Using Machine Learning to Reduce Design Time for Permanent Magnet Volume Minimization in IPMSMs for Automotive Applications

Yuki Shimizu∗, a) Student Member, Shigeo Morimoto∗, Senior Member, Masayuki Sanada∗, Senior Member, Yukinori Inoue∗, Member

(Manuscript received April 12, 2021, revised June 10, 2021)

Interior permanent magnet synchronous motors (IPMSMs) have been widely used as traction motors for electric vehicles. Finite element analysis is commonly used to design IPMSMs but is highly time-intensive. To shorten the design period for IPMSMs, various surrogate models have been constructed to predict relevant characteristics, and they have been used in the optimization of IPMSM geometry. However, to date, no surrogate models have been able to accurately predict the characteristics over the wide speed range required for automotive applications. Herein, we propose a method for accurately predicting the speed-torque characteristics of an IPMSM by using machine learning techniques. To improve the prediction accuracy, we set the motor parameters as the prediction target of the machine learning methods. We then used the trained surrogate model and a real-coded genetic algorithm to minimize the volume of the permanent magnet and showed that the design time can be significantly reduced compared with the case where only finite element analysis is used.

Keywords: interior permanent magnet synchronous motor (IPMSM), machine learning, surrogate model, support vector regression, XGBoost, real-coded genetic algorithm (RCGA)

1. Introduction

In recent years, interior permanent magnet synchronous motors (IPMSMs) have been widely adopted as traction motors for electric vehicles due to their high power, efficiency, and reliability [1,2]. Because IPMSMs have permanent magnets (PMs) embedded in their rotor core, their design has a high degree of freedom. To design IPMSMs, finite element analysis (FEA) is commonly used, but this method is highly time-intensive. Thus, IPMSMs face the problem of a long development period, and an automatic design system is needed that can efficiently design high-performance IPMSMs in a shorter period of time [3].

Many studies have aimed to reduce the computation time necessary for the design process by constructing surrogate models that can be used to accurately predict the motor characteristics for structural optimization while minimizing or eliminating the need for FEA. These papers report various methods for building surrogate models, such as the response surface method [4,5], radial basis function networks [6], convolutional neural networks [7,8], and transfer learning [9]. However, these studies assumed only a fixed-point driving condition and did not examine the characteristics of the wide operation ranges required for traction motors in automotive applications, which are the subject of this study.

In this paper, we propose a surrogate model construction method using supervised learning that can accurately predict the speed-torque characteristics of an IPMSM for automotive applications. The training data for our models are the geometries of motors randomly generated based on a previously proposed double-layered IPMSM [10], and their characteristics are obtained via time-consuming FEA. Therefore, considering the analysis time, the smaller the number of training data, the better. To build a practical surrogate model from a small number of data, this research is based on the following three perspectives.

• Selection of the prediction target; we set the motor parameters, PM flux linkage and d- and q-axis inductances as the indirect targets for the speed-torque characteristics prediction, instead of attempting to predict the torque and speed directly.

• Selection of the learning method; we compare three machine learning methods, namely, ridge regression (RR) [11], support vector regression (SVR) [12], and XGBoost [13], and select the most accurate method for each parameter.

• Definition of the applicability domain; we set the applicability domain of the surrogate models based on the density of the training data and optimize the design within that domain.

The proposed method based on the above ideas has the following features.

• Because motor characteristics can be predicted in a much shorter time than with FEA, a large-scale shape optimization design can be performed.

• Once surrogate models are built, they can be used many times.

To demonstrate these advantages, the trained surrogate
models and a real-coded genetic algorithm (RCGA) were used to minimize the PM volumes. The results of the PM volume minimization design show that our surrogate models can maintain high accuracy through the design optimization and that the design time can be significantly reduced compared with using only FEA. This paper is a revised version of the previously presented proceeding. 

2. Prediction Method

2.1 Ridge Regression

RR is a type of linear regression analysis, and the output \( f(\mathbf{x}) \) is expressed as follows \(^{(15)}\):

\[
 f(\mathbf{x}) = \mathbf{x}^\top \mathbf{w} + b, \tag{1}
\]

where \( \mathbf{x} \in \mathbb{R}^d \) is the input variable in \( d \) dimensions, \( \mathbf{w} \in \mathbb{R}^d \) is the coefficients, and \( b \in \mathbb{R} \) is the constant term. The loss function of RR is defined as adding the L2 regularization term to the least-squares loss function. Therefore, using the training data \( \{ (\mathbf{x}_i, y_i) \}_{i=1}^n \), the coefficients can be obtained by solving the optimization problem with equation (2).

\[
 \min_{\mathbf{w}, b} \frac{1}{2} \sum_{i=1}^n (f(\mathbf{x}_i) - y_i)^2 + \frac{\lambda}{2} \| \mathbf{w} \|^2. \tag{2}
\]

where \( \lambda > 0 \) is the regularization parameter that governs the relative importance of the regularization term compared with the sum-of-squares error term.

2.2 Support Vector Regression

SVR is a method that applies a support vector machine to regression analysis \(^{(15)}\). The output \( f(\mathbf{x}) \) of nonlinear SVR is expressed as follows.

\[
 f(\mathbf{x}) = \phi(\mathbf{x})^\top \mathbf{w} + b, \tag{3}
\]

where \( \phi \) is a certain nonlinear function. In SVR modeling, we consider the following coefficient optimization problem using the training data \( \{ (\mathbf{x}_i, y_i) \}_{i=1}^n \).

\[
 \min_{\mathbf{w}, b} \sum_{i=1}^n E(\mathbf{f}(\mathbf{x}_i) - y_i) + \frac{\lambda}{2} \| \mathbf{w} \|^2, \tag{4}
\]

where

\[
 E(\mathbf{f}(\mathbf{x}_i) - y_i) = \max(0, |\mathbf{f}(\mathbf{x}_i) - y_i| - \varepsilon). \tag{5}
\]

Although minimizing the first term of the loss function in Eq. (4) can accurately predict the training data, the possibility exists that the generalization performance may be degraded due to overfitting. For this reason, a regularization term is included as in ridge regression. The regularization parameter \( C > 0 \) governs the relative importance of the regularization term compared with the error term, and the parameter \( \varepsilon > 0 \) is the width of the dead zone for prediction errors.

By solving equation (4), the estimate in equation (3) can be expressed as in equation (6).

\[
 f(\mathbf{x}) = \sum_{i=1}^n a_i \mathbf{K}(\mathbf{x}, \mathbf{x}) + b, \tag{6}
\]

where

\[
 \mathbf{K}(\mathbf{x}, \mathbf{x}) = \phi(\mathbf{x})^\top \phi(\mathbf{x}). \tag{7}
\]

The value of \( a_i \) is determined from the optimization problem in equation (4). In this study, we use a Gaussian kernel as the kernel function \( K \).

\[
 K(\mathbf{x}, \mathbf{x}) = \exp(-\gamma \| \mathbf{x} - \mathbf{x} \|^2), \tag{8}
\]

where \( \gamma > 0 \) is a hyperparameter that adjusts the nonlinearity of the SVR model.

2.3 XGBoost

XGBoost is a gradient boosting method proposed by Chen and Guestrin \(^{(16)}\), which uses decision trees as weak learners. The learning process of XGBoost is described below.

First, the first decision tree is adapted from the input data. Next, a second decision tree is fitted to reduce the residuals of the first one. In this way, when \( K \) decision trees are constructed, the predicted value \( f(\mathbf{x}) \) is expressed by equation (9) with the \( k \)-th decision tree as \( f_k \).

\[
 f(\mathbf{x}) = \sum_{k=1}^K f_k(\mathbf{x}), \tag{9}
\]

When constructing the \( t \)-th decision tree from \( t-1 \) decision trees, the following optimization problem is solved.

\[
 \min_{f_t} \sum_{i=1}^n \left( \gamma \left( f_t(\mathbf{x}_i) - y_i \right)^2 + \frac{1}{2} \beta \| \mathbf{w} \|^2 \right), \tag{10}
\]

where \( \gamma \left( f_t(\mathbf{x}_i) - y_i \right)^2 \) is the prediction by the \( t \)-th decision trees, and the loss function \( l \) uses the root-mean-square error. \( \Omega \) is the regularization term defined in equation (11).

\[
 \Omega(\mathbf{w}) = \gamma T + \frac{1}{2} \beta \| \mathbf{w} \|^2, \tag{11}
\]

where \( T \) is the number of leaf nodes in the decision tree, \( \mathbf{w} \in \mathbb{R}^T \) is the value stored in the leaf node, and \( \gamma, \beta > 0 \) are hyperparameters.

3. Training Dataset Generation

3.1 Motor Parameters to be Predicted

In this study, we considered driving a motor under various operating conditions. To improve the robustness of predictions for the current vector conditions, the torque and speed are not set as direct targets. Instead, the PM flux linkage, \( \Psi_a \), and \( d \)- and \( q \)-axis inductances, \( L_d \) and \( L_q \), respectively, are estimated from the current conditions and geometrical parameters. These motor parameters for the training datasets are calculated from the flux linkage obtained from two-dimensional FEA and Eqs. (12)–(14).

\[
 \Psi_a = \Psi_a \cos \alpha, \quad (\beta = 0^\circ) \tag{12}
\]

\[
 L_d = \frac{\Psi_d \cos \alpha - \Psi_a}{i_d}, \tag{13}
\]

\[
 L_q = \frac{\Psi_d \sin \alpha}{i_d} = \frac{\Psi_d \sin \alpha}{\Psi_a \cos \beta}. \tag{14}
\]

where \( \Psi_a \) is the magnitude of the flux linkage vector, \( \alpha \) is the leading angle of the flux linkage vector from the \( d \)-axis, \( I_a \) is the magnitude of the current vector, \( \beta \) is the leading angle of the current vector from the \( q \)-axis, and \( i_d \) and \( i_q \) are the \( d \)- and \( q \)-axis currents, respectively. The PM flux linkage \( \Psi_a \) is calculated at \( \beta = 0^\circ \) and is assumed to depend only on the current amplitude and not on the current phase angle \( \beta \). Because this study does not consider the harmonics, we calculated the
average values of the motor parameters for one electric angle cycle.

The average torque $T$ and limit speed $N_{lim}$ are obtained by substituting the calculated motor parameters into Eqs. (15) and (16).

$$T = P_n \left( \Psi d I_a \cos \beta + \frac{1}{2} \left( L_q - L_d \right) I_q^2 \sin 2\beta \right), \cdots \cdots (15)$$

$$N_{lim} = \frac{V_{am} - R_d I_a}{\sqrt{\left( \Psi d - L_d I_a \sin \beta \right)^2 + \left( L_q I_a \cos \beta \right)^2}}, \cdots \cdots (16)$$

where $P_n$ is the number of pole pairs, $V_{am}$ is the maximum terminal voltage, and $R_d$ is the winding resistance.

### 3.2 Generating Structures and Geometric Constraints

Figure 1 shows the rotor structure and geometrical parameters for the IPMSMs to be generated, where ($r_1$, $\theta_1$) are the polar coordinates with the center of the shaft as the origin, and the geometrical parameter vector is defined as $x_{geom} = (r_1, \theta_1, d_1, d_2, \ldots, d_9)^T \in \mathbb{R}^{11}$. The geometrical parameters are defined based on the rotor geometry of the 8-pole, 48-slot IPMSM previously proposed by the authors. All variables not included in $x_{geom}$ or dimensions not shown in Fig. 1 are either constant or automatically determined. The arc shape at the end of the flux barrier has been removed for simplicity.

The upper and lower limits of the geometrical parameters were set with inter-variable dependencies to prevent the generation of un-designable shapes (e.g., PMs protruding from the rotor area). The rotor shapes for the training datasets were randomly generated according to a uniform distribution.

Table 1 lists the upper and lower limits of the geometrical parameters, where $R_e$ is the outer diameter of the rotor and $d_1$, $d_{31}$, $d_{51}$, and $d_9$ are defined by Eq. (17). Additionally, the directions of magnetization of the PMs are fixed in the thickness direction.

$$d_c = 2(R_e - d_5) \sin \left( \frac{\theta_1}{2} \right),$$

$$d_{31} = (R_e - d_3) \sin \theta_1 - (2d_1 + 0.6) \cos \theta_1, \cdots \cdots (17)$$

$$d_{51} = d_{c2} - (d_5 - d_4) - 0.6,$$

$$d_{32} = d_{c1} - 0.8(d_5 - d_4) - 0.6.$$

### 3.3 Training Data Generation and Analysis

In this section, we explain how we generated the training data. Figure 2 shows an example of the relationship between the amount of training data and the prediction accuracy. The number of test dataset was fixed at 400, and SVR was used as the machine learning method. As shown in Fig. 2, the more training data used to train the surrogate model, the more the prediction accuracy of the surrogate model improved, although the gradient gradually decreases. However, the smaller the amount of training data, the better when it comes to design time because the training data are generated by time-consuming FEA. Therefore, it is necessary to train the surrogate model with the appropriate amount of training data for each target parameter.

As mentioned in Sec. 3.1, only the PM flux linkage is independent of the current phase angle $\beta$ among the three motor parameters, and the input vector for its prediction model is one dimension smaller than that for prediction of the $d$- and $q$-axis inductance models. Therefore, the number of datasets required for training the surrogate model for the PM flux linkage is also smaller. Thus, if the amount of training data is set to be variable according to the motor parameters, we can reduce the total analysis time.

Based on these considerations, we examined a method for efficiently generating training data using FEA. The generation and analysis process for the training data is shown in Fig. 3 and is explained as follows:

1. For the PM flux linkage, according to Eq. (18), we randomly generate 2,000 cases of the phase current, $i_{d1}$.
From the analysis results, we predict the PM flux linkage $\Phi_a$ according to Eq. (18).

2. Compute $\Phi_a$ using Eq. (12) and train the surrogate model.

3. Generate current conditions and shapes of FEA for $L_d$ and $L_q$ according to Eq. (19).

4. Predict $\Phi_a$ and compute $L_d$ using Eq. (13).

4. Remove the outliers of the $L_d$ using the Hampel identifier method (Eq. (21)).

4. Compute $\Phi_a$ using Eq. (14).

5. Train the surrogate models for $L_d$ and $L_q$.

Fig. 3. Generation and analysis flowchart for the training data.

$I_e$ and the geometrical parameters, then perform two-dimensional FEA under the condition $\beta = 0^\circ$.

\[ \begin{align*}
I_e & \sim U(0, 140) \text{ (Arms)}, \\
\mathbf{x}_{\text{geom}}^{(i)} & \sim U(\mathbf{x}_{\text{invr}}^{(i)}, \mathbf{x}_{\text{upr}}^{(i)}) (j = 1, \ldots, 11), \\
\end{align*} \tag{18} \]

where $U(a,b)$ is a random variable with a uniform distribution on an open interval $(a,b)$, $\mathbf{x}_{\text{geom}}^{(i)}$ is the $j$-th geometrical parameter, and $\mathbf{x}_{\text{invr}}^{(i)}, \mathbf{x}_{\text{upr}}^{(i)}$ are its upper and lower bound, respectively.

2. From the analysis results, we compute the training data for PM flux linkages using Eq. (12) and train the model. The input vector for the model is defined as $\mathbf{x}_r = (I_e, r_1, \theta_1, d_1, d_2, \ldots, d_{13})^T \in \mathbb{R}^{12}$, which combines the current with the geometrical parameters.

3. For the $d$- and $q$-axis inductances, we randomly generate 6,000 cases of the phase current, the current phase angle and the geometrical parameters according to Eq. (19), then perform a two-dimensional FEA.

\[ \begin{align*}
I_e & \sim 140 \sqrt{U(0, 1)} \text{ (Arms)}, \\
\beta & \sim U(90, 0) \text{ (\degree)}, \\
\mathbf{x}_{\text{geom}}^{(i)} & \sim U(\mathbf{x}_{\text{invr}}^{(i)}, \mathbf{x}_{\text{upr}}^{(i)}) (j = 1, \ldots, 11), \\
\end{align*} \tag{19} \]

where the inverse transform method is used to generate the phase currents so that the current conditions are generated uniformly within the current limit circle in $i_d$-$i_q$ coordinates (see Fig. 4).

4. From the analysis results, we predict the PM flux linkages in each case using the surrogate model trained in Step 2 and compute the $d$- and $q$-axis inductances using Eqs. (13) and (14), respectively. Here, as shown in Eq. (20), the error in predicting the PM flux linkage remains in the computed $d$-axis inductance $L_d$.

\[ \begin{align*}
\tilde{L}_d & = \Phi_a \cos \alpha - \tilde{\Phi}_a \\
& = \frac{\Phi_a \cos \alpha - (\Phi_d + \epsilon_{\Phi_a})}{i_d} = L_d - \frac{\epsilon_{\Phi_a}}{i_d} \\
\end{align*} \tag{20} \]

where $\tilde{\Phi}_a$ is the predicted value of $\Phi_a$, and $\epsilon_{\Phi_a}$ is the prediction error in $\Phi_a$. In the case of small $i_d$, the error becomes too large to be non-negligible with high probability, so we remove the outliers of the computed $d$-axis inductances using the Hampel identifier method. The upper and lower thresholds of the Hampel identifier are given by

\[ \text{med}(L_d^{(i)}) \pm 3 \times 1.4826 \times \text{med}(|L_d^{(i)} - \text{med}(L_d^{(i)})|), \]

\[ \text{med}(\cdot) \text{ is a function that gives the median value of the data. Comparing the computed } d \text{-axis inductances with the thresholds, we remove the data with values higher than the upper threshold or lower than the lower threshold as shown in Fig. 5.} \]

5. We train the models to predict the $d$- and $q$-axis inductances from the calculation results. The input vectors are defined as $\mathbf{x}_L = (i_d, i_q, r_1, \theta_1, d_1, d_2, \ldots, d_{13})^T \in \mathbb{R}^{13}$, which combines the current vector with the geometrical parameters.

JMAG-Designer 16.1 was used as the analysis software, and the analysis times for Steps 1 and 3 were 18.1 and 58.5 hours, respectively. In Step 4, 76 datasets were removed by the Hampel identifier method, and 5,924 datasets were used to train the model for the prediction of the $d$-axis inductance. The analysis datasets were divided into training datasets and test datasets in a 4:1 ratio, and only the training datasets were used for training. All input and output data were standardized before training. Grid search with cross-validation was used for the RR and SVR methods to optimize the hyperparameters, while Bayesian optimization with the Optuna library was used for XGBoost.

4. Results of Motor Characteristics Prediction

4.1 Motor Parameter Prediction

Figure 6 shows the prediction results of each method, where the training datasets
Machine Learning to Reduce Design Time for IPMSMs  
(Yuki Shimizu et al.)

Fig. 6. Prediction results for each predictive method, where $r^2$ is the coefficient of determination for the $d$-axis inductance were calculated using the prediction results for the PM flux linkage by SVR. Plot points that fall on the solid black line indicate perfect prediction of the analysis results.

First, comparison of the prediction results for the PM flux linkage shows that RR, which is a linear regression model, was not able to represent the nonlinearity of the PM flux linkage, and consequently, its prediction accuracy was the lowest. Although XGBoost showed good accuracy for the training dataset, its prediction accuracy for the test dataset was relatively low, and there was a tendency to overfit. In contrast, SVR was found to be highly accurate for both the training and test datasets.

Next, a comparison of the $d$-axis inductance prediction results shows that all predictive methods have some plots where the predicted values deviate significantly from the analyzed values. This is because the training data include the prediction error in the PM flux linkage, as described for Eq. (20). In particular, XGBoost predicts the training data with high accuracy, but the accuracy for the test data is lower than that for the training data. This is because XGBoost learned perfectly even the prediction errors of the PM flux linkages, thereby losing its robustness to unknown test data. Conversely, SVR was stable and accurate even for unknown data, and was thus found to be the most suitable prediction method. The RR plots were curved against the solid black line. This is because the inductance shows a nonlinear (curve-like) change, especially with respect to the current, while the RR can only predict the change linearly.

Finally, a comparison of the $q$-axis inductance prediction results shows that XGBoost was the most accurate among the three predictive methods for both the training and test data. SVR was not able to accurately predict the inductance variation due to the relaxation of magnetic saturation in the low current range.

4.2 Speed-Torque Characteristics Prediction

Next, we compare the prediction results for the speed-torque characteristics. Figure 7 shows the rotor shapes used for the prediction of the speed-torque characteristics, Fig. 8 shows the prediction results for the speed-torque characteristics at maximum power control, and Table 2 lists the root mean square errors (RMSEs) for torque prediction. The conventional rotor shape is the previously proposed one (10), and for the upper and lower limit models, the geometrical parameters were set to the upper and lower limits in order from $r_1$ according to Table 1. SVR was used to predict $\Psi_a$ and $L_d$, and XGBoost was used to predict $L_q$. The direct prediction results for speed and torque were both obtained using SVR, which we trained with the same analysis results as those used to train the models for predicting $L_d$ and $L_q$.

The prediction errors for torque at each speed show that the proposed method with the motor parameter prediction improves the prediction accuracy over a wide range of currents compared with the direct prediction method. This is because there can be a point where the denominator of the limit speed becomes zero within the operating area, as shown in Eq. (16), and hence the nonlinearity of the limit speed for the shape and current vector is much stronger than that of the motor parameters. In particular, the prediction errors of the upper and lower limit models are larger than that of the conventional model. The reason is that the data density is low around the upper and lower bounds of the geometrical parameters and the prediction accuracy tends to decrease. Nevertheless, the proposed method maintains high prediction accuracy for the upper and lower bound models, indicating that the proposed method is robust to input variables.
Machine Learning to Reduce Design Time for IPMSMs (Yuki Shimizu et al.)

Fig. 7. Rotor shapes used for the speed-torque characteristics prediction

Fig. 8. Prediction results for speed-torque characteristics of phase current limits $I_{em}$ of 134 A and 30 A

Table 2. RMSEs of torques for prediction of speed-torque characteristics

Table:  
| Method of directly predicting the torque and speed | Proposed method with motor parameter prediction |
|--------------------------------------------------|-----------------------------------------------|
| Proposed method                                  | RMSE (Nm)                                     |
| $I_{em} = 134$ A                                 | 3.88                                          |
| $I_{em} = 30$ A                                  | 1.82                                          |
| $I_{em} = 134$ A                                 | 12.14                                         |
| $I_{em} = 30$ A                                  | 5.65                                          |

Fig. 9. Computation time histogram of the speed-torque characteristics prediction

4.3 Computation Time Finally, we discuss the time it takes to predict the characteristics. Figure 9 shows the histogram of the computation time of the speed-torque characteristics by the proposed method, where mean and std are the mean and standard deviation of the measurements, respectively. The horizontal axis shows the average computation time of the speed-torque characteristics for 10 individuals, and this histogram shows the results of 200 times of this average time computation. Because it was 12.4 minutes required for FEA of the speed-torque characteristics of the conventional model in Fig. 8, the surrogate models could reduce the characteristic computation time to 0.39%.

5. Permanent Magnet Volume Minimization Using Surrogate Models

To demonstrate the time advantage of the surrogate model constructed in the previous section, we minimized the PM volume of the double-layered rotor shape with the most accurate surrogate models, i.e., SVR for $P_a$ and $L_d$, and XGBoost for $L_q$, and RCGA.

5.1 Constraints The optimization algorithm considers three constraints: whether the geometry is designable, whether the motor can be driven at the required operating points, and whether the geometrical parameters are within the applicability domains of the surrogate models.

First, we used the upper and lower bound constraints in Table 1 for the geometric constraints. If an individual (design) violating this constraint is generated by real-coded crossover, we can choose the closest feasible solution in the geometrical parameter space as the correct solution. Because all the geometric constraints are imposed separately on each parameter, we used the following simple equation:

$$x_{\text{modified}}^{(i)} = \begin{cases} x_{\text{upr}}^{(i)} & (\text{if } x_c^{(i)} > x_{\text{upr}}^{(i)}), \\ x_{\text{lwr}}^{(i)} & (\text{if } x_c^{(i)} < x_{\text{lwr}}^{(i)}), \\ x_c^{(i)} & (\text{other}), \end{cases} \quad (22)$$
where $x_{\text{modified}}^{(j)}$ is the modified geometrical parameter and $x_{\text{original}}^{(j)}$ is the $j$-th geometrical parameter of the children generated by a real-coded crossover.

For the required operating points, we used the penalty function method. The amount of undelivered torque for $p$ required operating points $(N_{\text{req}}^{(i)}, T_{\text{req}}^{(i)})$ was calculated using Eq. (23), and this penalty was imposed on the fitness.

$$
P_T(x_{\text{geom}}) = \sum_i \max(0, T_{\text{req}}^{(i)} - T_{\text{pred}}^{(i)}(x_{\text{geom}})), \cdots (23)$$

where the torque prediction $T_{\text{pred}}^{(i)}$ is given as a result of maximum power control at each speed condition $N_{\text{req}}^{(i)}$ as

$$
T_{\text{pred}}^{(i)}(x_{\text{geom}}) = \max_{I_a \in (0, I_{\text{max}})} T_{\text{sur}}(I_a, \beta, x_{\text{geom}})
$$

s.t. $N_{\text{sur}}(I_a, \beta, x_{\text{geom}}) = N_{\text{req}}^{(i)}, \cdots (24)$

in which $I_{\text{max}}$ is the maximum armature current, and $T_{\text{sur}}$ and $N_{\text{sur}}$ are the torque and limit speed predicted by the surrogate models, respectively. The solution for maximum power control was obtained by brute-force search. The required drive points were set from the operating area of the reference motor in Ref. (10) to two points, $P_A (3,000 \text{ min}^{-1}, 197 \text{ Nm} \times 1.03)$ and $P_B (11,000 \text{ min}^{-1}, 40 \text{ Nm} \times 1.03)$, where a prediction error of 3% was taken into account.

The applicability domain of the surrogate models was also expressed using the penalty function method. The applicability domain is the domain where the prediction accuracies of the surrogate models are guaranteed\(^{(17)}\). In general, the torque decreases as PM volume is reduced, so the optimization algorithm with the surrogate model tends to converge to the region where the torque is overestimated by the surrogate model. Therefore, to prevent convergence to a region with low prediction accuracy, we set a constraint about the applicability domain.

In this study, the applicability domain was set from the training data density evaluated by a one-class support vector machine (OCSVM)\(^{(18)}\). All the training data (geometrical parameters only) were used for training the OCSVM, and the hyperparameter was set to produce a false-positive rate of 31.7%. The Gaussian kernel was used as the kernel function, and its hyperparameter was determined to maximize the variance of the Gram matrix. In the optimization design, to impose a penalty only when children are generated outside the applicability domain (i.e., the output of OCSVM is negative), the penalty function was defined as follows:

$$
P_{\text{AD}}(x_{\text{geom}}) = \max(0, -f_{\text{OCSVM}}(x_{\text{geom}})), \cdots (25)$$

where $f_{\text{OCSVM}}$ is the output of the OCSVM.

### 5.2 Objective and Parameters

Given the constraints in the previous section, the PM volume minimization can be achieved by searching for the solution of the unconstrained optimization problem represented by the following:

$$
\min_{x_{\text{geom}}} \left\{ \text{fitness} = \frac{V}{V_{\text{init}}} + P_T + P_{\text{AD}} \right\}, \cdots \cdots (26)
$$

where $V$ is the PM volume of each individual, as normalized by the PM volume of the conventional shape $V_{\text{init}}$ (100 cm\(^3\)).

Figure 10 shows a flowchart of the PM volume minimization process. UNDX-m\(^{(18)}\) was used as the real-coded crossover, and JGG\(^{(18)}\) was used as the generation alternation model. The population size in each generation was set to 77, and the number of children was set to 66. The algorithm was terminated when the evaluation value was not updated by more than 0.001 compared to the best individual for 50 consecutive generations or the number of generations reached 200.

### 5.3 Results and Validation of Computation

To evaluate the robustness of the algorithm, we ran the optimization design five times. Figure 11 shows the transition of the PM volumes for all the populations in the first optimization design. The best individuals of each population were determined by the fitness function in Eq. (26). From this transition, it can be seen that although the variance of the shapes is large in the initial generation, in subsequent generations, shapes with almost the same PM volume are generated due to the torque and applicability domain constraints. It is also found that, after the 20th generation, the individual with the smallest PM volume was selected as best, which implies that the penalty functions for all individuals are zero.

Figure 12 compares the prediction and FEA results for the
Machine Learning to Reduce Design Time for IPMSMs (Yuki Shimizu et al.)

Fig. 12. Prediction and FEA results for speed-torque characteristics of the best individual at the first optimization design

Table 3. Torque prediction results of the best individuals at the required drive points

| No. | Torque at 3000 min⁻¹ (P_A) | Torque at 11000 min⁻¹ (P_A) |
|-----|-----------------------------|-----------------------------|
|     | Pred. (Nm) | FEA (Nm) | Error (%) | Pred. (Nm) | FEA (Nm) | Error (%) |
| 1st | 204.5     | 197.9    | 3.3       | 50.1      | 48.9     | 2.6       |
| 2nd | 204.1     | 199.8    | 2.2       | 52.4      | 51.0     | 2.8       |
| 3rd | 203.5     | 198.9    | 2.8       | 51.0      | 49.5     | 3.0       |
| 4th | 203.5     | 198.1    | 2.7       | 50.7      | 48.4     | 4.7       |
| 5th | 203.1     | 201.8    | 0.7       | 51.0      | 50.4     | 1.1       |

Fig. 13. Rotor shapes of the best individuals

Next, to discuss the effect of the choice of machine learning method on the optimization results, we also examined the cases of using only SVR and only XGBoost for the three motor parameter predictions. Table 4 compares the torque prediction results for the optimal shape when only SVR and XGBoost were used, and Fig. 14 shows the rotor shapes of the best individuals. The comparison results showed that the prediction accuracy of the cases of only SVR and only XGBoost is low at the maximum torque, and neither of them satisfies the required point P_A. This result implies the importance of choosing the best learning method for each parameter.

Finally, we compare the total computation time of the proposed method with FEA. Here, optimal design with FEA in the problem setting of this study requires several months or more and is difficult to perform in practice. Therefore, we estimated the optimal design time from the time required for FEA of the conventional shape in Fig. 8 and compared the design time. Table 5 shows a comparison of the computation times between the proposed method and FEA-only optimization design, where the time required for FEA of the conventional shape was 12.4 minutes. A comparison of the total computation time shows that the total computation time of the proposed method using machine learning can be reduced to 2.9–6.9% compared to the computation time for only FEA, even including the time for FEA of the training dataset.

6. Conclusion

In this paper, we proposed a surrogate model construction method for accurately predicting the speed-torque characteristics of double-layered IPMSMs and then showed that our surrogate models could significantly reduce the design time for PM volume minimization.

A comparison of the models trained on the training datasets generated by FEA showed that SVR was the most accurate for predicting $P_A$ and $L_d$, while XGBoost was the most accurate for $I_q$. When the speed-torque characteristics were predicted with these methods, the prediction errors for torques were reduced for phase current limits of 134 A and 30 A,
respectively, compared with the direct prediction method of the torque and speed.

Then, we used the surrogate models and the RCQA to minimize the PM volume. The results showed that all the error rates of torque prediction results of the best individuals were less than 5%, revealing that the proposed method could minimize the PM volume while satisfying the required driving points. Moreover, the total computation time of the proposed method using machine learning was reduced to 2.9–6.9% of the computation time for FEA only, even including the analysis time for the training dataset.

In the future, we will develop surrogate models that can also predict the demagnetization characteristics and the iron loss characteristics and will use these developed surrogate models to optimize the shape. Moreover, we will apply the proposed method to IPMSMs with other rotor topologies and motors with other structures, such as surface permanent magnet synchronous motors and synchronous reluctance motors, to show the universality of the proposed method.

Acknowledgment
This work was supported by JST, ACT-X Grant Number JPMJAX20AE, Japan.

References

(1) X. Liu, H. Chen, J. Zhao, and A. Belahcen: “Research on the Performances and Parameters of Interior PMSM Used for Electric Vehicles”, IEEE Trans. Ind. Electron., Vol.63, No.6, pp.3533–3545 (2016)
(2) J. Nerg, M. Rilla, V. Ruuskanen, J. Pyrhönen, and S. Ruotsalainen: “Direct-Driven Interior Magnet Permanent-Magnet Synchronous Motors for a Full Electric Sports Car”, IEEE Trans. Ind. Electron., Vol.61, No.8, pp.4286–4294 (2014)
(3) V. Ghorbanian, A. Salmi, and D.A. Lowther: “A Computer-Aided Design Process for Optimizing the Size of Inverter-Fed Permanent Magnet Motors”, IEEE Trans. Ind. Electron., Vol.65, No.4, pp.1819–1827 (2018)
(4) X. Xu, Z. Sun, B. Du, and L. Ai: “Pole Optimization and Thrust Ripple Suppression of New Halbach Consequent-Pole PMSLM for Ropeless Elevator Propulsion”, IEEE Access, Vol.8, pp.62042–62052 (2020)
(5) W. Zhao, A. Ma, J. Ji, X. Chen, and T. Yao: “Multiobjective Optimization of a Double-Side Linear Vernier PM Motor Using Response Surface Method and Differential Evolution”, IEEE Trans. Ind. Electron., Vol.67, No.1, pp.80–90 (2020)
(6) J. Hao, S. Sun, Y. Yang, W. Wang, and X. Chen: “Optimization of Torque Ripples in an Interior Permanent Magnet Synchronous Motor Based on the Orthogonal Experimental Method and MIGA and RBF Neural Networks”, IEEE Access, Vol.8, pp.27202–27209 (2020)
(7) S. Barnada, N. Fontana, L. Suni, D. Thomopoulos, and M. Tucci: “Deep Learning and Reduced Models for Fast Optimization in Electromagnetics”, IEEE Trans. Magn., Vol.56, No.3, pp.1–4 (2020)
(8) H. Sasaki and H. Igarashi: “Topology Optimization Accelerated by Deep Learning”, IEEE Trans. Magn., Vol.55, No.6, pp.1–5 (2019)
(9) J. Asanuma, S. Doi, and H. Igarashi: “Transfer Learning Through Deep Learning: Application to Topology Optimization of Electric Motor”, IEEE Trans. Magn., Vol.56, No.3, pp.1–4 (2020)
(10) Y. Shimizu, S. Morimoto, M. Sanada, and Y. Inoue: “Efficiency improvement and downsizing of double-layered IPMSMs containing a strong magnet for automotive applications”, 2017 19th European Conference on Power Electronics and Applications (EPE’17 ECCE Europe), Warsaw, Poland (2017)
(11) A.E. Hoerl and R.W. Kennard: “Ridge Regression: Biased Estimation for Nonorthogonal Problems”, Technometrics, Vol.12, No.1, pp.55–67 (1970)
(12) A.J. Smola and B. Schölkopf: “A tutorial on support vector regression”, Stat. Comput., Vol.14, No.3, pp.199–222 (2004)
(13) T. Chen and C. Guersten: “XGBoost: A Scalable Tree Boosting System”, Proc. 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp.785–794, San Francisco, California, USA (2016)
(14) Y. Shimizu, S. Morimoto, M. Sanada, and Y. Inoue: “Reducing Design Time of Permanent Magnet Volume Minimization for IPMSM for Automotive Applications Using Machine Learning”, the IEEE International Workshop on Sensing, Actuation, Motion Control, and Optimization, Online (2021)
(15) K. Harada, J. Sakuma, I. Ono, and S. Kobayashi: “Constraint-Handling Method for Multi-objective Function Optimization: Pareto Descent Repair Operator”, Evolutionary Multi-Criterion Optimization, pp.156–170 (2007)
(16) I.I. Baskin, N. Kireeva, and A. Varnek: “The One-Class Classification Approach to Data Description and to Models Applicability Domain”, Mol. Inform., Vol.29, No.8–9, pp.581–587 (2010)
(17) B. Schölkopf, J.C. Platt, J. Shawe-Taylor, A.J. Smola, and R.C. Williamson: “Estimating the support of a high-dimensional distribution”, Neural Comput., Vol.13, No.7, pp.1443–1471 (2001)
(18) H. Kita, I. Ono, and S. Kobayashi: “Multi-parental extension of the unimodal alternation model for real-coded GA using multi-parent: Proposal and evaluation of just generation gap (JGG)”, Proc. the 19th SICE Symposium on Decentralized Autonomous Systems, Vol.19, pp.341–346 (2007)
Machine Learning to Reduce Design Time for IPMSMs (Yuki Shimizu et al.)

Shigeo Morimoto (Senior Member) received the B.E., M.E., and Ph.D. degrees from Osaka Prefecture University, Sakai, Japan, in 1982, 1984, and 1990, respectively. In 1984, he joined Mitsubishi Electric Corporation, Tokyo, Japan. Since 1988, he has been with the Graduate School of Engineering, Osaka Prefecture University, where he is currently a Professor. His main areas of research interest are permanent magnet synchronous machines, reluctance machines and their control systems. Dr. Morimoto is a member of the IEEE, the Society of Instrument and Control Engineers of Japan, the Institute of Systems, Control and Information Engineers, the Japan Institute of Power Electronics, and Society of Automotive Engineers of Japan.

Masayuki Sanada (Senior Member) received the B.E., M.E., and Ph.D. degrees from Osaka Prefecture University, Sakai, Japan, in 1989, 1991, and 1994, respectively. Since 1994, he has been with the Graduate School of Engineering, Osaka Prefecture University, where he is currently an Associate Professor. His main areas of research interest are permanent-magnet motors for direct-drive applications, their control systems, and magnetic field analysis. Dr. Sanada is a member of the IEEE, the Japan Institute of Power Electronics, and the Japan Society of Applied Electromagnetics and Mechanics.

Yukinori Inoue (Member) received the B.E., M.E., and Ph.D. degrees from Osaka Prefecture University, Sakai, Japan, in 2005, 2007, and 2010, respectively. Since 2010, he has been with the Graduate School of Engineering, Osaka Prefecture University, where he is currently an Associate Professor. His research interests include control of electrical drives, in particular, the direct torque control of permanent magnet synchronous motors and position sensorless control of these motors. Dr. Inoue is a member of the IEEE and the Japan Institute of Power Electronics.