Speed Control of Switched Reluctance Motor Using ANFIS and GA

Prabaharan. A, Balamurugan. E
PG Scholar, Pandian Saraswathi Yadav Engineering College, Sivagangai, India
Assistant Professor, Pandian Saraswathi Yadav Engineering College, Sivagangai, India

ABSTRACT: In this paper, an innovative methodology for Switched Reluctance Motor (SRM) drive control using ANFIS and GA based optimization techniques are presented. The proposed system uses individual and social intelligences, so that it can search responses among local optimums of the problem adaptively. This matter is done by applying the proposed system to a multi-objective function including both speed error and torque ripple. This controller is implemented for an 8/6, 4-kW SRM. The simulation and experimental results confirm the improved performance of GA in comparison with the ANFIS based optimization technique. Excellent dynamic performance, reduced torque ripple and current oscillation can be achieved by using GA.

KEYWORDS: ANFIS and GA based optimization, Speed control, Switched reluctance motor.

I. INTRODUCTION

In recent years, there is a growing interest for using the SRM, because of low cost, simple rugged structure, and relatively high torque–volume ratio and low maintenance cost. They are the best candidate for direct-drive applications because of their ability to generate high torques at very low speed in comparison with other conventional motors [2,3]. Nevertheless, it suffers some drawbacks such as; high torque ripple and significant acoustic noise as well as speed oscillations. In order to resolve these problems, albeit various advanced control systems are proposed [4,5], PID controllers are most common controller for industrial applications [6] in spite of some disadvantages such as; tuning coefficients. Hence many researches have been presented about optimization of the controllers. Optimization methods are used in many complex engineering problems [7]. Among various optimization methods, bio-inspired methods are developed more than classic techniques. Genetic algorithm (GA) is one of the first bio-inspired methods [8]. Based on its demonstrated ability to reach near-optimum solutions in complex problems, the GA technique is used for many applications [9]. In [10], GA is applied for optimal design of PI speed controller coefficient with considering minimum torque ripple as a cost function in simulation tests. Despite its benefits, GA requires long pre-cissing time to find optimum solution. Furthermore, sometimes a local minimum instead of global minimum may be introduced by GA.

Recently, some new techniques inspired of different natural processes are proposed such as; particle swarm optimization (PSO) algorithm [11], invasive weed optimization (IWO) algorithm [12], artificial bee colony (ABC) algorithm [13], and artificial immune (AI) algorithm [14]. Moreover, some of the recently pre-sented advanced algorithms are; Directed Searching Optimization algorithm (DSO), which is proposed to solve constrained optimization problems [15]. In [16], a control mechanism based PI control-ler for speed control of Switched Reluctance Motor (SRM) with torque ripple reduction using non-dominated sorting genetic algorithm has been presented. However, the proposed method has been only validated by simulation.

II. ADAPTIVE NEURO FUZZY INFERENCE SYSTEM (ANFIS)

The acronym ANFIS derives its name from adaptive neuro-fuzzy inference system. Using a given input/output data set, the toolbox function anfis constructs a fuzzy inference system (FIS) whose membership function parameters are tuned (adjusted) using either a backpropagation algorithm alone or in combination with a least squares type of method. This adjustment allows your fuzzy systems to learn from the data they are modeling.
Basic Structure of artificial neural network:

input layer:
   The bottom layer is known as input neuron network in this case x1 to x5 are input layer neurons.

Hidden layer:
   The in-between input and output layer the layers are known as hidden layers where the knowledge of past experience / training is the

Output Layer:
   The topmost layer which give the final output. In this case z1 and z2 are output neurons.

Network architectures:

1). Single layer feedforward networks:
   In this layered neural network the neurons are organized in the form of layers. In this simplest form of a layered network, we have an input layer of source nodes those projects on to an output layer of neurons, but not vise-versa.

2). Multilayer feedforward networks:
   The second class of the feed forward neural network distinguishes itself by one or more hidden layers, whose computation nodes are correspondingly called neurons or units. The function of hidden neurons is intervene between the external i/p and the network o/p in some useful manner. The ability of hidden neurons is to extract higher order statistics is particularly valuable when the size of i/p layer is large. The i/p vectors are feedforward to 1st hidden layer and this pass to 2nd hidden layer and so on until the last layer i.e. output layer, which gives actual network response.
3). Recurrent networks:

A recurrent network may consist of a single layer of neurons with each neuron feeding its output signal back to the inputs of all the other neurons. Network may have hidden layers or not.

![Figure 4: Single node with feedback to itself](image)

**III. FUZZY LOGIC SYSTEM (FLS)**

FLS can be defined as the non linear mapping of an input data set to a scalar output data. The richness of the FL is that there are enormous numbers of possibilities which lead to lots of different mappings. This richness does require a careful understanding of FL and the elements that comprise a FLS. A FLS consists of four components: fuzzifier, rules, inference engine and defuzzifier.

The fuzzifier maps crisp numbers into fuzzy sets. It is needed in order to activate rules which are in terms of linguistic variables, which have fuzzy sets associated with them. The inference engine of the FLS maps input fuzzy sets into output fuzzy sets. It handles the way in which rules are combined, just as humans use many different types of inferential procedures to help understanding things or to make decisions. In many applications, crisp number must be obtained as the output of a FLS. The defuzzifier maps output sets into crisp numbers. The algorithm of above process is explained below.

![Figure 5: Basic structure of a fuzzy logic control](image)

**Fuzzy Rules:**

In a FLS, a rule base is constructed to control the output variable. The decisions are made by forming a series of rules that relate the input variables to the output variables using simple IF-THEN statements. The IF-THEN rule statements are used to formulate conditional statements that comprise fuzzy logic. IF part of the rule is called antecedent and THEN part of the rule is called consequent. A single fuzzy if then rule can be represented in the following manner.

If (x is A) then (y is B)

where

A and B are linguistic values defined by fuzzy sets in the ranges (universes of discourse) X and Y respectively, x and y are the fuzzy variables, “x is A” is called the antecedent and “y is B” is called consequent.
IV. GA Based optimization

Genetic Algorithms are the heuristic search and optimization techniques that mimic the process of natural evolution. Thus genetic algorithms implement the optimization strategies by simulating evolution of species through natural selection.

GA operators
1. Selection
2. Crossover
3. Mutation

1. Selection:
   - There are different techniques to implement selection in Genetic Algorithms.
   - They are:
     - Tournament selection
     - Roulette wheel selection
     - Proportionate selection
     - Rank selection
     - Steady state selection, etc

2. Crossover:
   - The most popular crossover selects any two solutions strings randomly from the mating pool and some portion of the strings is exchanged between the strings.

3. Mutation:
   - Mutation is the occasional introduction of new features into the solution strings of the population pool to maintain diversity in the population.
V. SIMULATION CIRCUIT DIAGRAM

Figure 7 Simulink circuit diagram.

VI. RESULTS

Network training tool:

Figure 8 training tool
FIS rule viewer:

Figure 9 Rule viewer.

Figure 10 Experimental results of rotor speed using ANFIS

Figure 11 Experimental results of rotor speed using GA.

VII. CONCLUSION

In this work, an optimization method based on ANFIS and GA has been presented. These techniques have been used to optimize the coefficients of speed controller for SRM drive by resolving a multi-objective function. In order to evaluate the better results, both methods have been compared with each other and the results show superiority of the GA over the other.

On the other hand, simulation and experimental results confirmed the improved performance of the GA for speed control.
control of the SRM, the proposed optimization techniques can be easily adopted for other different applications.

REFERENCES