Design Optimization Of Switched Reluctance Drives Using Artificial Neural Networks

A. Matveev 1, T. Undeland 2, R. Nilssen 2
1 Moscow Power Engineering Institute (Technical University)
Krasnokazarmennaja str., 13, 111250, Moscow E-250, Russia
phone: (+7) 095 974 14 47 - fax: (+7) 095 974 15 54
e-mail: alexei.matveev@vacon.com
2 Norwegian University of Science and Technology
7491, Trondheim, Norway
phone: (+47) 73 59 42 44 - fax: (+47) 73 59 42 79
e-mail: tore.undeland@elkraft.ntnu.no, robert.nilssen@elkraft.ntnu.no

Keywords
switched reluctance drives, design, artificial neural networks.

Abstract
The paper proposes a new method for the optimal design of switched reluctance drives. The method utilizes Artificial Neural Networks (ANNs) to provide good initial designs as a start points for iterative search techniques (optimization algorithms). A major advantage of the method is its quite short response time in obtaining the optimal design.

Introduction
Artificial neural networks are computing systems, whose structures are inspired by a simplified model of the human brain. They can learn from experimental data or from simulations, grasping the essential characteristics of a physical system. ANNs are finding applications in various fields of engineering, including adaptive pattern recognition, adaptive signal processing, adaptive dynamic modeling, adaptive control, expert systems etc. However there were few attempts in using the ANN for design optimization [1], [2]. Switched reluctance drives (SRD) are beneficial and practical because they have a high torque to weight ratio and are very reliable. This makes them ideal for applications in the industry, especially in the area of variable speed applications [3]. SR motors can have various topologies (number of phases, number of stator and rotor teeth, winding connections), the convertors for SR motors can also have various topologies depending on SR motor type and mode of operation. In addition all the processes in SR motor are highly non-linear. All this makes the task of SRD designer very complicated and time consuming.

In this paper, an approach to optimal design of SRD is proposed (fig. 1) where trained ANN is applied as an expert system which leads designer from given specification directly (in a matter in milliseconds) to "almost optimal" design. Standard optimization techniques (for example gradient-based) can be applied then to achieve optimal geometric shape and winding parameters of the motor. The commercially available software MATLAB® is used to develop the corresponding PC program.

The approach description
To provide design parameters near to optimal for any desired specific requirements the ANN should first achieve the input \( I_i \) to output \( O_i \) mapping function which describes the governing input-output relations in an electrical drive system. The inputs to the ANN include the desired characteristics of the drive (specification data in fig. 1). The outputs of the network are the motor and converter parameters such as motor and convertor topologies, stator diameter and stack length of the motor, number of turns in the stator coil etc. To achive the input-output mapping function the ANN has to be trained
using appropriate learning algorithm. The training procedure is characterized by the following main points:

- Creation of the training set (training patterns)
- Training of the ANN to match the defined training set
- Validation of the trained ANN

Training patterns can be prepared by numerical simulation or from already existing designs. Since the SR drive is new drive type there is no catalogue data available so we can use only the numerical simulation.

The ANN training and validation phases are realized with help of standard procedures of the Neural Network Toolbox of the MATLAB®/SIMULINK™ software [4]. Due to the advanced features of the software the program code can be very compact:

```matlab
% Creation of the new feedforward network of "5-200-6" topology, with tan-sigmoid transfer functions
net=newff(minmax(inputs),[200,6],{'tansig','tansig'},'traingd');
% Initializing weights
net = init(net);
% Training the network
[net,tr]=train(net,inputs,targets);
% Simulation (validation)
result=sim(net,data);
```

ANN of any type can not be universal – it is always destined for solving some limited tasks. In this paper the selected task is to build and to train the ANN for design of SR drives for industrial VSD-applications. The SR drives configurations are limited to traditional motor topologies (8/6, 6/4 and 12/8) with classical H-type converters and the motors designs are based on standard frames of induction motors. So for any given load, supply voltage, insulation class etc. we can make choice only between existing frames, i.e. the dimensions of active parts of the SR motor can be varied only step-wise. The task of the ANN is to find the best topology of motor, the most suitable frame (to choose the frame means to define stack length and stator diameter) and some initial geometric shape of motor cross-section and number of turns in the coils.
It is important to note, that ANN must be trained through sets of data which describe only optimal designs, then the ANN will provide better starting conditions (initial designs) – for any specification we will be able to get “almost optimal” design parameters. This means we should have not only software for simulation of SRD but also an optimization software.

The initial geometric shape of motor cross-section can be roughly defined by two parameters: \( d = D_r / D_s \) (see fig. 2) and \( \beta_s \). The parameters can be varied in the following ranges:

\[
0.4 < d < 0.7,
\frac{2 \cdot \pi}{m \cdot N_r} < \beta_s < \frac{\pi}{N_s},
\]

where \( m \) is number of phases, \( N_r \) – number of rotor poles, \( N_s \) - number of stator poles.

Number of turns in coil \( w \) is limited from below by necessity to produce required MMF and from above by slot area \( A_{\text{slot}} \) and admissible inverter current \( i_{\text{max}} \) (IGBT rating):

\[
w_{\text{min}} < w < w_{\text{max}} = \frac{2 \cdot T_{\text{max}} \cdot K_{\text{pls}} \cdot g}{L_{\text{st}} \cdot \mu_0 \cdot D_r},
\]

where \( T_{\text{max}} \) is maximum needed torque at the rated speed, \( K_{\text{pls}} \) – the coefficient of torque pulsations, \( L_{\text{st}} \) – stack length, \( g \) – air gap.

\[
w_{\text{max}} < \frac{J_{\text{fill}} \cdot A_{\text{slot}} \cdot \sqrt{3}}{2 \cdot i_{\text{max}}},
\]

where \( J_{\text{fill}} \) is the slot fill factor.

To find the optimal solution for given specification (as in fig. 1) we should carry out optimization for the following variable parameters:

- number of stator and rotor poles (motor topology)
- stack length and stator outer diameter (defined by the chosen frame)
- the ratio \( d = D_r / D_s \)
- stator pole arc \( \beta_s \)
- number of turns in coil \( w \)
- inverter IGBT rating

The optimization should be carried out for several speeds in the given working speed range. For every selected design an internal optimization have to be done to define the best control parameters (on angle, dwell interval, etc.).

As motor topology and frame dimensions can have only definite values the stochastic optimization techniques are preferable, for example Genetic Algorithms.

![Fig. 2. Crosssection of SR motor.](image)

![Fig. 3. Magnetization curves.](image)
Mathematical model of SRD

To estimate the drive performance and to calculate the objective function during the optimization process for every selected design we need mathematical model. In this paper two different models are presented. The first model does not take into account magnetic and electric coupling of SR motor phases, commutation processes in the convertor, mutual influence between supply net and the SRD and load changes. For every selected design the magnetization curves are calculated analytically using the method of Miller and McGilp [5]. End-effects are taken into account with help of expressions from [6]. The agreement between experimental and calculated data is relatively good (fig. 3). Main advantage of the model is its rapid operation. The core of the "rapid" SIMULINK™ model of the drive is shown in Fig. 4. The model core is realizing the well-know phase voltage equation

$$u = i * r + \frac{d\Psi}{dt}.$$  

The model of thermal processes is applied for calculation of hot-spot temperature rise.

![Fig. 4. The core of "rapid" SR motor SIMULINK™ model.](image)

ANN architecture selection and training

Various structures of the feedforward and recurrent ANNs, with various activation functions of neurons in the hidden layers were tested. There are no general way to determine the optimal number of neurons for a given system to obtain a good compromise between the network accuracy and convergence speed, so several variants were regarded. The appropriate number of data required to train the network is approximately the number of neural network weights times the inverse of the accuracy parameter $\varepsilon$ (Fig. 7), so the number of training patterns had to be large – up to several hundreds. It was found that the best generalization capabilities have two

![Fig. 5. Structure of generic feed-forward ANN (Multi-Layer Perceptron).](image)
ANN types – the well-known feed-forward (fig. 5) and radial basis function (RBF) networks. The RBF network may require more neurons than standard feed-forward back-propagation networks, but often they can be designed in a fraction of the time it takes to train standard feed-forward networks. They work best when many training vectors are available.

To achieve better performance for both selected ANN-types the normalization of input data and denormalization of targets should be done before training and during implementation. To normalize input values to the range [-1…1] the following expression can be used:

\[ x = \frac{2 \cdot (X - X_{\text{max}}) - (X_{\text{min}})}{X_{\text{max}} - X_{\text{min}}} \]

where \( x \) is normalized value, \( X \) - real value that is to be normalized, \( X_{\text{max}} \) and \( X_{\text{min}} \) - possible maximal and minimal values of the parameter.

The "rapid" SRD model was used for training the ANNs.

**A practical application of the proposed method**

A practical application of the proposed method is presented in Table 1. First the Genetic Algorithm was used to design the SR drive for pump application. The total design seized 12 minutes. Then the same task was solved with help of trained feedforward ANN. Near to optimal design was achieved in 0.1 sec. It can be seen that motor topology, frame type and IGBT rating are found correctly. The "convex hull" optimization task was solved by classical gradient-based method in 8 seconds.

<table>
<thead>
<tr>
<th>Initial data (specification)</th>
<th>Optimization with Genetic Algorithm</th>
<th>The proposed ANN-based approach</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Results</td>
<td>Calculation time</td>
</tr>
<tr>
<td>Load curve – square</td>
<td>Topology - 8/6</td>
<td>12 min</td>
</tr>
<tr>
<td>Rated speed – 3000 rpm</td>
<td>Casing type - 90S</td>
<td>ANN Simulation:</td>
</tr>
<tr>
<td>Rated power – 0.4 kW</td>
<td>( \beta_i = 22.4 )</td>
<td>Topology - 8/6</td>
</tr>
<tr>
<td>Speed range – 1:2</td>
<td>( d = 0.54 )</td>
<td>Casing type - 90S</td>
</tr>
<tr>
<td>Supply voltage – 220 V</td>
<td>IGBT =3.1 A</td>
<td>( w = 301 )</td>
</tr>
<tr>
<td>Cooling – self-ventilation</td>
<td></td>
<td>( \beta_i = 21.9 )</td>
</tr>
<tr>
<td>Insulation class – F</td>
<td></td>
<td>( d = 0.60 )</td>
</tr>
<tr>
<td>Duty – S1</td>
<td></td>
<td>IGBT =3.1 A</td>
</tr>
<tr>
<td>Optimized parameter –</td>
<td></td>
<td>Local optimization:</td>
</tr>
<tr>
<td>efficiency</td>
<td>Total calculation time</td>
<td>Topology - 8/6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Casing type - 90S</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( w = 290 )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( \beta_i = 22.5 )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( d = 0.52 )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IGBT =3.1 A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8 sec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Total calculation time</td>
</tr>
<tr>
<td></td>
<td></td>
<td>12 min</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8.1 sec</td>
</tr>
</tbody>
</table>

It should be noted that the total time for preparation of 200 training sets for teaching ANN can be approximately estimated as 30 hours. ANN training takes 30-40 seconds. The above two steps (especially the first one) are computationally extensive, however these are done off line.

**Training procedure with advanced model**

More advanced – the so-called "precise" model (fig. 6) takes into account all the above mentioned effects of coupling between phases and is suitable for "virtual tests" of SRD. The model is realized in the Power System Blockset of the MATLAB®/SIMULINK™ package [7]. The drawback of the
model is its slow operation. The ANN model can be coupled with the SRD model (fig. 7) and get permanent “on-line” training any time the SRD model is used by designer for various tasks. It will slow the work of the SRD model but the ANN characteristics can be permanently improved. Both "rapid" and "precise" models are included together with optimization algorithms into united program package called SRD-DASP. The package allows user to find optimal design of SRD for any desired specification. After the training the neural network can be used separately as a stand-alone expert system.

![Fig. 6. Block diagram of the "precise" model of SR drive.](image)

**Conclusion**

The proposed method utilizes ANNs for the optimal design of SR drives for variable speed applications. A trained neural network is used to find a set of near to optimal design parameters of electrical drive for specified task. Major advantages of the method are short response time in obtaining the optimal geometry and topology of the motor, high probability of finding global optimum and automated generation of training sets. Several ANN architectures were tested for the proposed task and it was found out that not only Multi-Layer Perseptrons but also Radial Basis Function Networks can show acceptable performance. The results of the example implemented in this paper confirmed excellent potential of ANNs in the tasks of optimal design.

**References**


