IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received April 20, 2021, accepted April 29, 2021, date of publication May 10, 2021, date of current version May 19, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3078668

A Data-Driven Automatic Design Method for Electric Machines Based on Reinforcement Learning and Evolutionary Optimization

TAKAHIRO SATO AND MASAFUMI FUJITA, (Member, IEEE)
Toshiba Energy Systems & Solutions Corporation, Yokohama 230-0045, Japan

Corresponding author: Takahiro Sato (takahiro26.sato@toshiba.co.jp)

ABSTRACT The design problems of electric machines are actually treated as a kind of mixed-integer problem, because the machine shapes are defined by some integer variables, such as number of slots, and the other variables, such as the tooth width, which are here called the fundamental and shape variables, respectively. To automatically solve these design problems, this article presents an automatic design method by combining the reinforcement learning and evolutionary optimization. In the proposed method, the decision process is modeled as a tree structure to seek for the fundamental variables, which are determined as a result of the tree search depending on the value functions of the nodes. Then, the shape variables are estimated from the function of the fundamental variables. These functions are constructed based on the design data, to generate which the reinforcement learning and evolutionary optimization are employed. As a result, the proposed method can automatically be adapted to unexperienced design problems through the data generation and function learning. The proposed method is applied to a design problem of a linear induction motor. It is shown that the machine designs with the prescribed performance for given specifications are automatically obtained. Moreover, it is also shown that the acceptable candidate designs can immediately be generated when the given specification is similar to the previously-solved problems by utilizing the design data generated by the past explorations.

INDEX TERMS CMA-ES, design optimization, electric machine, evolutionary algorithms, reinforcement learning, tree search.

I. INTRODUCTION

There is a major trend for electrification of various mobilities and infrastructure systems, such as automobiles [1], [2], public transports [3], [4], cooling, heating [5], [6], and so on. In addition, renewable energy resources are playing an important role in electric power systems [7]–[9]. For those systems, electric machines and electromagnetic devices are one of the key components. For example, permanent magnet motors are actively studied as drive systems of electric vehicles [10]–[12]. Moreover, doubly-fed induction generators [13], [14] and permanent magnet synchronous generators [15], [16] are utilized as generators in wind power systems. The electric machines should, therefore, carefully be designed to realize the high-performance systems in reasonable cost according to various design requirements. However, to obtain acceptable machine designs, designers have to take much effort.

Numerical design optimizations are one of the effective and helpful techniques to obtain acceptable designs with a relatively little effort [17], [18]. Thus, various optimization approaches for electric machines have been being studied such as parameter optimizations [19]–[22] and topology optimizations [23]–[27]. In these optimizations, design problems are generally expressed as the following optimization problem:

$$\begin{aligned} &\text{Minimize } F(\boldsymbol{x}), \\ &\text{subject to } \bar{G}_i(\boldsymbol{x}) = 0 \quad i = 1, \ldots, N_1, \\ &\qquad\qquad \tilde{G}_i(\boldsymbol{x}) < 0 \quad i = 1, \ldots, N_2, \end{aligned} \tag{1}$$

where $\boldsymbol{x}$ is a vector of design variables, $F(\boldsymbol{x})$ is an objective function. In addition, $\bar{G}_i(\boldsymbol{x})$ and $\tilde{G}_i(\boldsymbol{x})$ denote equality and inequality constraints, respectively. In the real-world

The associate editor coordinating the review of this manuscript and approving it for publication was Emre Koyuncu.

applications, the problem of (1) is generally black-box optimization, that is, the value of $F(\boldsymbol{x})$ itself can be evaluated by some numerical approaches when $\boldsymbol{x}$ is given, but function form of $F(\boldsymbol{x})$ is generally unknown. Therefore, the problem of (1) is sometimes solved by some heuristic optimization algorithms, like the genetic algorithms, particle swarm optimizations, and so on. For example, some evolutionary algorithms have successfully been applied to the optimization of the electric machines [21], [25], [26].

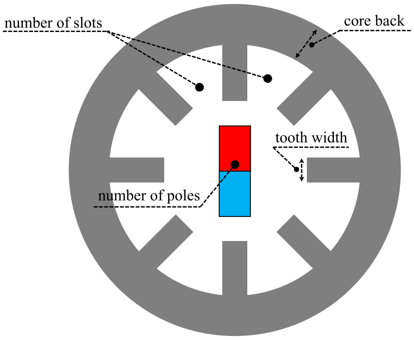


FIGURE 1. Example of design variables in electric machine, where fundamental variables are number of poles and number of slots, and shape variables are tooth width and core back.

Although the design optimizations are helpful for the designers, there remain some problems. First, in case of the electric machines, the optimization problem of (1) is actually considered as a kind of the mixed-integer problem, because there exist some fundamental integer variables which determine basic machine shapes, such as number of poles, number of slots, as shown in Fig. 1. Namely, the design variables, $\boldsymbol{x}$, can be written as follows:

$$\boldsymbol{x} = \left[\boldsymbol{x}_{\mathrm{fnd}}^{\mathrm{T}}, \boldsymbol{x}_{\mathrm{var}}^{\mathrm{T}}\right]^{\mathrm{T}}, \tag{2}$$

where $\boldsymbol{x}_{\mathrm{fnd}}$ denotes the fundamental variables, and $\boldsymbol{x}_{\mathrm{var}}$ is the other variables like the tooth width shown in Fig. 1, which are here called the shape variables. From this expression, it makes clear that the appropriate values of $\boldsymbol{x}_{\mathrm{var}}$ strongly depend on $\boldsymbol{x}_{\mathrm{fnd}}$, e.g., feasible interval of the tooth width is determined by the number of slots. For this reason, the mixed-integer optimization problems of electric machines are difficult to solve. Thus, in general cases, $\boldsymbol{x}_{\mathrm{fnd}}$ is determined by the designers in advance, and only $\boldsymbol{x}_{\mathrm{var}}$ is optimized under the fixed $\boldsymbol{x}_{\mathrm{fnd}}$. Next, when a new design specification is given, we have to solve the optimization problem of (1) in each case, even if the given specification is similar to previously-solved problems. This computational time is sometimes time-consuming when the designers have to soon prepare various candidate designs. Hence, for the further advancement of the optimization of electric machines, we should overcome these issues.

In this paper, to overcome the above-mentioned problems in the current optimizations, we present an automatic design method by combining the reinforcement learning and evolutionary optimization. In this method, to effectively treat the mixed-integer problems, the decision process of $\boldsymbol{x}_{\mathrm{fnd}}$ is modeled as a tree structure. The values of $\boldsymbol{x}_{\mathrm{fnd}}$ are determined by visiting the nodes depending on the value function of the nodes. Then, appropriate $\boldsymbol{x}_{\mathrm{var}}$ for the determined $\boldsymbol{x}_{\mathrm{fnd}}$ is determined using an estimator which is the function of $\boldsymbol{x}_{\mathrm{fnd}}$ and specifications. The value functions and the estimator are constructed from the design data, which are generated by the explorations based on the reinforcement learning and evolutionary optimizations. The designs produced by the expert designers and those generated in the past explorations are also utilized to construct the value functions and the estimator. Thanks to the utilization of the stored design data, we can immediately obtain appropriate candidate designs as a result of the greedy tree search when the given specification is similar to the past experience.

The proposed method is applied to a design problem of a linear induction motor. It is shown that the machine designs with prescribed performance can be obtained corresponding to given specifications. Then, the design data generated in these automatic designs are stored and utilized in the next design problems. It is shown that the acceptable candidate designs can immediately be obtained by the proposed method when the given specifications are similar to the previously-solved problems. Moreover, even if clearly-unexperienced specifications are given, it is also shown that the proposed method can be adapted to new specifications by performing the additional data generations.

This paper is organized as follows. Section II introduces the new automatic design method for electric machines, and section III presents detailed implementation of the proposed method. Section IV presents the automatic design results, and the last section summarizes the results.

II. AUTOMATIC DESIGN METHOD FOR ELECTRIC MACHINES

A. BASIC CONCEPT OF AUTOMATIC DESIGN

The purpose of this paper is to present the automatic design method of electric machines for determining both $\boldsymbol{x}_{\mathrm{fnd}}$ and $\boldsymbol{x}_{\mathrm{var}}$ corresponding to any specifications. This ideally means that we introduce the following inverse function:

$$\boldsymbol{x}_{\mathrm{fnd}}, \boldsymbol{x}_{\mathrm{var}} = F^{-1}\left(\boldsymbol{x}_{\mathrm{spec}}\right), \tag{3}$$

where $\boldsymbol{x}_{\mathrm{spec}} \in \mathrm{R}^{N_0}$ denotes the specification. Obviously, it is quite difficult to prepare such inverse function.

Instead of constructing the inverse function, let us consider to treat a design flow as a kind of stochastic process. We assume that m-th component of $\boldsymbol{x}_{\mathrm{fnd}}$ depends on $\boldsymbol{x}_{\mathrm{spec}}$ and the other components before m-th as follows:

$$x_{\mathrm{fnd},m} \sim p\left(x_{\mathrm{fnd},m} \middle| \boldsymbol{x}_{\mathrm{spec}}, x_{\mathrm{fnd},1}, \ldots, x_{\mathrm{fnd},m-1}\right), \tag{4}$$

where $x_{\mathrm{fnd},m}$ is m-th component of $\boldsymbol{x}_{\mathrm{fnd}}$, and $p(x)$ denotes probability function. Moreover, it is also assumed that $\boldsymbol{x}_{\mathrm{var}}$ stochastically depends on $\boldsymbol{x}_{\mathrm{fnd}}$ and $\boldsymbol{x}_{\mathrm{spec}}$ as follows:

$$\boldsymbol{x}_{\mathrm{var}} \sim p\left(\boldsymbol{x}_{\mathrm{var}} \middle| \boldsymbol{x}_{\mathrm{fnd}}, \boldsymbol{x}_{\mathrm{spec}}\right). \tag{5}$$

These stochastic modeling allow us to search prescribed designs stochastically depending on the current probability

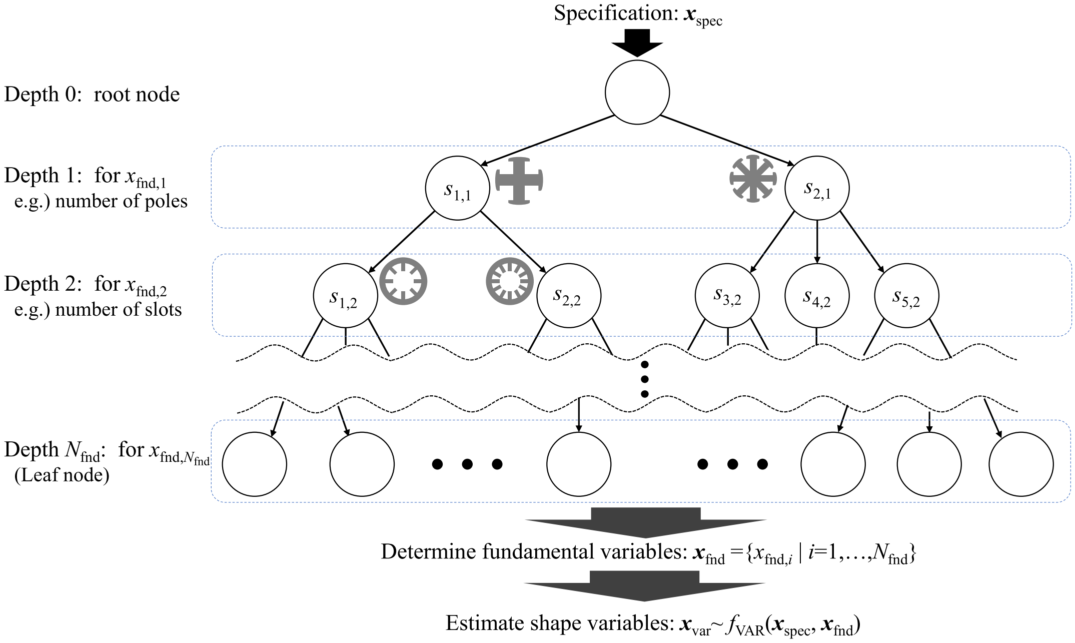

Specification: $\boldsymbol{x}_{\text{spec}}$
Depth 0: root node
Depth 1: for $x_{\text{fnd},1}$
e.g.) number of poles
Depth 2: for $x_{\text{fnd},2}$
e.g.) number of slots
Depth N_{fnd}: for $x_{\text{fnd},N_{\text{fnd}}}$
(Leaf node)
$s_{1,1}$
$s_{2,1}$
$s_{1,2}$
$s_{2,2}$
$s_{3,2}$
$s_{4,2}$
$s_{5,2}$
Determine fundamental variables: $\boldsymbol{x}_{\text{fnd}} = \{x_{\text{fnd},i} \mid i=1,\ldots,N_{\text{fnd}}\}$
Estimate shape variables: $\boldsymbol{x}_{\text{var}} \sim f_{\text{VAR}}(\boldsymbol{x}_{\text{spec}}, \boldsymbol{x}_{\text{fnd}})$

FIGURE 2. **Design search tree, whose nodes are enumerated based on feasible combinations of fundamental variables. By visiting a node in *m*-depth, *m*-th component of $\boldsymbol{x}_{\text{fnd}}$ is determined.**

functions, which can be updated using design data obtained by the stochastic searches. Moreover, the designs produced by the expert designers are also useful to update those probability functions. Namely, the stochastic modeling makes it possible to utilize past experience as well as the adaptation to unexperienced design problems through stochastic explorations.

The above-mentioned search and learning approach are almost similar to the reinforcement learning [28]. Therefore, we here propose a design method of the electric machines by introducing reinforcement learning in addition to the evolutionary optimizations. The reinforcement learning has attracted recent attentions by collaborating the deep learning [29]. Especially, the computer player for games of perfect information has achieved super-human performance starting from random play based on the reinforcement learning approach [30], [31], in which a general-purpose Monte Carlo tree search algorithm is effectively employed. In the proposed method, thus, we refer to such tree-search-based learning approach to model the decision processes of (4) and (5).

B. DESIGN SEARCH TREE

To stochastically generate the designs, the decision processes of (4) and (5) are approximately modeled based on a tree structure. In this work, we propose the tree structure shown in Fig. 2, which is here called the design search tree. The depth of the design search tree becomes $N_{\text{fnd}} + 1$, where N_{fnd} denotes the length of $\boldsymbol{x}_{\text{fnd}}$. The first layer receives the specification, $\boldsymbol{x}_{\text{spec}}$, and m-th layer corresponds to the determination of $x_{\text{fnd},m}$. In each layer, feasible numbers of the nodes are enumerated depending on the previous node. This means that the designers offer the feasible list of $x_{\text{fnd},m}$ to the computer through the tree structure.

To explore on the design search tree, the value function is defined for each node. In the tree search, the next node is stochastically selected based on the transient probability determined from the value functions. After reaching a leaf node, $\boldsymbol{x}_{\text{var}}$ are estimated as the output of an estimator f_{VAR} that is a function of $\boldsymbol{x}_{\text{fnd}}$ and $\boldsymbol{x}_{\text{spec}}$. As a result, one candidate design, i.e., the pair of ($\boldsymbol{x}_{\text{fnd}}$, $\boldsymbol{x}_{\text{var}}$), is generated. Although it is possible to adopt this design data as it is, evolutionary optimizations are employed to refine this candidate design.

C. AUTOMATIC DESIGN METHOD USING DESIGN SEARCH TREE

As a proposed automatic design method, the reinforcement learning approach is applied to the adaptation of the design search tree. Namely, the value functions and the estimator are adapted using various design data, which are generated based on the tree search and optimizations. The proposed automatic design flow is shown in Fig. 3. Here after, the proposed method is called the DEsign SEARch Tree-based Automatic design Method: DeSearTAM. The data generation using the design search tree is performed based on the value functions, the estimator f_{VAR}, and the optimizations. The data generation flow is shown in Fig. 4.

Let us represent k-th node of m-depth with $s_{k,m}$. The value function of a node $s_{k,m}$ is defined as follows:

$$V\left(s_{k,m}\right) = f_V\left(\boldsymbol{x}_{\text{spec}}, \bar{\boldsymbol{x}}_{k,m}\right), \tag{6}$$

where f_V denotes a function approximator, $\bar{\boldsymbol{x}}_{k,m}$ is a vector of the determined values of $\boldsymbol{x}_{\text{fnd}}$ after visiting $s_{k,m}$. For example, if the transient path is ($s_{0,0}$, $s_{0,1}$, $s_{3,2}$), $\bar{\boldsymbol{x}}_{k,m} = [x_1(s_{0,1}), x_2(s_{3,2})]^{\text{T}}$, where $x_m(s_{k,m})$ denotes the determined $x_{\text{fnd},m}$ by visiting $s_{k,m}$. The transient probability, $\pi_m(k', k)$, from $\text{s}_{k,m}$

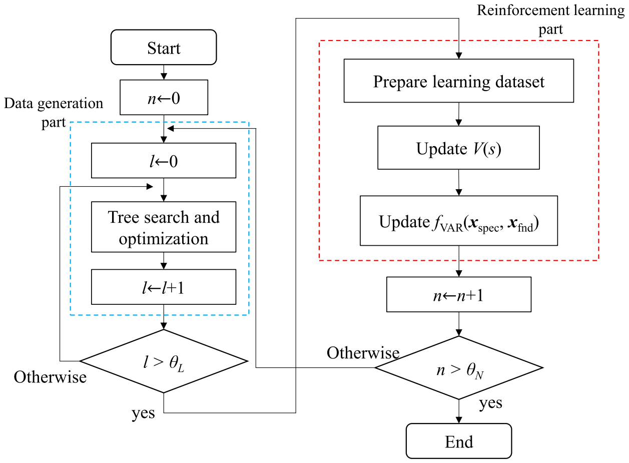


FIGURE 3. Flow of automatic design.

to a next node $s_{k',m+1}$ is obtained based on the value functions as follows:

$$\pi_m\left(k', k\right) = \frac{\frac{V\left(s_{k',m+1}\right)}{T}}{\sum\limits_{j\in S_m(k)} \frac{V\left(s_{j,m+1}\right)}{T}}, \tag{7}$$

where $S_m(k)$ is a set of the nodes reachable from the current node $s_{k,m}$, and T is a temperature parameter to control randomness.

After reaching a leaf node, the estimator generates $\boldsymbol{x}_{\mathrm{var}}$ using $\boldsymbol{x}_{\mathrm{fnd}}$ and $\boldsymbol{x}_{\mathrm{spec}}$ as follows:

$$\boldsymbol{x}_{\mathrm{var}} \sim f_{\mathrm{VAR}}\left(\boldsymbol{x}_{\mathrm{spec}}, \boldsymbol{x}_{\mathrm{fnd}}\right), \tag{8}$$

where f_{VAR}: $\mathrm{R}^{N_0+N_{\mathrm{fnd}}} \rightarrow \mathrm{R}^{N_{\mathrm{var}}}$. It is possible to stochastically generate $\boldsymbol{x}_{\mathrm{var}}$ by employing some probability density functions as f_{VAR} for the stochastic search of $\boldsymbol{x}_{\mathrm{var}}$. However, for simplicity in this work, we treat f_{VAR} as a deterministic function, and the parameter optimization of $\boldsymbol{x}_{\mathrm{var}}$ is performed as an effective search of $\boldsymbol{x}_{\mathrm{var}}$.

It is here noted that we can take arbitrary implementations of f_V, f_{VAR}, and the optimization of $\boldsymbol{x}_{\mathrm{var}}$. In the next section, thus, our implementations are provided.

III. DETAILED IMPLEMENTATIONS

A. VALUE FUNCTION AND TRANSIENT PROBABILITY

The value function of the node is expressed by a function approximator as described in (6). Although we can use any approximators, it is a natural approach to prepare N_{fnd} different approximators because the input dimension of the approximator is increased with the depth of the nodes. In this work, we employ the Gaussian process regression (GPR) [32], because it can be relatively easily constructed without the heavy learning process if the learning dataset is not so huge. Using GPR, the value function of the m-depth nodes for the current dataset is expressed as follows:

$$V\left(s_{k,m}\right) = \boldsymbol{\varphi}_m\left(D_m, s_{k,m}\right)^{\mathrm{T}} \Phi_m^{-1}\left(D_m\right) \boldsymbol{y}\left(D_m\right), \tag{9}$$

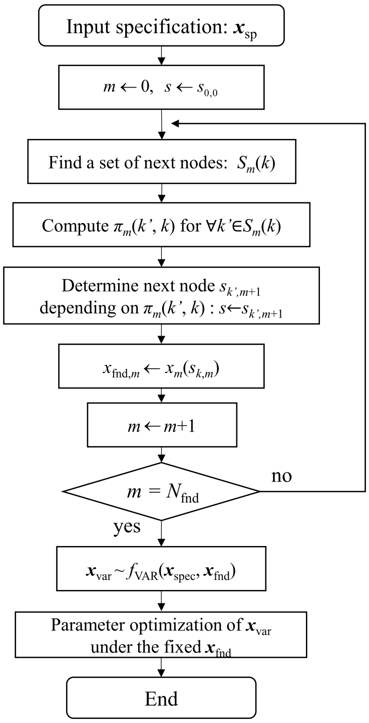

Input specification: $\boldsymbol{x}_{\mathrm{sp}}$
$m \leftarrow 0,\ s \leftarrow s_{0,0}$
Find a set of next nodes: $S_m(k)$
Compute $\pi_m(k', k)$ for $\forall k' \in S_m(k)$
Determine next node $s_{k',m+1}$ depending on $\pi_m(k', k)$: $s \leftarrow s_{k',m+1}$
$x_{\mathrm{fnd},m} \leftarrow x_m(s_{k,m})$
$m \leftarrow m$+1
$m = N_{\mathrm{fnd}}$
no
yes
$\boldsymbol{x}_{\mathrm{var}} \sim f_{\mathrm{VAR}}(\boldsymbol{x}_{\mathrm{spec}}, \boldsymbol{x}_{\mathrm{fnd}})$
Parameter optimization of $\boldsymbol{x}_{\mathrm{var}}$ under the fixed $\boldsymbol{x}_{\mathrm{fnd}}$
End

FIGURE 4. Flow of data generation using design search tree.

where D_m is the learning dataset for m-depth. Furthermore,

$$\boldsymbol{\varphi}_m\left(D_m, s_{k,m}\right) = \left[\phi\left(\tilde{\boldsymbol{x}}_m^1, \tilde{\boldsymbol{x}}_{k,m}\right), \cdots, \phi\left(\tilde{\boldsymbol{x}}_m^K, \tilde{\boldsymbol{x}}_{k,m}\right)\right]^{\mathrm{T}}, \tag{10}$$

$$\Phi_m\left(D_m\right) = \begin{bmatrix} \phi\left(\tilde{\boldsymbol{x}}_m^1, \tilde{\boldsymbol{x}}_m^1\right) & \cdots & \phi\left(\tilde{\boldsymbol{x}}_m^1, \tilde{\boldsymbol{x}}_m^K\right) \\ \vdots & \ddots & \vdots \\ \phi\left(\tilde{\boldsymbol{x}}_m^K, \tilde{\boldsymbol{x}}_m^1\right) & \cdots & \phi\left(\tilde{\boldsymbol{x}}_m^K, \tilde{\boldsymbol{x}}_m^K\right) \end{bmatrix}, \tag{11}$$

$$\boldsymbol{y}\left(D_m\right) = \left[y_1 \cdots y_K\right]^{\mathrm{T}}, \tag{12}$$

where $\tilde{\boldsymbol{x}}_m^i$ and $\tilde{\boldsymbol{x}}_{k,m}$ are the vector of $[\boldsymbol{x}_{\mathrm{spec}}^{\mathrm{T}}, x_{\mathrm{fnd},1}, \ldots, x_{\mathrm{fnd},m}]^{\mathrm{T}}$ corresponding to the i-th learning data in D_m and the node $s_{k,m}$, respectively. Moreover, y_i is the fitness of i-th learning data in D_m for a given design problem. In addition, ϕ is a kernel function, as which the radial basis function is adopted.

The use of GPR allows us to evaluate the uncertainness of (9) as follows:

$$\sigma\left(s_{k,m}\right) = \phi\left(\tilde{\boldsymbol{x}}_{k,m}, \tilde{\boldsymbol{x}}_{k,m}\right) \\ -\boldsymbol{\varphi}_m\left(D_m, s_{k,m}\right)^{\mathrm{T}} \Phi_m^{-1}\left(D_m\right) \boldsymbol{\varphi}_m\left(D_m, s_{k,m}\right). \tag{13}$$

This information is effective to control the trade-off between exploitation and exploration. For this reason, the transient probability of (7) is modified as follows:

$$\pi_m\left(k', k\right) = \frac{\frac{U\left(s_{k',m+1}\right)}{T}}{\sum\limits_{j\in S_m(k)} \frac{U\left(s_{j,m+1}\right)}{T}}, \tag{14}$$

where $U(s_{k,m})$ denotes the UCB acquisition function [34] in the Bayesian optimization as follows:

$$U\left(s_{k,m}\right) = V\left(s_{k,m}\right) + \beta\sigma\left(s_{k,m}\right), \tag{15}$$

where β is a user-defined hyper-parameter. When $\beta > 0$, un-visited nodes are actively visited, and in case of $\beta < 0$, the probability to visit them becomes small. By changing β and T during the automatic design, the trade-off between exploitation and exploration is controlled.

B. ESTIMATOR OF x_{var}

The GPR is also employed as the estimator, f_{VAR}, described in (8). Since all the components of $\boldsymbol{x}_{\mathrm{var}}$ must be estimated, we here simply apply GPR to the estimation of each component. When the reached leaf node is $s_{k,N\mathrm{fnd}}$, j-th component of $\boldsymbol{x}_{\mathrm{var}}$, which is expressed as $x_{\mathrm{var},j}$, is estimated as follows:

$$x_{\mathrm{var},j} = \boldsymbol{\varphi}_{N_{\mathrm{fnd}}}\left(D_{N_{\mathrm{fnd}}}, \tilde{\boldsymbol{x}}_{k,N_{\mathrm{fnd}}}\right)^{\mathrm{T}} \Phi^{-1}_{N_{\mathrm{fnd}}}\left(D_{N_{\mathrm{fnd}}}\right) \boldsymbol{y}_{\mathrm{var},j}\left(D_{N_{\mathrm{fnd}}}\right), \tag{16}$$

where $\boldsymbol{y}_{\mathrm{var},j} = [x^0_{\mathrm{var},j}, \ldots, x^K_{\mathrm{var},j}]^{\mathrm{T}}$, and $x^k_{\mathrm{var},j}$ denotes $x_{\mathrm{var},j}$ corresponding to the k-th design data in $D_{N\mathrm{fnd}}$.

C. PREPARATION OF LEARNING DATASET

The learning dataset, D_m, is prepared from the stored design data. The design search tree is expanded corresponding to the stored data. In each leaf node, the assigned data with best fitness is adopted as a learning data. In one upper depth, the child node with the best fitness is referred. This process is repeated until the root node. This flow is summarized as the following steps:

1. For each data, d, in the stored design dataset,
   A. Expand the design search tree to find the corresponding leaf node.
   B. If the corresponding leaf node is not expanded, the data d is assigned to the leaf node. Otherwise, if the fitness of d is better than that of the current assigned data, d is newly assigned to the leaf node.
2. The learning dataset for leaf nodes, $D_{N\mathrm{fnd}}$, is composed of all the data assigned to the leaf nodes.
3. $m \leftarrow N_{\mathrm{fnd}} - 1$
   (Repeat the following steps until $m = 1$)
4. For each node $s_{k,m}$ in m-depth, find its child node with the best fitness value.
5. The fitness of $s_{k,m}$ is set to that of the best child.
6. The learning dataset for m-depth, D_m, is composed of all the data for the nodes in m-depth.
7. $m \leftarrow m$-1, and go back to the step 4.

D. PARAMETER OPTIMIZATION METHOD

As a parameter optimization method for $\boldsymbol{x}_{\mathrm{var}}$, we employ the covariance matrix adaptation evolution strategy (CMA-ES) [34]–[36] with the restart [37]. The CMA-ES uses the multivariate normal distribution as a sampling distribution of candidate solutions, and the distribution parameters of the normal distribution are updated based on the objective ranking of the candidate solutions.

In DeSearTAM, the initial mean vector of the normal distribution, $\boldsymbol{m}_0$, is set to $\boldsymbol{x}_{\mathrm{var}}$ estimated as (16). Then, at the algorithmic iteration t, the candidate solutions, $\boldsymbol{X}_i \in \mathrm{R}^{N_{var}}$ $(i = 1, \ldots, M)$, are generated as follows:

$$\boldsymbol{X}_i = \boldsymbol{m}_t + \sigma_t\sqrt{\mathrm{C}_t}z_i, \tag{17}$$

where σ_t and C_t are the other distribution parameters. Moreover,

$$z_i \sim N(\boldsymbol{0}, \mathrm{I}), \tag{18}$$

and I is identity matrix with the size of $N_{\mathrm{var}} \times N_{\mathrm{var}}$. Then, the candidate solutions are evaluated. In the evaluations, $\boldsymbol{x}_{\mathrm{fnd}}$ determined by the tree search is employed, and the box-constraint condition is considered:

$$\max[(1-\delta)\boldsymbol{x}_{\mathrm{var}}, \boldsymbol{\varepsilon}] \leq \boldsymbol{X}_i \leq (1+\delta)\boldsymbol{x}_{\mathrm{var}}, \tag{19}$$

where ε is a vector for the minimum values of $\boldsymbol{x}_{\mathrm{var}}$, and δ is a hyper-parameter of DeSearTAM to control search interval. After the evaluations of $\boldsymbol{X}_i$, the ranking of the candidate solutions is computed, where MCR-mod [38] is employed to handle (19) and the constraints of the given design problem. Based on the rankings, the distribution parameters are updated. Please see the references [34]–[36] for the detailed update process. As a result of CMA-ES-based parameter optimization, the best candidate solution is utilized as the refined $\boldsymbol{x}_{\mathrm{var}}$.

IV. AUTOMATIC DESIGN OF LINER INDUCTION MOTOR

A. PROBLEM DEFINITION

In this work, we here define a design problem of a linear induction motor based on the references [39]–[41]. The machine performance of the linear induction motor is simply evaluated based on the equivalent circuit whose circuit constants are derived from the geometric shape of the machine. Hence, this design problem is suitable for the test of the effectiveness of DeSearTAM. The main purpose of this design problem is to find some machine designs which realize prescribed rotor speed and thrust force with as small volume as possible considering some constraint conditions. The overview of the machine shape is shown in Fig. 5. The specification, the fundamental and shape variables are summarized in Table 1, 2, respectively. Note here that, although n_d and n_c are integer variables, these are treated as the shape variables, because the basic shape of the machine does not strongly depend on them.

The design problem of the liner induction motor is formulated as follows:

$$F\left(\boldsymbol{x}_{\mathrm{fnd}}, \boldsymbol{x}_{\mathrm{var}} | \boldsymbol{x}_{\mathrm{spec}}\right) = V_{\mathrm{motor}} \rightarrow \min., \tag{20}$$

subject to

$$\eta > \theta_\eta, \tag{21}$$

$$F_x > \theta_{F1}, \tag{22}$$

$$T_c < \theta_T, \tag{23}$$

$$B_{core} < \theta_B, \tag{24}$$

$$B_{teeth} < \theta_B, \tag{25}$$

$$B_{gap} < \theta_{B2}, \tag{26}$$

$$\frac{F_{\max}}{F_x} > \theta_{F2}, \tag{27}$$

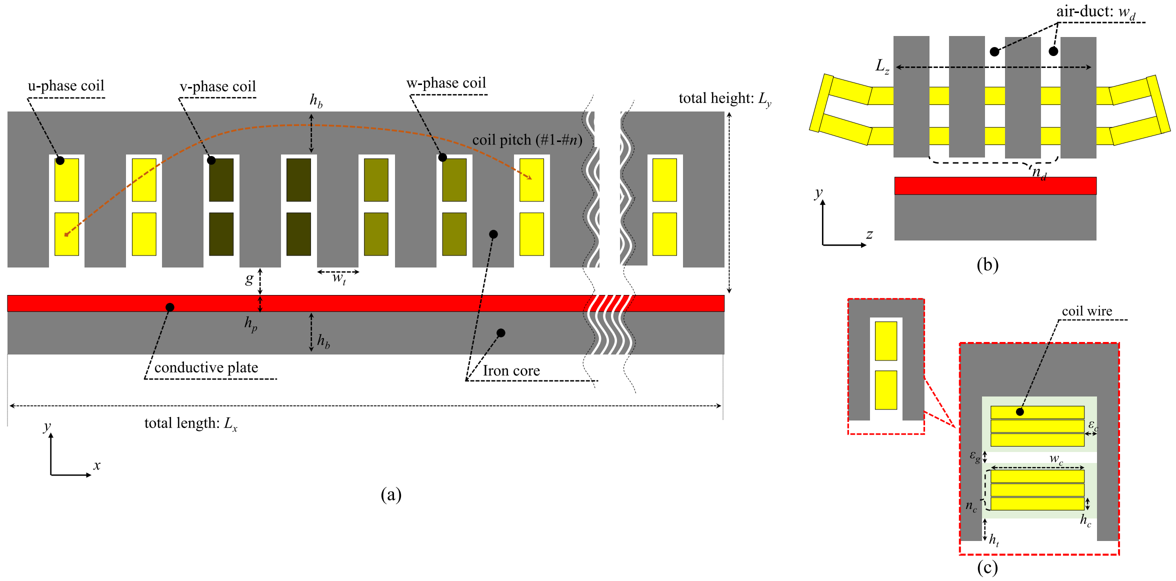


FIGURE 5. Linear induction motor. a) x-y plane, b) y-z plane, c) cross-sectional area of coil.

$$s_{\max} - s_{rated} > 0, \tag{28}$$

$$h_c < \delta_{skin}, \tag{29}$$

where η is efficiency, and V_{motor} denotes the total volume of the iron cores and coils. Furthermore, F_x and $F_{\max}$ are thrust force in rated condition and maximum condition, respectively. The temperature rise of the coils is expressed by T_c, and B_{core}, B_{teeth}, B_{gap} are flux density of the core back, tooth, gap, respectively. Moreover, s_{rated} and $s_{\max}$ denote the slip for rated and the maximum force condition, respectively. In addition, δ_{skin} is skin depth of the coil wire. The computational processes for the above machine-performance are summarized in the appendix.

The design search tree for the linear induction motor is defined here. Because the number of fundamental variables of the linear induction motor is 3, the total depth of the design search tree becomes 4. In this work, the feasible numbers of each fundamental variable are set as follows:

- Number of poles: 2, 4, 6, 8
- Number of slots per pole and phase: 1, 2, 3, 4,5, 6
- Coil pitch: integer values with interval of $\beta_c = [0.7, 1.0]$, where $\beta_c = (\#n\text{-}1)/(3N_{\text{per}})$ is percent coil pitch.

For example, the design search tree is expanded as shown in Fig. 6.

The value function $V(s)$, which is to be maximized, is simply defined based on the death-penalty approach as follows:

$$V(s) = -\alpha_1 V_{\text{motor}} - \alpha_{\text{death}} N_{\text{viol}}, \tag{30}$$

where N_{viol} denotes the number of violated constraint conditions, and $\alpha_{\text{death}} = 10^6$ is a penalty coefficient. In addition, $\alpha_1 = 10^{-9}$ is a weighting coefficient to balance V_{motor} and penalty term. On the other hand, during the CMA-ES-based parameter optimization, we employ the MCR-mod [38], instead of the penalty-based approach.

TABLE 1. Specification for linear induction motor: $\boldsymbol{x}_{\text{spec}}$.

Symbol	Name
V_{in}	Input line voltage (V)
f	Frequency (Hz)
v_r	Rated rotor speed (km/h)

TABLE 2. Design variables for liner induction motor.

	Symbol	Name
Fundamental variables: $\boldsymbol{x}_{\text{fnd}}$	p	Number of poles
	N_{per}	Number of slots per pole and phase (Number of slots, N_{sl}=3pN_{per})
	#n	Coil pitch
Shape variables: $\boldsymbol{x}_{\text{var}}$	w_t	Tooth width (mm)
	h_b	Core back width (mm)
	g	Gap length (mm)
	L_z	Core length (mm)
	n_d	Number of ducts
	w_c	Wire width (mm)
	h_c	Wire height (mm)
	n_c	Number of turns
	h_p	Plate thickness (mm)
Fixed variables	w_d=5mm	Duct width
	h_t=10mm	Teeth tip thickness
	ε_c=1mm	Main insulation
	ε_g=1mm	Interval between coils

B. AUTOMATIC DESIGNS THROUGH EXPLORATION

The design problem of the linear induction motor is solved using DeSearTAM from empty design dataset. We here set six cases as summarized in Table 3. In each case, prescribed designs are explored over 150 search iterations: $\theta_N = 150$. The hyper-parameters of DeSearTAM in each iteration are summarized in Table 4 so that various designs are widely explored in early stage and hopeful ones are intensively searched in latter stage, respectively. The other thresholds for the constraint conditions are set as summarized in Table 5.

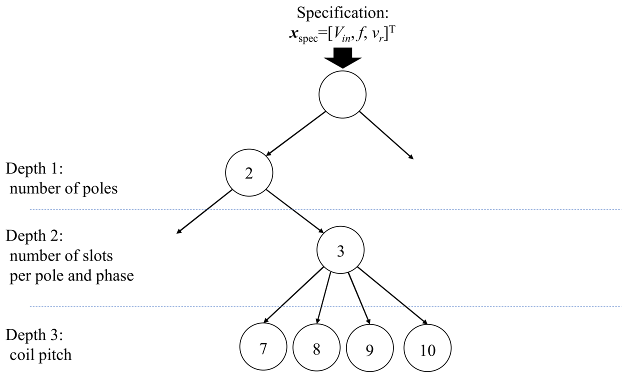


FIGURE 6. Example of expanded design search tree for linear induction motor.

TABLE 3. Design cases.

Case	specifications			thresholds
	V_{in} (V)	f (Hz)	v_r (km/h)	θ_{F1} (N)
Case-A	200	10	30	2000
Case-B	↑	↑	↑	4000
Case-C	↑	↑	↑	6000
Case-D	↑	↑	60	2000
Case-E	↑	↑	↑	4000
Case-F	↑	↑	↑	6000

TABLE 4. Hyper-parameter setting of DeSearTAM.

Iteration number, n	θ_L: number of tree searches in each iteration (see Fig. 4)	T: temperature parameter in transient probability	β: coefficient of UCB function	δ: search interval of x_{var} in CMA-ES
0-50	5	1000	1	10
51-100	↑	1	0	1
101-150	↑	0.01	-1	0.5

TABLE 5. Thresholds in design problems of linear induction motor.

θ_η (%)	θ_T (degC)	θ_B (T)	θ_{B2} (T)	θ_{F2} (-)
70	120	1.5	1.0	1.5

As for the parameter optimization, the number of algorithmic iterations and number of candidate solutions in CMA-ES are set to 300 and 40, respectively. The other hyper-parameters in CMA-ES are set to the default values [36].

The changes in the best value of (30) against the search iterations for the Cases-A, D, F, for instance, are shown in Fig. 7. We can see that the best value is gradually increased with the search iterations. From about 50 to 100 iterations, the best value is almost constant. This means that acceptable designs are obtained by 100 iterations through the tree search and optimizations. After 100 iterations, the best value is again improved. This is because the greedy searches are performed after 100 iterations, and as a result, currently-obtained designs are well refined.

The resultant machine data are summarized in Table 6 in detail, and the resultant cross-sectional machine shapes are shown in Fig. 8. It can be seen that the integer values and

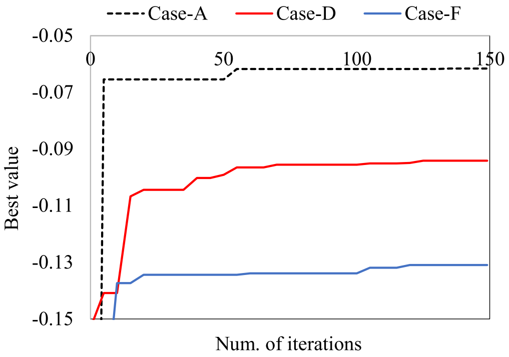


FIGURE 7. Changes in best value against search iterations.

L_z vary according to the specifications. In case of $\theta_{F1} = 2000$Nm, L_z is relatively short, while L_x becomes relatively long due to larger p and N_{per}. This is because longer L_x is effective to reduce V_{motor} for producing smaller F_x. It is also shown that p and N_{per} become larger when $v_r = 60$km/h in comparison with the cases when $v_r = 30$km/h, because the pole pitch should be set long to make the rotor speed higher. On the other hand, in case of the Case-F, p and N_{per} are small as the Cases-A, B, C, whereas w_t is clearly larger than the other cases. This is due to the fact that w_t must be set long to suppress B_{teeth} when prescribed v_r and F_x are large. From these resultant designs, it can be concluded that DeSearTAM can effectively solve the mixed-integer problems corresponding to the given specifications and constraints.

C. UTILIZATION OF STORED DESIGN DATA

Next, we try to solve other design problem by utilizing the design data obtained by the automatic designs described in the previous section. Hence, all the generated data are collected and stored.

We here modify the design problem of the linear induction motor as follows:

$$F\left(\boldsymbol{x}_{\text{fnd}}, \boldsymbol{x}_{\text{var}} | \boldsymbol{x}_{\text{spec}}\right) = \eta \rightarrow \max., \tag{31}$$

subject to

$$L_x < \theta_X, \tag{32}$$
$$L_y < \theta_Y, \tag{33}$$
$$L_z < \theta_Z, \tag{34}$$

and the constraints of (22)-(29) are also considered. In short, we next try to find high-efficiency machine designs under the limited machine volume. In general optimization methods, because the design problem is changed, we have to perform optimization computations from zero-start. The DeSearTAM, on the other hand, would find acceptable candidate designs without the optimization by utilizing the stored design data.

First, we here consider two cases as follows:

Case-A2: $v_r = 30$km/h, $\theta_{F1} = 6000$Nm

Case-B2: $v_r = 60$km/h, $\theta_{F1} = 6000$Nm

The other specifications are the same as summarized in Table 3, and $\theta_X, \theta_Y, \theta_Z$, are set to 3500mm, 50mm, 150mm,

TABLE 6. Resultant design data.

	Case-A	Case-B	Case-C	Case-D	Case-E	Case-F
p	4	2	2	2	2	2
N_{per}	2	2	2	4	4	2
$\#n$	5	5	5	9	9	5
w_t (mm)	52	50	50	45	41	103
h_b (mm)	169	100	97	136	92	200
g (mm)	5	7	6	5	5	5
L_z (mm)	72	185	228	146	235	135
n_d (mm)	5	10	16	9	14	9
w_c (mm)	2.1	2.9	3	2.4	2.8	3.2
h_c (mm)	20.1	26.8	26.8	23.3	27.7	31.9
n_c (mm)	17	20	19	10	9	15
h_p (mm)	1	1	1	1	1	1
$10^{-3}\times V_{\mathrm{motor}}$ (mm^3)	61570.59	70658.92112	77284.28	94127.45	114141.45	130963.37
η (%)	71.64	72.24	72.33	82.55	84.73	87.67
F_x (Nm)	2009.8	4484.9	6644.2	2010.1	4019.2	6075.9
$F_{\max}$ (Nm)	6697.3	7058.8	10194.5	7732.3	21177.8	20297.7
T_c (degC)	113.6	114.2	116.3	113.6	118.9	115.5
B_{core} (T)	1.5	1.5	1.48	1.49	1.49	1.5
B_{teeth} (T)	1.22	1.5	1.44	1.12	0.84	1.46
B_{gap} (T)	0.79	0.5	0.43	0.37	0.27	0.51
s_{rated} (%)	9.4	16.7	16.7	4.3	4.6	4.8
$s_{\max}$ (%)	41.4	39.1	37.6	18.2	22.6	17.6
L_x (mm)	1840.0	1000.4	1000.4	1741.8	1747.4	1750.6
L_y (mm)	27.4	25.4	24.7	21.2	17	32.8

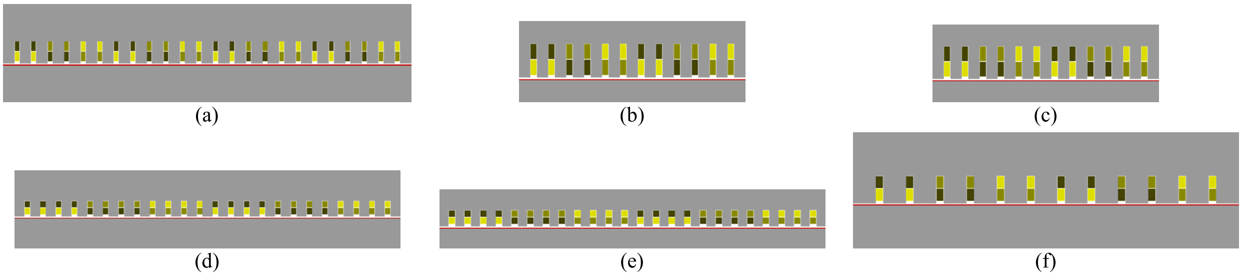

(a) (b) (c) (d) (e) (f)

FIGURE 8. Resultant cross-sectional machine shapes, a) Case-A, b) Case-B, c) Case-C, d) Case-D, e) Case-E, f) Case-F.

respectively. For these two cases, we generate design data by the greedy search without the exploration and parameter optimization; T is set to 0.01 and CMA-ES is not employed. These results are summarized in the left columns of Table 7, from which we can see that the acceptable designs with the prescribed performance are found only from the tree search. It can be concluded from this result that DeSearTAM can immediately generate the candidate designs without any optimizations by utilizing the design data. Moreover, by applying the parameter optimization, these designs can be refined. The right columns of Table 7 summarize the refined designs. From this Table, it is shown that the efficiency of the refined designs is improved. The resultant machine shapes before and after the optimization are shown in Fig. 9, which shows that the coil area is widened by the optimization to reduce the copper loss. Like these results, it can be seen that DeSearTAM can easily produce some candidate designs if we have appropriate datasets, and those are refined for more suitable ones by the parameter optimizations.

Next, we set another design case as follows:

Case-C2: $v_r = 20$km/h, $\theta_{F1} = 6000$Nm

The other specifications and settings are the same as the previous ones. Again, the greedy search is performed to produce one design data, to which parameter optimization is also applied. The resultant machine designs are summarized in the 1st and 2nd rows of Table 8. It is shown from this table that both designs have too small F_x and η. This is because we have no data for the given v_r, and thus, appropriate integer values cannot be found. For this reason, to adapt to this new specification, we generate new design data through the exploration using the design search tree and the optimization. The search settings are the same as summarized in Table 4. The resultant machine design is summarized in the 3rd row of Table 8, which makes it clear that we can obtain the machine data with the prescribed F_x and η. This machine design has different p and N_{per} before the exploration, that is, the value functions are adapted to the newly-given specification.

From these numerical results, it is shown that DeSearTAM can be adapted to various specifications by repeating the

TABLE 7. Resultant design data for Cases-A2 and -B2.

	Case-A2	Case-B2	Case-A2	Case-B2
	Before Optimization		After Optimization	
p	6	2	6	2
N_{per}	1	1	1	1
$\#n$	4	3	4	3
w_t (mm)	102	200	69	185
h_b (mm)	104	199	131	191
g (mm)	5	5	5	5
L_z (mm)	99	136	103	146
n_d (mm)	1	1	1	1
w_c (mm)	5.6	4.8	9	6.8
h_c (mm)	43.7	47.7	69.3	67.8
n_c (mm)	16	20	15	20
h_p (mm)	1	1	2	1
$10^{-3}\times V_{\mathrm{motor}}$ (mm^3)	205030.2	185685.0	360940.0	260757.0
η (%)	84.93	94.23	92.77	95.78
F_x (Nm)	7681.9	6126.8	6021.3	6044.7
F_{max} (Nm)	39915.5	105594.8	100261.5	118646
T_c (degC)	73.7	97.4	23.6	25.3
B_{core} (T)	1.44	1.5	1.17	1.45
B_{teeth} (T)	0.49	1.49	0.74	1.5
B_{gap} (T)	0.73	0.74	0.98	0.74
s_{rated} (%)	9.7	2	3.9	2.9
s_{max} (%)	51.3	19.6	46	25.8
L_x (mm)	2767.8	1700.6	2601.6	1716.2
L_y (mm)	31.6	42.7	43.3	49.9

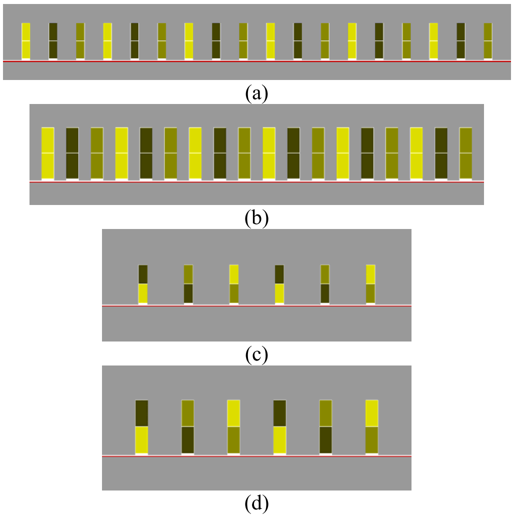


FIGURE 9. Resultant cross-sectional machine shapes for modified design problem, a) Case-A2 before optimization, b) Case-A2 after optimization, c) Case-B2 before optimization, d) Case-B2 after optimization.

exploration and learning, and it can be concluded that DeSearTAM is effective for solving the mixed-inter design problems of the electric machines.

V. CONCLUSION

For the automatic design of electric machines considering all the fundamental and shape variables, an automatic design method named DeSearTAM has been proposed. In this method, the design process for determining the fundamental variables is modeled by the tree structure. Based on the reinforcement learning approach, the value functions of the nodes and the estimator for the shape variables are constructed using the various design data generated by the explorations.

TABLE 8. Resultant design data for Case-C2.

	Before optimization	After optimization	After additional exploration
p	6	6	8
N_{per}	4	4	1
$\#n$	13	13	3
w_t (mm)	50	20	47
h_b (mm)	179	81	78
g (mm)	13	21	5
L_z (mm)	169	188	104
n_d (mm)	11	18	1
w_c (mm)	10	4.1	11.4
h_c (mm)	41.1	16.4	45.9
n_c (mm)	6	6	17
h_p (mm)	5	2	5
$10^{-3}\times V_{\mathrm{motor}}$ (mm^3)	1069548.8	127896.5	255576.0
η (%)	7.29	21.31	89.9
F_x (Nm)	19.9	2571.5	6969.5
F_{max} (Nm)	263.7	2695.5	24019.9
T_c (degC)	2.6	322.3	24.6
B_{core} (T)	0.48	1.24	1.49
B_{teeth} (T)	0.14	0.42	0.62
B_{gap} (T)	0.12	0.23	0.97
s_{rated} (%)	75.4	40.8	4.8
s_{max} (%)	3	30	25.4
L_x (mm)	6782.0	2813.6	2334.2
L_y (mm)	33.2	17.1	49.9

The DeSearTAM has been applied to the design problem of the linear induction motor. It has been shown that machine designs whose thrust force is larger than the given thresholds are automatically obtained corresponding to the given specifications including the appropriate selection of the number of poles and slots. Moreover, all the generated designs are stored and utilized to solve other design problems. It has been shown that the candidate designs whose efficiency is higher than 80% can automatically and immediately be obtained without any optimizations thanks to the utilization of the various design data. Even if the given specification is unexperienced one, it has also been shown that DeSearTAM can effectively be adapted to newly-given specifications through the tree search and optimization. From these results, it has been concluded that DeSearTAM is effective for solving the design problems of electric machines considering all the fundamental and shape variables.

For the future works, we will apply the proposed method to various design problems of electric machines. Moreover, the proposed method will be modified for the multi-objective design problems.

APPENDIX

The design problem of the linear induction motor used in this work is based on the references [39]-[41]. We here describe the evaluation formula of the machine-performance based on the equivalent circuit, provided that we do not consider the end-region effect of the iron cores and the iron loss for simplicity, because the purpose of this work is to present the automatic design method, not to develop high-performance linear motors. For the same reason, we do not evaluate the machine-performance of the resultant motors by the finite element method in detail. The difference between the finite

TABLE 9. Symbols used in evaluations of linear induction motor.

Symbol	Descriptions	Value
g_0	Magnetic air-gap (mm)	g+h_p
g_e	Effective air-gap (mm)	$k_g g_0$
H_s	Slot height (mm)	$2n_c h_c$+$4\varepsilon_c$+ε_g+h_t
I	Stator current (A)	
k_d	Distribution factor	sin(π/6) / (N_{per}×sin(π/6 N_{per}))
k_g	Cater coefficient	(W_s+w_t) / (W_s+w_t-γg_0)
k_p	Pitch factor	sin(0.5$\pi\beta_c$)
k_w	Winding factor	$k_d k_p$
l_{ce}	Coil length in end-region (mm)	2×1.5$\beta_c\tau$
L_{eff}	Effective core length (mm)	L_z-$w_d n_d$
l_{st}	Total length of stator coil per turn (mm)	2(L_z+1.5$\beta_c\tau$)
L_x	Total motor length (mm)	$3pN_{per}$(W_s+w_t)+w_t
L_y	Total motor height (mm)	g+H_s+h_b
N_1	Number of turns per phase	$n_c N_{\text{slot}}$/3
N_{slot}	Number of slots	$3pN_{per}$
s	Slip	
S_{wire}	Cross-sectional area of coil wire (mm^2)	$w_c h_c$
τ	Pole pitch (mm)	L_x/p
V_0	Input phase voltage (V)	$V_{\text{in}}/\sqrt{3}$
v_s	Synchronous velocity (m/s)	$2\tau f$
W_s	Slot width (mm)	w_c+$2\varepsilon_c$
β_c	Coil pitch (percentage)	(#n-1) / (N_{slot}/3)
κ	Thermal conductivity of coil (W/(m^{2}degC))	30.0
σ_c	Conductivity of coil wire (S/m)	4.8×10^7
σ_P	Conductivity of conductive plate (S/m)	3.5×10^7

element method and the equivalent circuit-based evaluation has been discussed in the references [39], [40].

The thrust force and efficiency are evaluated based on the equivalent circuit. The circuit constants of the equivalent circuit are derived from the geometric shape of the machine. The other features of the machine are also computed from the machine shape analytically. The symbols used in the evaluations are summarized in Table 9. The thrust force, F_x, and efficiency, η, are computed as follows:

$$F_x = \frac{3I^2R_2}{\left(\frac{1}{s^2G^2}+1\right)v_s s}, \tag{35}$$

$$\eta = \frac{F_x v_s (1-s)}{F_x v_s + 3I^2R_1}, \tag{36}$$

where

$$R_1 = \frac{N_1 l_{st}}{\sigma_c S_{\text{wire}}}, \tag{37}$$

$$R_2 = \frac{X_m}{G}, \tag{38}$$

$$X_1 = \frac{2\pi\mu_0 \Big[\Big[\Big\{\lambda_s\left(1+\frac{3}{p}\right)+\lambda_d\Big\}\frac{W_s}{N_{per}}+\lambda_e l_{ce}\Big]N_1^2}{p}, \tag{39}$$

$$X_m = \frac{24\pi\mu_0 f W_s k_w N_1^2 \tau}{\pi^2 p g_e}, \tag{40}$$

$$G = \frac{2\pi\mu_0 f \tau^2 \sigma_p h_p}{\pi g_e}. \tag{41}$$

$$I = \frac{V_0}{\left\| R_1 + jX_1 + \frac{jsR_2X_m}{sR_2+jX_m} \right\|}, \tag{42}$$

and

$$\lambda_s = \frac{H_s\left(1+3k_p\right)}{12W_s}, \tag{43}$$

$$\lambda_e = 0.3\left(3k_p - 1\right), \tag{44}$$

$$\lambda_d = \frac{5\left(\frac{g_e}{W_s}\right)}{5+4\left(\frac{g_0}{W_s}\right)}. \tag{45}$$

Moreover,

$$\begin{aligned} V_{\text{motor}} &= \{w_t H_s (N_{\text{slot}}+1) + L_x h_b\} L_{eff} \\ &\quad + L_x \left(h_p + h_b\right) L_z \\ &\quad + 2\left(L_z + 1.5\tau\beta_c\right) w_c h_c n_c N_{\text{slot}}, \end{aligned} \tag{46}$$

$$T_c = \frac{3I^2R_1}{\kappa S_{surf}}, \tag{47}$$

$$B_{teeth} = \frac{\phi}{\frac{N_{\text{slot}}}{3} w_t L_{eff}}, \tag{48}$$

$$B_{core} = \frac{\phi}{2h_b L_{eff}}, \tag{49}$$

$$B_{gap} = \frac{\phi}{\tau L_z k_g}, \tag{50}$$

where

$$\phi = \frac{V_0}{\sqrt{2}\pi f N_1 k_w}, \tag{51}$$

$$S_{surf} = (2 + n_d) L_x H_s + L_x L_z + 2H_s L_z. \tag{52}$$

When one candidate design is generated, the rated slip is calculated as follows:

$$s_{rated} = \frac{v_s - v_r}{v_s}, \tag{53}$$

and the machine-performance for the rated condition is computed. In addition, the slip for the maximum force condition, $s_{\max}$, is simply searched by increasing s in small increments from 0.0 to 1.0. Then, the constraints of (27) and (28) are evaluated.

REFERENCES

[1] B. Lequesne, "Automotive electrification: The nonhybrid story," *IEEE Trans. Transport. Electrific.*, vol. 1, no. 1, pp. 40–53, Jun. 2015.

[2] A. Adib, M. B. Shadmand, P. Shamsi, K. K. Afridi, M. Amirabadi, F. Fateh, M. Ferdowsi, B. Lehman, L. H. Lewis, B. Mirafzal, and M. Saeedifard, "E-mobility—Advancements and challenges," *IEEE Access*, vol. 7, pp. 165226–165240, 2019, doi: 10.1109/ACCESS.2019.2953021.

[3] P. J. Ansell and K. S. Haran, "Electrified airplanes: A path to zero-emission air travel," *IEEE Electrific. Mag.*, vol. 8, no. 2, pp. 18–26, Jun. 2020.

[4] G. Sulligoi, A. Vicenzutti, and R. Menis, "All-electric ship design: From electrical propulsion to integrated electrical and electronic power systems," *IEEE Trans. Transport. Electrific.*, vol. 2, no. 4, pp. 507–521, Dec. 2016.

[5] M. B. Blarke, "Towards an intermittency-friendly energy system: Comparing electric boilers and heat pumps in distributed cogeneration," *Appl. Energy*, vol. 91, no. 1, pp. 349–365, Mar. 2012.

[6] D. Papadaskalopoulos, G. Strbac, P. Mancarella, M. Aunedi, and V. Stanojevic, "Decentralized participation of flexible demand in electricity markets—Part II: Application with electric vehicles and heat pump systems," *IEEE Trans. Power Syst.*, vol. 28, no. 4, pp. 3667–3674, Nov. 2013.

[7] J. M. Carrasco, L. G. Franquelo, J. T. Bialasiewicz, E. Galvan, R. C. PortilloGuisado, M. A. M. Prats, J. I. Leon, and N. Moreno-Alfonso, "Power-electronic systems for the grid integration of renewable energy sources: A survey," *IEEE Trans. Ind. Electron.*, vol. 53, no. 4, pp. 1002–1016, Jun. 2006.

[8] Y. M. Atwa, E. F. El-Saadany, M. M. A. Salama, and R. Seethapathy, ''Optimal renewable resources mix for distribution system energy loss minimization,'' *IEEE Trans. Power Syst.*, vol. 25, no. 1, pp. 360–370, Feb. 2010.

[9] D. Gielen, F. Boshell, D. Saygin, M. D. Bazilian, N. Wagner, and R. Gorini, ''The role of renewable energy in the global energy transformation,'' *Energy Strategy Rev.*, vol. 24, pp. 38–50, Apr. 2019.

[10] T. Arakawa, M. Takemoto, S. Ogasawara, K. Inoue, O. Ozaki, H. Hojo, and H. Mitani, ''Examination of an interior permanent magnet type axial gap motor for the hybrid electric vehicle,'' *IEEE Trans. Magn.*, vol. 47, no. 10, pp. 3602–3605, Oct. 2011.

[11] G. Pellegrino, A. Vagati, B. Boazzo, and P. Guglielmi, ''Comparison of induction and PM synchronous motor drives for EV application including design examples,'' *IEEE Trans. Ind. Appl.*, vol. 48, no. 6, pp. 2322–2332, Nov. 2012.

[12] Y. Yang, S. M. Castano, R. Yang, M. Kasprzak, B. Bilgin, A. Sathyan, H. Dadkhah, and A. Emadi, ''Design and comparison of interior permanent magnet motor topologies for traction applications,'' *IEEE Trans. Transport. Electrific*, vol. 3, no. 1, pp. 86–97, Mar. 2017.

[13] Arani and El-Saadany, ''Implementing virtual inertia in DFIG-based wind power generation,'' *IEEE Trans. Power Syst.*, vol. 28, no. 2, pp. 1373–1384, May 2013.

[14] R. Cardenas, R. Pena, S. Alepuz, and G. Asher, ''Overview of control systems for the operation of DFIGs in wind energy applications,'' *IEEE Trans. Ind. Electron.*, vol. 60, no. 7, pp. 2776–2798, Jul. 2013.

[15] S. Li, T. A. Haskew, R. P. Swatloski, and W. Gathings, ''Optimal and direct-current vector control of direct-driven PMSG wind turbines,'' *IEEE Trans. Power Electron.*, vol. 27, no. 5, pp. 2325–2337, May 2012.

[16] Y. Wang, J. Meng, X. Zhang, and L. Xu, ''Control of PMSG-based wind turbines for system inertial response and power oscillation damping,'' *IEEE Trans. Sustain. Energy*, vol. 6, no. 2, pp. 565–574, Apr. 2015.

[17] C. Bianchini, F. Immovilli, E. Lorenzani, A. Bellini, and M. Davoli, ''Review of design solutions for internal permanent-magnet machines cogging torque reduction,'' *IEEE Trans. Magn.*, vol. 48, no. 10, pp. 2685–2696, Oct. 2012.

[18] Y. Duan and D. M. Ionel, ''A review of recent developments in electrical machine design optimization methods with a permanent-magnet synchronous motor benchmark study,'' *IEEE Trans. Ind. Appl.*, vol. 49, no. 3, pp. 1268–1275, May 2013.

[19] T. P. M. Bazzo, J. F. Kolzer, R. Carlson, F. Wurtz, and L. Gerbaud, ''Multidisciplinary design optimization of direct-drive PMSG considering the site wind profile,'' *Electric Power Syst. Res.*, vol. 141, no. 4, pp. 467–475, Dec. 2016.

[20] T. de Paula Machado Bazzo, J. Fabio Kölzer, R. Carlson, F. Wurtz, and L. Gerbaud, ''Multiphysics design optimization of a permanent magnet synchronous generator,'' *IEEE Trans. Ind. Electron.*, vol. 64, no. 12, pp. 9815–9823, Dec. 2017.

[21] A. G. Sarigiannidis, M. E. Beniakar, and A. G. Kladas, ''Fast adaptive evolutionary PM traction motor optimization based on electric vehicle drive cycle,'' *IEEE Trans. Veh. Technol.*, vol. 66, no. 7, pp. 5762–5774, Jul. 2017.

[22] H. Chen and C. H. T. Lee, ''Parametric sensitivity analysis and design optimization of an interior permanent magnet synchronous motor,'' *IEEE Access*, vol. 7, pp. 159918–159929, 2019, doi: 10.1109/ACCESS.2019.2950773.

[23] Y. Okamoto, Y. Tominaga, S. Wakao, and S. Sato, ''Topology optimization of rotor core combined with identification of current phase angle in IPM motor using multistep genetic algorithm,'' *IEEE Trans. Magn.*, vol. 50, no. 2, pp. 725–728, Feb. 2014.

[24] T. Sato, K. Watanabe, and H. Igarashi, ''Multimaterial topology optimization of electric machines based on normalized Gaussian network,'' *IEEE Trans. Magn.*, vol. 51, no. 3, Mar. 2015, Art. no. 7202604.

[25] P. Gangl, S. Amstutz, and U. Langer, ''Topology optimization of electric motor using topological derivative for nonlinear magnetostatics,'' *IEEE Trans. Magn.*, vol. 52, no. 3, Mar. 2016, Art. no. 7201104.

[26] S. Doi, H. Sasaki, and H. Igarashi, ''Multi-objective topology optimization of rotating machines using deep learning,'' *IEEE Trans. Magn.*, vol. 55, no. 6, Jun. 2019, Art. no. 7202605.

[27] Y. Yamashita and Y. Okamoto, ''Design optimization of synchronous reluctance motor for reducing iron loss and improving torque characteristics using topology optimization based on the level-set method,'' *IEEE Trans. Magn.*, vol. 56, no. 3, Mar. 2020, Art. no. 7510704.

[28] R. Sutton and A. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 1998.

[29] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, and I. Antonoglou, ''Playing atari with deep reinforcement learning,'' in *Proc. NIPS Deep Learn. Workshop*, 2013, pp. 1–5.

[30] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, ''Mastering the game of go with deep neural networks and tree search,'' *Nature*, vol. 529, no. 7587, pp. 484–489, Jan. 2016, doi: 10.1038/nature16961.

[31] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, T. Lillicrap, K. Simonyan, and D. Hassabis, ''A general reinforcement learning algorithm that masters chess, shogi, and go through self-play,'' *Science*, vol. 362, no. 6419, pp. 1140–1144, Dec. 2018, doi: 10.1126/science.aar6404.

[32] C. E. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning*. Cambridge, MA, USA: MIT Press, 2006.

[33] N. Srinivas, A. Krause, S. Kakade, and M. Seeger, ''Gaussian process optimization in the bandit setting: No regret and experimental design,'' in *Proc. Int. Conf. Mach. Learn.*, 2010, pp. 1–10, doi: 10.1109/TIT.2011.2182033.

[34] N. Hansen and A. Ostermeier, ''Completely derandomized self-adaptation in evolution strategies,'' *Evol. Comput.*, vol. 9, no. 2, pp. 159–195, Jun. 2001.

[35] N. Hansen and A. Auger, ''Principled design of continuous stochastic search: From theory to practice,'' in *Theory and Principled Methods for the Design of Metaheuristics* (Natural Computing Series), Y. Borenstein and A. Moraglio, Eds. Berlin, Germany: Springer. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-642-33206-7_8, doi: 10.1007/978-3-642-33206-7_8.

[36] Y. Akimoto and N. Hansen, ''Diagonal acceleration for covariance matrix adaptation evolution strategies,'' *Evol. Comput.*, vol. 28, no. 3, pp. 405–435, 2019, doi: 10.1162/evco_a_00260.

[37] N. Hansen and A. Auger, ''A restart CMA evolution strategy with increasing population size,'' in *Proc. IEEE Congr. Evol. Comput.*, Oct. 2005, pp. 1–7, doi: 10.1109/CEC.2005.1554902.

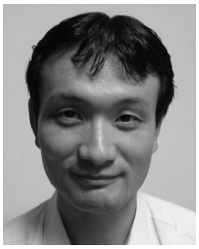
[38] Y. B. Dwianto, H. Fukumoto, and A. Oyama, ''On improving the constraint-handling performance with modified multiple constraint ranking (MCR-mod) for engineering design optimization problems solved by evolutionary algorithms,'' in *Proc. Genetic Evol. Comput. Conf.*, Jul. 2019, pp. 762–770, doi: 10.1145/3321707.3321808.

[39] A. H. Isfahani, B. M. Ebrahimi, and H. Lesani, ''Design optimization of a low-speed single-sided linear induction motor for improved efficiency and power factor,'' *IEEE Trans. Magn.*, vol. 44, no. 2, pp. 266–272, Feb. 2008.

[40] C. Lucas, Z. Nasiri-Gheidari, and F. Tootoonchian, ''Application of an imperialist competitive algorithm to the design of a linear induction motor,'' *Energy Convers. Manage.*, vol. 51, no. 7, pp. 1407–1411, Jul. 2010.

[41] T. Higuchi and S. Naoka, ''On the design of single-sided linear induction motors using a nonlinear optimization technique,'' *IEEJ Trans. Electr. Electron. Eng.*, vol. 114, no. 2, pp. 1171–1187, 1994, doi: 10.1002/eej.4391140209.

TAKAHIRO SATO received the B.E. and M.E. degrees in engineering and the Ph.D. degree in information science from Hokkaido University, Sapporo, Japan, in 2010, 2012, and 2015, respectively. Since 2015, he has been working as a Research Engineer with Toshiba Corporation, where he is currently with Toshiba Energy Systems & Solutions Corporation. His research interests include computational electromagnetism, design optimization, and evolutionary algorithms.



MASAFUMI FUJITA (Member, IEEE) received the B.E. and M.E. degrees in engineering from Kyoto University, Kyoto, Japan, in 1989 and 1991, respectively, and the M.E. degree from the University of Bath, in 2004. Since 1991, he has been working as a Research Engineer with Toshiba Corporation, where he is currently with Toshiba Energy Systems & Solutions Corporation. His research interests include computational electromagnetism and the development of large-capacity rotating machines.