Neural Networks for Electric Machine Inverse Design

Xu, Yihao; Wang, Bingnan; Sakamoto, Yusuke; Yamamoto, Tatsuya; Nishimura, Yuki; Koike-Akino, Toshiaki; Wang, Ye

TR2023-040 May 16, 2023

Abstract

In electric motor design tasks, multiple design goals often need to be placed on a single motor, and multi-objective optimization plays a significant role. Trade-offs and Pareto front searching are needed, as these design goals or responses cannot be optimized concurrently due to their interdependent nature. However, tuning the motor parameters in the iterative optimization process is typically ineffective and heavily dependent on the expertise of the engineers due to the large number of time- consuming finite-element simulations required to evaluate each motor design candidate. In this paper, we propose an inverse design approach for electric machines based on a tandem neural network, which can effectively provide desired motor design candidates for various design targets without iteration. The one- to-many mapping problem can be avoided by the tandem neural network, which constructs loss functions based on the responses of the generated motor designs. The proposed intelligent design strategy is generally applicable for the design tasks of different types of electric motors.

IEEE International Electric Machines and Drives Conference (IEMDC) 2023

Mitsubishi Electric Research Laboratories, Inc.
201 Broadway, Cambridge, Massachusetts 02139

Tandem Neural Networks for Electric Machine Inverse Design

Yihao Xu
Mechanical and Industrial Engineering
Northeastern University
Boston, MA 02120 USA
xu.yiha@northeastern.edu

Bingnan Wang
Mitsubishi Electric Research Laboratories (MERL)
Cambridge, MA 02139 USA
bwang@merl.com

Yusuke Sakamoto
Mitsubishi Electric Research Laboratories (MERL)
Cambridge, MA 02139 USA
sakamoto@merl.com

Tatsuya Yamamoto
Advanced Technology R&D Center
Mitsubishi Electric Corporation
Amagasaki, Hyogo 661-8661 Japan
yamamoto.tatsuya@ah.mitsubishielectric.co.jp

Yuki Nishimura
Advanced Technology R&D Center
Mitsubishi Electric Corporation
Amagasaki, Hyogo 661-8661 Japan
nishimura.yuki@dw.mitsubishielectric.co.jp

Toshiaki Koike-Akino
Mitsubishi Electric Research Laboratories (MERL)
Cambridge, MA 02139 USA
koike@merl.com

Ye Wang
Mitsubishi Electric Research Laboratories (MERL)
Cambridge, MA 02139 USA
yewang@merl.com

***Abstract*—In electric motor design tasks, multiple design goals often need to be placed on a single motor, and multi-objective optimization plays a significant role. Trade-offs and Pareto front searching are needed, as these design goals or responses cannot be optimized concurrently due to their interdependent nature. However, tuning the motor parameters in the iterative optimization process is typically ineffective and heavily dependent on the expertise of the engineers due to the large number of time-consuming finite-element simulations required to evaluate each motor design candidate. In this paper, we propose an inverse design approach for electric machines based on a tandem neural network, which can effectively provide desired motor design candidates for various design targets without iteration. The one-to-many mapping problem can be avoided by the tandem neural network, which constructs loss functions based on the responses of the generated motor designs. The proposed intelligent design strategy is generally applicable for the design tasks of different types of electric motors.**

Index Terms—Topic— Electric motor, inverse design, machine learning, tandem neural network, surrogate model

I. Introduction

Electric motors are essential components in many aspects of the modern society, including transportation networks, industrial machinery, and household appliances. The need for power-dense, highly efficient, and economically viable motors is ever increasing. When trying to find the best designs for motors, multi-objective design optimization is frequently used to take into account various characteristics that are critical for machine design, such as average torque generation, torque ripple, cogging torque, weight, and material cost. The design parameters are often iteratively updated using evolutionary algorithms such as genetic algorithms, and finite-element analysis (FEA) based numerical simulations are used to assess the performance of each design candidate within the optimization loop [1]. The challenges of this design optimization process can be summarized into the following three aspects. First, the design goals are typically interrelated, which implies that there may be trade-offs or even conflicts between them, making it difficult for one motor to satisfy all the design goals at once. Second, multiple motor design candidates need to be evaluated with numerical simulations (e.g. FEA), which are very time-consuming, especially when a large number of rotor positions or operating points of a motor design candidate need to be evaluated. Lastly, the optimization process is typically not a one-shot operation, meaning the best design(s) cannot be generated within a single optimization step, but rather an iterative process of trial and error. As a result, this procedure requires extremely long computation time with these problems combined. Strong domain knowledge and know-how from motor designers are also necessary in order to conduct the optimization tasks.

In recent years, intelligent design approaches, based on contemporary numerical optimization and machine learning algorithms, have had considerable success across multiple disciplines, including photonic [2], acoustic [3], and mechanical [4] device designs. They have the potential to address these motor design issues and offer a more effective and efficient

This work was done when Y. Xu are with Mitsubishi Electric Research Laboratories. Corresponding Author: Bingnan Wang (`bwang@merl.com`).

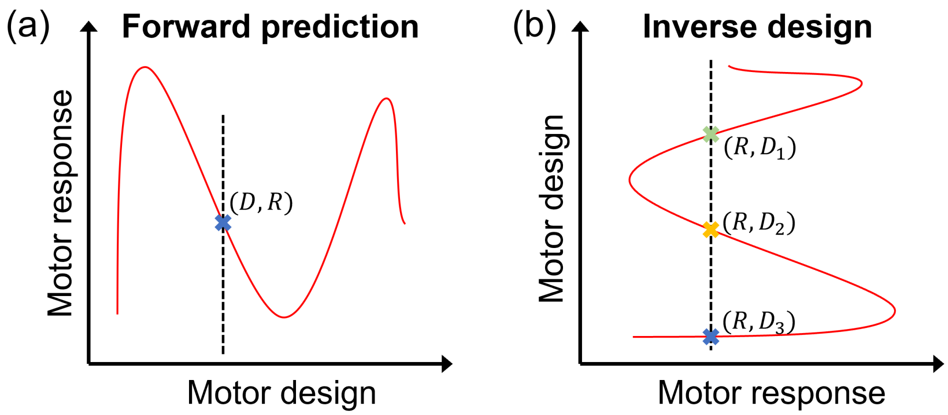


Fig. 1. Mappings of forward prediction and inverse design. (a) Forward prediction mapping, each design D will have one determined response R, (b) Inverse design mapping, a given target response R may correspond to multiple designs D_1, D_2, D_3, etc.

approach for electric motor design tasks. For instance, deep neural networks (DNNs) mimic the neuron behaviors in the human brain, connecting layers with a linear transformation and a nonlinear activation [5]–[7]. The DNNs may be able to map or infer any functions by stacking multiple layers of neurons. In physical systems, DNNs can be constructed and trained as a forward surrogate model that can assess a specific physical design and output the predicted response without physical simulation. Such surrogate models can be used to speed up the design optimization process. Various deep learning models have been proposed as surrogate models for electric motors to replace finite-element simulations in the design optimization process [8]–[14]. Note that the iterative optimization process is still required with DNN based surrogate models.

On the other hand, DNNs can also be built to work as an inverse model that provides physical device design candidates as the output of the model for a particular set of design requirements, without going through the iterative optimization process [15]. One main technical challenge exists in the training of inverse design model. While forward surrogate models represent a one-to-one mapping between input and output that produces a deterministic response for a given design, inverse models using DNNs often deal with one-to-many mapping problems, as shown in Fig. 1. If we build DNN models for inverse design in a similar manner as forward models by swapping the input and output data, they may not be able to converge in the training process, or simply fail to produce effective designs [16].

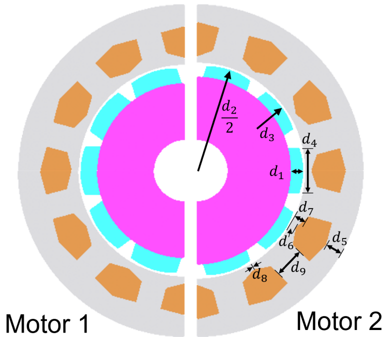
In this paper, we propose to use a tandem neural network as an inverse design strategy for electric machines, which avoids the one-to-many problem. We demonstrate the effectiveness of this technique by applying it to a surface-mount permanent magnet (SPM) motor design problem, and show that it can providing good motor designs for a set of design goals without iteration. The generated designs are validated with FEA simulations, and show good accuracy of the model, as well as the effectiveness of achieving pre-set design goals.

II. Inverse Design Process based on Tandem Neural Network

For motor design tasks, DNN-based surrogate models take in a motor design, represented in either a set of parameters or its cross-section image, as input, and output a prediction to one or more responses or performance metrics of the motor. However, a typical motor design task is considered an inverse design process: find the best motor design that meets the design goals. In order to utilize DNN for inverse design, we need to take motor design goals as input, and construct the DNN model such that it provides motor designs as its output. The forward surrogate model is a relatively straightforward one-to-one mapping between a motor design and its response, as shown in Fig. 1(a), while multiple motors with different design parameters may exhibit similar or even identical responses, as shown in Fig. 1(b). The training dataset may therefore contain "conflicting" data, where three designs D_1, D_2 and D_3 have the same responses R.

To be more specific, we illustrate the one-to-many mapping problem with SPM motor designs. The motor has 10 poles and 12 slots, and a total of 9 design parameters (shown in Fig. 2) are varied for design optimization, while other major geometrical parameter are fixed. Fig. 2 shows the comparison of the cross-sections of two motor design candidates. Although the two designs on the left (D_1) and right (D_2) side of the figure are distinct in design parameters, they have almost identical responses R in the categories we evaluate, including slot area r_1, 12^{th} Fourier order of cogging torque r_2, magnetic flux r_3, 1^{st} Fourier order of induced voltage r_4, and its total harmonic distortion r_5.

These "conflict" data pairs are the essential component of the one-to-many mapping. When we receive a target response R, there might be two (or even more) designs in the dataset, D_1 and D_2, that match the response. When R is provided as input, a deterministic model like the fully-connected neural network will only predict one design D^* based on the input. Now, because $D_1 \neq D_2$, the components in the loss function, i.e. $(D_1 - D^*)^2$ and $(D_2 - D^*)^2$, cannot



Motor parameters D

Unit: mm	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9
Motor 1	3.8	47.5	16	11	5	1.6	3	0	5.75
Motor 2	2.8	48.5	16	10.5	4	1.4	3	0	7

Motor response R

Unit:	r_1 (mm^2)	r_2 (N$\cdot$ m)	r_3 (Wb)	r_4 (V)	r_5 (1)
Motor 1	29.08	3.3×10^{-4}	0.0449	19.21	0.0324
Motor 2	29.12	5.2×10^{-5}	0.0448	19.17	0.0316

Fig. 2. Example of a one-to-many mapping: two different motor designs have the nearly identical responses. Figure shows half of cross-section magnetic design for the two motors side-by-side, with motor design parameters indicated in the figure, and exact values shown in the top right table. The responses of the two motors are shown in the bottom right table.

be minimized concurrently. As a result, back-propagation using the loss function cannot effectively optimize the neural network parameters. Although the designs D_1 and D_2 are different, we know that their responses are the same, i.e., $f(D_1) = R_1 = R = R_2 = f(D_2)$. Here, $f(\cdot)$ denotes the forward evaluation of the motor designs, which could be either provided by numerical simulation or a DNN-based surrogate model. With this in mind, an alternative approach to get around the problem is to evaluate the generated design first to get $R^* = f(D^*)$, then compare the responses of design candidates (R^*) to the target responses ($R_1 = R_2 = R$). This avoids the drawback of directly comparing the design candidate (D^*) with design candidates in the dataset (D_1 and D_2). This strategy is not only effective in avoiding the one-to-many mapping problem, but also makes sense in real-world design applications. Indeed, when a design goal is given, we do not restrict our design candidates to the motor designs readily available in the dataset; instead, any motor design that can meet the design goal is considered a good design. Additionally, by using this forward evaluation, there is no longer a conflict because $R_1 = R_2 = R$. Therefore, the loss can be simultaneously minimized for the two data points and the parameters in the DNN can be effectively updated using back-propagation.

Based on this concept, we construct a tandem neural network, whose schematic diagram is shown in Fig. 3. The tandem network comprises an inverse design model coupled to a pre-trained surrogate model. The surrogate model should be either a physics-based analytical model or a pre-trained DNN in order for the gradient of the loss to successfully back-propagate to the inverse design model. Only the response R is fed into the inverse model during training as the design goal. The design candidates D^* for the target response will then be generated by the inverse model, which is based on a fully-connected neural network. The design candidate will then be evaluated using the pre-trained surrogate model, and a predicted response R^* will be provided by the output layer. By comparing the target response R with the expected response R^*, a root-mean-squared error (RMSE) based prediction loss function $\mathcal{L}_P(R, R^*) = \sqrt{(R - R^*)^2}$ can be constructed. As a result, it avoids making a direct comparison between the ground truth parameters and the predicted parameters (in the intermediate layer), which could be unstable due to the one-to-many mapping problem. To obtain accurate response prediction for the generated design candidate, it is important to note that only the network parameters in the inverse design model are tuned in the back-propagation process, while the parameters in the pre-trained surrogate model are fixed during the training of the tandem network to ensure an accurate evaluation of these generated designs.

III. Model Implementation & Test Results

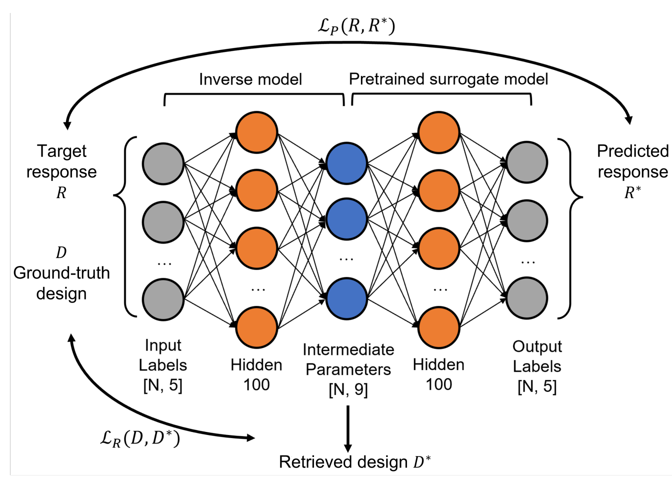


Fig. 3. Schematic of a tandem neural network.

To train and validate the performance of the tandem network, the first step is to generate a dataset. We used FEA simulations with JMAG to construct the SPM dataset, which involved evaluating a total of $8,916$ motor designs. Each design is characterized by a set of geometrical parameters, denoted by $D : [d_1, d_2, ..., d_9]$, and a corresponding set of motor responses, denoted by $R : [r_1, r_2, ..., r_5]$. The representation of each design parameter (d_i) can be found in Fig. 2, while the motor response (r_i) is described in the previous section. An 80:20 split ratio is used for the training and test data from the dataset.

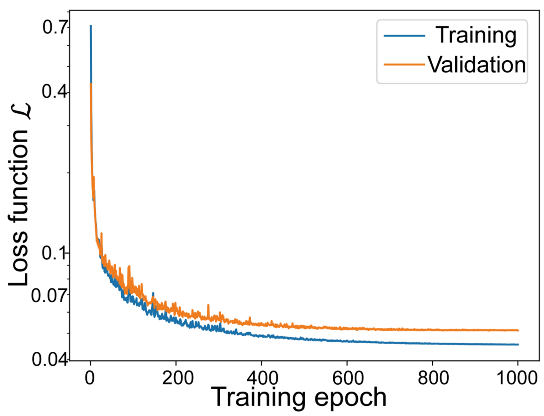


Fig. 4. Training loss and validation loss at different training epochs.

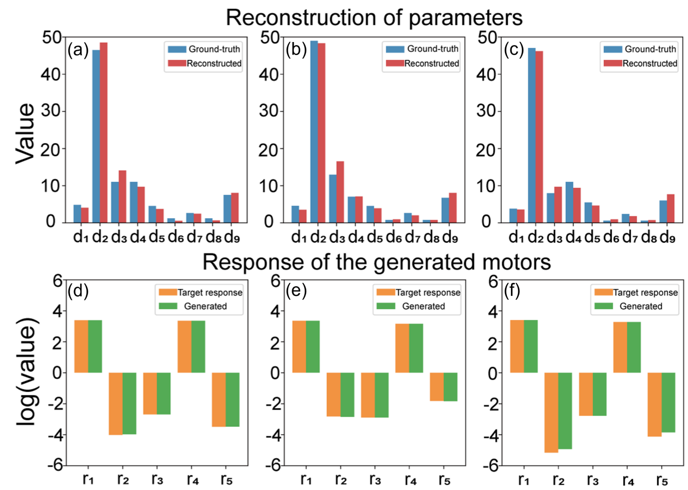


Fig. 5. (a-c) Reconstructed vs. ground-truth parameter of motors and (d-f) predicted vs. target motor responses for three motor design examples from the dataset.

The inverse design problem can be defined as follows: given a target response R (a five-dimensional vector), the model should provide a motor design candidate D, described by nine design parameters, such that the resulting response R^* equals the target response R. This problem is central to the performance validation of our tandem network model. It is important to highlight that the inverse problem we defined allows us to parameterize the motor designs. This is different from designs generated by methods such as topology optimization, which rely on image representations of motors. By using parameter descriptions, we can mitigate the manufacturing difficulty associated with testing designs in the real world. As a result, the motor designs generated by our tandem network model are most likely to be feasible in both physical and manufacturing considerations, provided that they satisfy some simple criteria regarding the parameters. We will discuss these criteria further in the following text.

We constructed the surrogate model for our study using Pyrenn, an open-source tool for deep neural network (DNN) implementation with the Levenberg-Marquardt (LM) algorithm [17], [18]. The LM algorithm is an iterative optimization method that alternates between Newton's method and stochastic gradient descent. It aims to efficiently update the neural network in each iteration to achieve the best possible performance. The surrogate model we created using Pyrenn consists of an input layer with 9 nodes that represent the input parameters. These nodes are fully connected to a hidden layer with 100 neurons, which are activated using the default $\tanh(\cdot)$ activation function. The output layer consists of 5 nodes, each representing a predicted response. We trained the surrogate model for 1000 epochs. By using Pyrenn with the LM algorithm, we were able to create a highly accurate and efficient surrogate model that can accurately predict the response of the motor design based on its parameters.

The inverse model is constructed and implemented in PyTorch, an open-source tool for DNN implementation. The model contains one input layer, one hidden layer with 100 neuron nodes, and one output layer. It is directly connected to a pretrained surrogate model which is also constructed with 100 nodes in a single hidden layer.

To ensure that the motor designs generated by our tandem network model are valid and feasible for manufacturing, we have included several criteria in the inverse design model. Firstly, all the design parameters must be non-negative. To enforce this condition, we have added a rectified linear unit (ReLU) layer at the end of the inverse model, which ensures that the output parameters are non-negative. Secondly, the slot opening of the motor should not be larger than the slot width. This means that the value of parameter d_7 should be greater than or equal to parameter d_6 (i.e., $d_6 \leq d_7$). Additionally, the magnets in the motor should not overlap with their neighboring magnets. This condition can be expressed as $(d_2 - 2 \cdot d_1) \cdot \sin(\pi/10) \geq d_4$. If a motor design violates either of these two constraints, a penalty term is added to the loss function, which effectively discards these design candidates. By enforcing these criteria, we can guarantee that the motor designs generated by the tandem network model are physically allowed and feasible for manufacturing.

Both the surrogate model and the inverse model in our study have been designed to be lightweight, which allows for efficient computation speed, especially when dealing with large amounts of parallel inverse design tasks. The tandem network model is trained with the training dataset. The test dataset is used to validate the model during training phase. Fig. 4 plots the training and validation error as function of training epoch during the training phase, which clearly show convergence of the loss function $\mathcal{L}$ toward the end of the training.

The trained model is then tested on the test dataset, with the responses of each motor design in the test dataset as the input of the inverse model. We can check the motor design parameter reconstructed by the inverse model, and compare them with the corresponding parameters associated with the input responses. We can also compare the response generated

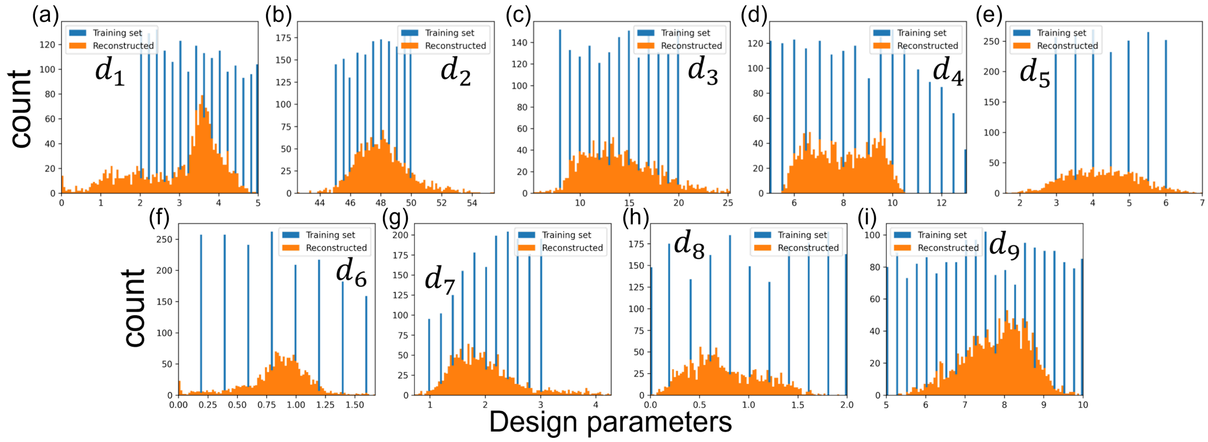


Fig. 6. The distribution of all parameters in the dataset (blue) and all reconstructed parameters from the tandem network with $w_r = 0.1$ (orange).

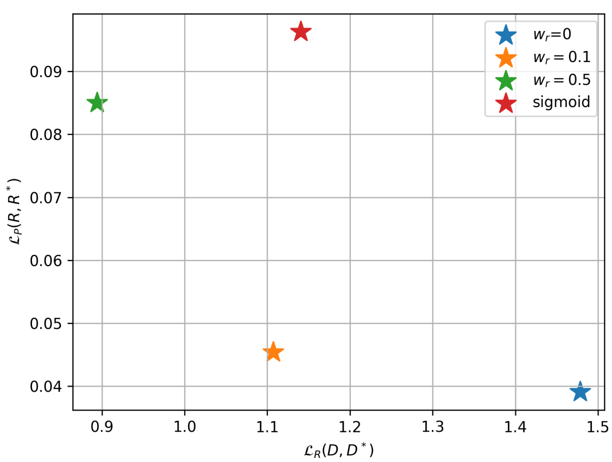


Fig. 7. Comparison of results from different loss functions and tandem network settings.

from the tandem network with the input target response. Three test examples are shown in Fig. 5, with Fig. 5(a)-(c) show the reconstruction capability of the inverse model, and Fig. 5 (d)-(f) show the responses of the generated motor designs as compared with the design target. The response plot in the second row are shown in logarithmic scale, since the order of different responses varied a lot.

While the main goal of the tandem network is to provide motor designs with minimal prediction loss, it is still necessary to evaluate the overall reconstruction loss on the test dataset, which compares generated designs and the available designs in the dataset. The main reason is that here we are using a DNN-based surrogate model which is trained based on the same dataset. We should be mindful that, the predicted responses are generated using the surrogate model, which also has some errors. The overall error of the inverse design model should consist of both the prediction error $\mathcal{L}_P$, and the mismatch between the actual response and the predicted response from the surrogate model (denote as $\mathcal{L}_S$). In the tandem network, only the first component $\mathcal{L}_P$ is evaluated. To account for the second error, ideally, we can refer to the previous section for the error of the parameter-based surrogate model. But the case is more complicated in our case: If we look at Fig. 6 which visualizes the distributions of the design parameters in the dataset (blue) and from the inverse model (orange), we can readily see that their range still have slight deviation, especially for d_1 and d_2 in 6(a)-(b). We may anticipate an even larger prediction loss of the surrogate model when it is used for the generated design parameters which follow the orange distribution if we keep in mind that the surrogate models were pre-trained with the same dataset following the blue distribution.

In this case, we can use another loss component $\mathcal{L}_S$ to evaluate the RMSE between the model prediction of the motor response and the one obtained from numerical simulation. Note that the surrogate models were pre-trained with the same dataset. A higher $\mathcal{L}_S$ is anticipated when it is used to evaluate the generated design parameters which lie outside of the parameter domain defined by the training dataset. Only with additional numerical simulations or a surrogate model trained on a separate dataset can this error be eliminated. However, we can positively correlate $\mathcal{L}_S$ with the reconstruction loss $\mathcal{L}_R$ which describes the difference between the reconstructed designs generated by the inverse model, and those in the training dataset as the RMSE $\mathcal{L}_R(D, D^*) = \overline{\sqrt{(D - D^*)^2}}$. It is therefore important to lower both $\mathcal{L}_R$ and $\mathcal{L}_P$ during the training phase.

A weighted loss function $\mathcal{L} = \mathcal{L}_P + w_r \cdot \mathcal{L}_R$ is then used in the training process, where w_r is the regularization weight on reconstruction loss $\mathcal{L}_R$. Several models are trained and evaluated with different w_r values. By changing the weighting factor w_r value, a trade-off is apparent: on the one hand, the generated parameters will have a distribution substantially different from the dataset if w_r is minimal, which lowers the

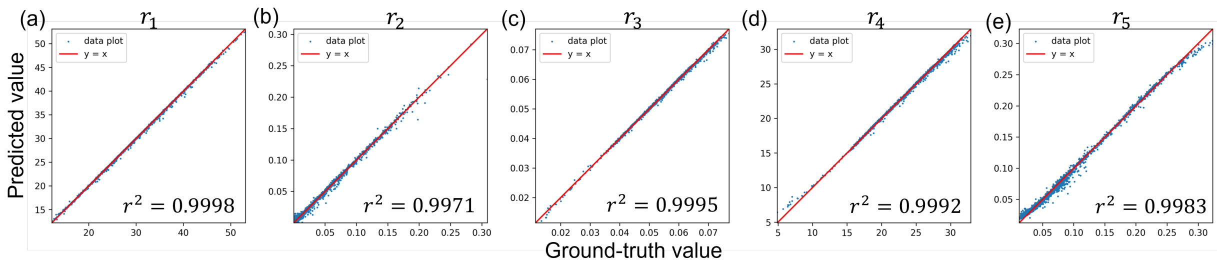


Fig. 8. The predicted responses of the generated design vs. the target responses, for $w_r = 0.1$. The responses are (a) slot area, (b) cogging torque, (c) magnetic flux, (d) induced voltage and (e) harmonic distortion.

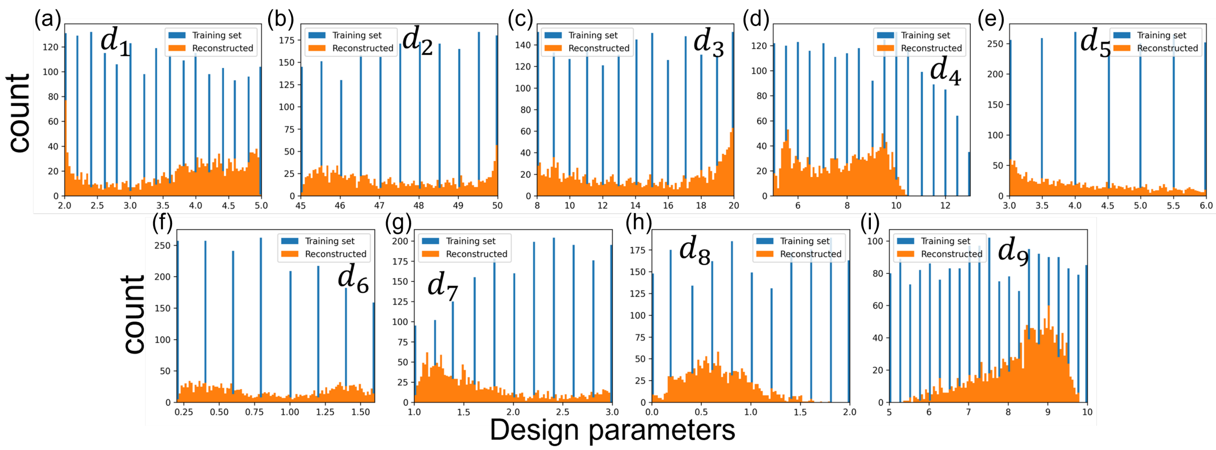


Fig. 9. The distribution of all parameters in the dataset (blue) and all reconstructed parameters from the tandem network with lower and upper constraints for the parameters (orange).

accuracy of the surrogate model; on the other hand, if the weight w_r is high, the inverse models will concentrate on obtaining the same design as the ground truth, which will exacerbate the one-to-many mapping problem. Therefore, we should balance the two loss terms with a suitable loss weight w_r. We may infer from Fig. 7 that it will be easier to reduce $\mathcal{L}_R$ and maintain an extremely low $\mathcal{L}_P$ simultaneously to achieve modest and reliable inverse design if the weight of the reconstruction loss is relatively low (between 0 and 0.1).

The test result of this model when $w_r = 0.1$ is shown in Fig. 8, which plots the predicted responses of the generated design from the inverse model R^* vs. the design goals R. Excellent matches are obtained for all 5 responses. This verifies that the performance of the inverse design is almost flawless. Based on the current results, we can see that the trained inverse model based on the tandem network is very effective in providing motor design for a given set of design goals.

In certain inverse design scenarios, the desired solution may be limited to a specific range, defined by lower and upper bounds. To address this requirement, we can make slight modifications to our inverse model by incorporating a sigmoid function at the output layer. This ensures that the generated designs adhere to the desired data range, as demonstrated in Fig. 9. However, the inclusion of additional constraints through this modification leads to an increase in prediction loss, which represents a trade-off between the desired range constraint and the overall performance of the inverse design model. Fig. 7 shows how all of these models are compared.

As previously mentioned, there may be unexpected errors when evaluating the inverse design with the surrogate model. To ensure accuracy, we conducted FEA simulations to evaluate the motors based on the inverse design model. In Fig. 10, we randomly selected 30 target responses that included all five metrics (r_1,...,r_5). We then used the tandem network proposed to obtain 30 motor design parameters from the inverse design. These 30 designs were simulated using FEA to obtain accurate true responses (as opposed to using only the surrogate model for prediction). The resulting true responses were plotted as green markers against the target responses. The r^2 value confirms that these motor designs from the tandem network are valid in exhibiting the user-defined target responses, albeit with a slightly larger error for the cogging torque (Fig. 10(b)).

It should be noted that the inverse model is not limited to a specific design problem. The method is generalizable to different design tasks as long as the data structure is similar, i.e., a relationship between design parameters and responses. For new design tasks, one can easily figure out the design parameters and corresponding response, generate a

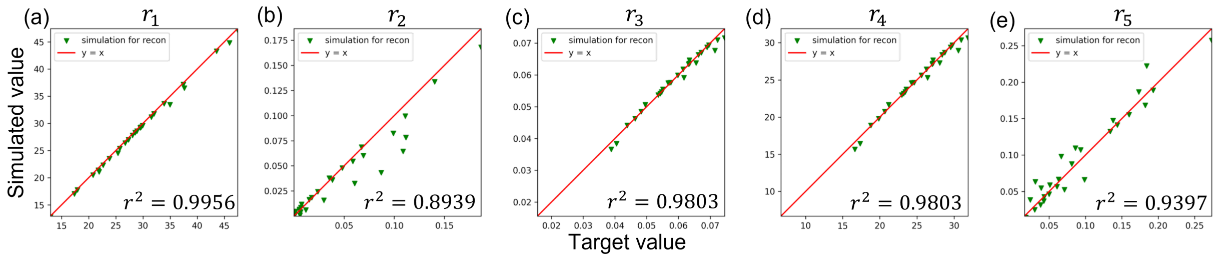


Fig. 10. Validation of the inverse designed motors with FEA simulation. The responses are (a) slot area, (b) cogging torque, (c) magnetic flux, (d) induced voltage and (e) harmonic distortion.

new training dataset, and retrain the same inverse model (or slightly modified model) to work for the task. Our experiments have demonstrated the accuracy of the inverse model, and its generalization to other design tasks is a promising direction for future research.

One downside of the tandem network is that, while it can bypass the one-to-many mapping problem and demonstrate excellent inverse design performance, it can only provide one solution for the target response, even though there are multiple solutions available due to the deterministic nature of the inverse model. In the future, we will evaluate other model architectures and develop deep generative models which can generate multiple motor design candidates for a given design target.

IV. Conclusions

In this paper, we proposed an inverse design strategy for electric machines using a tandem neural network architecture. Trained machine learning-based inverse design models have the potential to generate desired motor designs almost instantaneously and avoid the iterative optimization process with numerical simulations. One challenge in the inverse model is the one-to-many mapping problem, which creates problems in training neural networks based on back-propagation. We designed a tandem neural network, where an inverse model and a pre-trained surrogate model are combined to avoid the problem. We demonstrated the effectiveness of the method through the design of a surface-mount permanent magnet motor. Results show that the inverse model can effectively generate motor design candidates very close to the design target, as validated with FEA simulations. The proposed method can be generally applied to other motor design tasks.

References

[1] G. Bramerdorfer, J. A. Tapia, J. J. Pyrhönen, and A. Cavagnino, “Modern electrical machine design optimization: Techniques, trends, and best practices,” *IEEE Transactions on Industrial Electronics*, vol. 65, no. 10, pp. 7672–7684, 2018.

[2] Z. Liu, L. Raju, D. Zhu, and W. Cai, “A hybrid strategy for the discovery and design of photonic structures,” *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 10, no. 1, pp. 126–135, 2020.

[3] B. Zheng, J. Yang, B. Liang, and J.-c. Cheng, “Inverse design of acoustic metamaterials based on machine learning using a Gauss–Bayesian model,” *Journal of Applied Physics*, vol. 128, no. 13, p. 134902, 2020.

[4] Q. Zeng, Z. Zhao, H. Lei, and P. Wang, “A deep learning approach for inverse design of gradient mechanical metamaterials,” *International Journal of Mechanical Sciences*, vol. 240, p. 107920, 2023.

[5] W. Gerstner and W. M. Kistler, *Spiking neuron models: Single neurons, populations, plasticity*. Cambridge university press, 2002.

[6] W. Ma, Z. Liu, Z. A. Kudyshev, A. Boltasseva, W. Cai, and Y. Liu, “Deep learning for the design of photonic structures,” *Nature Photonics*, vol. 15, no. 2, pp. 77–90, 2021.

[7] Y. Xu, X. Zhang, Y. Fu, and Y. Liu, “Interfacing photonics with artificial intelligence: an innovative design strategy for photonic structures and devices based on artificial neural networks,” *Photonics Research*, vol. 9, no. 4, pp. B135–B152, 2021.

[8] S. Doi, H. Sasaki, and H. Igarashi, “Multi-objective topology optimization of rotating machines using deep learning,” *IEEE transactions on magnetics*, vol. 55, no. 6, pp. 1–5, 2019.

[9] H. Sasaki and H. Igarashi, “Topology optimization accelerated by deep learning,” *IEEE Transactions on Magnetics*, vol. 55, no. 6, pp. 1–5, 2019.

[10] A. Khan, V. Ghorbanian, and D. Lowther, “Deep learning for magnetic field estimation,” *IEEE Transactions on Magnetics*, vol. 55, no. 6, pp. 1–4, 2019.

[11] A. Khan, M. H. Mohammadi, V. Ghorbanian, and D. Lowther, “Efficiency map prediction of motor drives using deep learning,” *IEEE Transactions on Magnetics*, vol. 56, no. 3, pp. 1–4, 2020.

[12] Z. Pan, S. Fang, H. Wang, and Y. Zhong, “Accurate and efficient surrogate model-assisted optimal design of flux reversal permanent magnet arc motor,” *IEEE Transactions on Industrial Electronics*, 2022.

[13] B. Wang, A. Khaled Ahsan Talukder, and Y. Sakamoto, “Topological data analysis for image-based machine learning: Application to electric motors,” in *2022 International Conference on Electrical Machines (ICEM)*, 2022, pp. 1015–1021.

[14] A. K. A. Talukder, B. Wang, and Y. Sakamoto, “Electric machine two-dimensional flux map prediction with ensemble learning,” in *2022 25th International Conference on Electrical Machines and Systems (ICEMS)*, 2022, pp. 1–4.

[15] M. H. Tahersima, K. Kojima, T. Koike-Akino, D. Jha, B. Wang, C. Lin, and K. Parsons, “Deep neural network inverse design of integrated photonic power splitters,” *Scientific reports*, vol. 9, no. 1, p. 1368, 2019.

[16] D. Liu, Y. Tan, E. Khoram, and Z. Yu, “Training deep neural networks for the inverse design of nanophotonic structures,” *Acs Photonics*, vol. 5, no. 4, pp. 1365–1369, 2018.

[17] S. Ide, “Striations, duration, migration and tidal response in deep tremor,” *Nature*, vol. 466, no. 7304, pp. 356–359, 2010.

[18] D. Mardquardt, “An algorithm for least square estimation of parameters,” *J. Soc. Ind. Appl. Math*, vol. 11, pp. 431–441, 1963.