

1-1-2009

An Intelligent Decision Support System for Design of Brushless Direct Current Motors

Vahab Akbarzadeh
Ryerson University

Follow this and additional works at: <http://digitalcommons.ryerson.ca/dissertations>



Part of the [Computer Engineering Commons](#)

Recommended Citation

Akbarzadeh, Vahab, "An Intelligent Decision Support System for Design of Brushless Direct Current Motors" (2009). *Theses and dissertations*. Paper 1072.

This Thesis is brought to you for free and open access by Digital Commons @ Ryerson. It has been accepted for inclusion in Theses and dissertations by an authorized administrator of Digital Commons @ Ryerson. For more information, please contact bcameron@ryerson.ca.

AN INTELLIGENT DECISION SUPPORT SYSTEM FOR DESIGN OF BRUSHLESS DIRECT CURRENT MOTORS

by

Vahab Akbarzadeh

BSc, Shahid Beheshti University, Iran, 2006

A thesis

Presented to Ryerson University

In partial fulfillment of the

Requirements for the degree of

Master of Science

In the program of

Computer Science

Toronto, Ontario, Canada, 2009

© Vahab Akbarzadeh, 2009

I hereby declare that I am the sole author of this thesis or dissertation.

I authorize Ryerson University to lend this thesis or dissertation to other institutions or individuals for the purpose of scholarly research.



I further authorize Ryerson University to reproduce this thesis or dissertation by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

AN INTELLIGENT DECISION SUPPORT SYSTEM FOR DESIGN OF BRUSHLESS DIRECT CURRENT MOTORS

Vahab Akbarzadeh

Msc, Computer Science, Ryerson University, 2009

Abstract

Brushless DC (BLDC) motors are among the most widely used electrical motors. Design of a BLDC motor is the most fundamental problem when dealing with the BLDC motors. This thesis presents an intelligent decision support system that can be used to design BLDC motors. A hybrid approach, that includes an object oriented paradigm using frames and procedural attachments together with a rule based mechanism, is used to build the knowledge base of the proposed architecture. The design strategy is implemented using a rule-based successive iterative method. An evolutionary fuzzy system was used to derive the modification rules of the system. The antecedent and consequent of each fuzzy modification rule was encoded as the individual of an evolutionary system. The evolutionary system evolves the set of modification rules to find a set of optimized rules. The proposed system developed design which had superior efficiency, weight and motor constant compared to design developed using the conventional design method.

Acknowledgments

I would like to express my sincere gratitude for the diligent assistance of my advisor, Dr. Alireza Sadeghian, in helping me complete my Master degree. He has shown continued support and provided valuable advice throughout my graduate studies.

I would also like to extend my gratitude for the technical input and guidance of reading committee as well.

To my parents, to whom I owe everything.

Table of Contents

Abstract	III
Acknowledgments.....	IV
List of Tables	IX
List of Figures	X
1 Introduction	1
1.1 Brushless DC motors.....	1
1.2 Design issue.....	2
1.3 Existing approaches.....	3
1.4 Decision support systems	4
1.5 Proposed approach	6
1.6 Methodology	6
1.7 Thesis outline	7
2 Background	9
2.1 Design of BLDC motors	9
2.1.1 Selection of the material	9
2.1.2 Computation of geometric properties	10
2.2 Intelligent decision support systems	12
2.3 Fuzzy system	14
2.4 Evolutionary Computation	18
2.5 Evolutionary fuzzy system.....	21
3 Decision Support Systems for Design – An Overview	23

3.1	Application of knowledge based expert systems in the design of electromagnetic devices	24
3.2	Application of soft computing approaches in the design of electromagnetic devices ...	28
4	Methodology	36
4.1	IDSS structure	36
4.2	KBS architecture	37
4.3	Evolutionary fuzzy system	40
4.3.1	Fuzzy IF-THEN rules	40
4.3.2	Rule presentation	43
4.3.3	Fitness function	45
4.3.4	Selection	47
4.3.5	Mutation	48
4.3.6	Crossover	48
4.3.7	Membership function	49
4.3.8	Design algorithm	50
5	Results and Discussion	52
5.1	Modification rule derivation subsystem	52
5.1.1	Motor specifications	52
5.1.2	Casebase	54
5.1.3	Modification rules structure	57
5.1.3.1	Antecedent of fuzzy modification rules	59
5.1.3.2	Consequent of fuzzy modification rules	61
5.1.3.3	Evolutionary System	62
5.2	Knowledge base system implementation	64
5.3	Results	68

6 Conclusion	71
Appendix A. Design equations for the radial flux BLDC motor [36]	73
Appendix B. Determination of hysteresis and Eddy current coefficients for calculation of the total core loss of the motor [83]	78
Appendix C. List of rules in the KB of the system.....	80
References	83

List of Tables

Table 1. Heuristic knowledge for permanent magnet selection and the recommended alternative material for different DC motor sizes [34].	10
Table 2. A sample design rule.....	40
Table 3. Design specifications	53
Table 4. Parameter range for casebase generation.....	55
Table 5. slot/pole combinations for three phase motors [71].....	55
Table 6. Predicting attributes used in the antecedent of modification rules	59
Table 7. The actions in the consequent of fuzzy modification rules	61
Table 8. Evolutionary system parameters.....	62
Table 9. Some of the derived modification rules	63
Table 10. Results.....	68
Table 11. Two modification rules in conventional and fuzzy KBSs	69

List of Figures

Figure 1. Cross-sectional view of a BLDC motor showing some of the geometrical parameters	11
Figure 2. BLDC motor design steps	12
Figure 3. Membership functions showing three linguistic values of <i>Height</i>	15
Figure 4. Fuzzy inference system	17
Figure 5. Structure of an evolutionary algorithm.....	20
Figure 6. User interaction with the IDSS.....	36
Figure 7. Design steps.....	37
Figure 8. Knowledge Base (KB) structure of the system	38
Figure 9. Dataset S.....	41
Figure 10. The fuzzy reasoning method	43
Figure 11. A sample rule of the system	45
Figure 12. Steps of the evolutionary rule derivation system	47
Figure 13. Mutation	48
Figure 14. Crossover	49
Figure 15. Membership functions of predicting attributes.....	50
Figure 16. Membership function definitions for predicting attributes.....	60
Figure 17. Application of a fuzzy modification rule on an unsatisfactory design.....	67

1 Introduction

1.1 Brushless DC motors

Application of brushless DC (BLDC) motors has increased significantly over the past decades. This is mainly due to high reliability and efficiency of BLDC motors as well as their ability to reach very high speed. Brushless DC motors are rotational brushless permanent magnet motors which are driven by DC current and use electronic control systems instead of the brushes that are usually used in conventional DC motors. Compared to conventional commutator type DC motors, both types have straight line speed torque line, but BLDC motors are more efficient, need less maintenance and have longer life span. On the other hand, the control system of BLDC motors needs a rotor positioning mechanism, and the magnets might gradually demagnetize [19], [77]. BLDC motors have been used in a wide variety of applications from industrial to household devices. Typical examples include industrial tools (pumps, compressors), power tools (drills, hammers), transportation (electric vehicles), and household devices (electric shavers, mixers) [28]. Small BLDC motors have also been extensively used in precision devices including medical equipment, computer drives, hard disks, and players.

BLDC motors operate based on electromagnetic principles where the interaction between the stator current and the magnetic field of permanent magnets produces the force which drives the motor. BLDC motors use Permanent Magnet (PM) material for excitation instead of electromagnetic excitation. This substitution brings the benefit of higher efficiency, higher torque per volume, better dynamic performance, and simplification of construction and maintenance. The permanent magnetic materials are getting stronger and their prices are dropping which make PM motors more popular. In BLDC motors the magnet is mounted on the

rotor and the current carrying coils are mounted on the stator, so there is no need for brushes to change the current of the coils. This will reduce the maintenance cost by 90%, because most of the times the maintenance routines are related to the brushes [28]. As the coils are mounted on the stators, a cooling system can be easily used to transfer the heat and increase the current density of the coils.

BLDC motors can be classified from different perspectives. From a structural point of view, BLDC motors can be classified as: a) Radial Flux Motors and b) Axial Flux Motors. As their name suggests, in the radial flux motors the flux crosses in the radial direction from the rotor to the stator, while this direction is axial in the axial flux motors. The radial flux motors can be further divided into: a) Inner Rotor and b) Outer Rotor structures. In both designs the windings are placed on the stator and the magnets are bonded to the rotor. Comparing the inner rotor and outer rotor motors, the outer rotor has more magnetic material so the total flux is higher and lower energy magnetic materials can be used. Besides, these types of motors have higher rotor inertia, so they can be used in applications where this higher inertia is beneficial (e.g. cooling fans). On the other hand, the inner rotor motor has small inertia so the acceleration is faster [83]. Also, the distributed winding provides a smoother operation and better copper utilization. We focus on the design of inner rotor radial flux motors in this thesis, but the principles can be easily modified for alternative structures.

1.2 Design issue

Design of BLDC motors is the most important and fundamental problem when dealing with BLDC motors. Motor design is a sample of routine design process, where the relationship between the functional requirements and the structure is known. Also, the parameters which can

be varied are also known. The goal of a routine design is to find the optimal value for the variable parameters in an attempt to optimize one or more than one objective values.

Motor design process can be seen as an optimization problem which is subject to a number of constraints. The goal of the design process is to maximize, *e.g.* efficiency, or minimize, *e.g.* weight, some objectives at the same time. This objective can be formalized as a vector-optimization problem: $F(x) = (f_1(x), f_2(x), \dots, f_n(x))$, where $f_i(x)$ represents an objective of the design to be optimized, and x is the set of variable design parameters. The final goal of the design process is to find the extremum of the $F(x)$ function [28]. This optimization process is also subject to a number of constraints. These constraints are defined as $x_{min} < x < x_{max}$ and $g_i(x) > 0$, where x_{min} and x_{max} are the lower and the upper bounds of design parameters and $g_i(x)$ represents the performance requirements which have been formulated as an inequality constraint.

1.3 Existing approaches

Traditional approaches for the design of electrical motors were based on the implementation of a physical model of the motor. This approach had two drawbacks. First, making a physical model is very expensive and time consuming. Second, when a physical model is made which does not satisfy the required specifications, the designer cannot modify the developed model to comply with the given requirements. With the advent of the computers, designers can make a virtual model of the motor in the computer. This approach is much faster and less expensive and provides an easier way to modify the designs either manually or with the help of optimization methods [81].

Most designers rely on an iterative design process for the design of electrical motors. In this method, the designer adjusts the design parameters recursively until the design constraints are

satisfied. In most cases, for a given set of specifications, different designers achieve different design details which are attributed with the experience and expertise of the designer in modifying a specific set of design parameters.

The experience related to the design of BLDC motors can be integrated into a computerized system in different ways. The first approach is to extract the design knowledge from an expert designer and embed this knowledge into a knowledge base. The other approach which uses the search abilities of the computers uses an automated method to extract knowledge from a set of examples of the previous cases of the design to induce knowledge and automatically construct a knowledge base.

It is not usually a good idea to integrate a fully automated optimization method in the design system as a replacement for the human designer, because the design process involves many delicate compromises and judgments that a fully automated system is not capable of accomplishing. Therefore, usage of an automated method as an assistant to the expert designer seems to be the most proper way to incorporate the human expert knowledge and the automatic optimization method at the same time. Intelligent Decision Support System (IDSS) is a good candidate for a semi-automated system that can help the motor designer during the design process. In this approach, the system uses the automated optimization method to provide a suggestion to the designer on how to improve the performance of the design.

1.4 Decision support systems

Decision Support System (DSS) is a computer based information system which is developed to help the decision maker during the decision making process. DSS brings the human judgment and computerized information to semi-structured or non-structured situations [79]. IDSS was

proposed as an extension to the concept of DSS. IDSS incorporates Artificial Intelligence (AI) techniques in order to provide human like behavior. The AI techniques help the DSS to deal with complex and uncertain environment and be able to respond to unseen situations without human intervention. Evolutionary Fuzzy Systems (EFS) is a branch of Soft Computing (which itself is a branch of AI) that could be incorporated into an IDSS. EFSs come from the combination of evolutionary algorithms and fuzzy systems.

Evolutionary Algorithms (EA), [29,39] inspired from the principles of natural evolution dates back to the principle of Darwin's theory of evolution "The survival of the fittest". This field of study has become an active research area in soft computing with enormous applications. Although EA is the simplified model of its biological counterpart, it is a powerful search and optimization technique.

A Fuzzy System (FS) is a rule-based system which is based on the concept of fuzzy logic instead of Boolean logic. Fuzzy logic tries to resemble human reasoning method in the way that each logical statement can be partially true or false. This approach is an extension of the conventional Boolean logic where every statement could be either true or false.

Evolutionary methods are well-known as strong search methods which can do well in complex search environments, and fuzzy systems can provide flexible inference mechanisms in uncertain and imprecise environments. A possible combination of these two methods is feasible through the concept of Evolutionary Fuzzy System (EFS). EFS is a fuzzy system which uses evolutionary methods to learn and tune different parameters of the fuzzy system [16].

1.5 Proposed approach

The proposed IDSS incorporates EFS to suggest intelligent adaption to an unsatisfactory BLDC motor design. The system starts when the user gives the set of specification to the system. These specifications are given as input to the EFS to derive a set of modifications to improve the performance of an unsatisfactory design. The input specifications are also used to generate an initial design using the conventional design strategy. The developed initial design will be tested against the given specifications to see if the design satisfies the specifications or not. If the design does not satisfy the requirements the modification proposed by the EFS are applied on it to generate new models. The system will repeat among the test and modify parts until a set of satisfactory designs are generated. At the end of the design session, the system has generated a number of models and the system designer can select the one which satisfies his needs as much as possible.

1.6 Methodology

Sadeghian et al. [68] have developed a knowledge based expert system to design inductors. The system imitated the recursive process used by expert designers to design inductors. The principles of that system were used here to develop an intelligent decision support system to design BLDC motors.

We propose an intelligent decision support system which can provide alternative choices for the design parameters of the BLDC motor to the designer. The designer can make the final decision among the proposed models based on the importance of different criteria of the goal motor. The knowledge in the underlying system is represented as rules and procedural attachments which are understandable for an expert designer.

In this system, first an initial design is developed using the conventional design strategy. Then the quality of the developed design is compared to the requested criteria. Next, the developed design goes under a set of modification rules which changes the design parameters. Afterward, iteration starts between the test and the modification phases until the developed design can satisfy the requested criteria. Designs that can satisfy the requirements as much as possible are given to the designer as the output of the system.

An evolutionary fuzzy system was developed to derive the modification rules used in the design process. The modification rules have different objectives, such as improving the efficiency, improving the motor constant or decreasing the weight of the motor. These rules are derived from a casebase of possible designs. The fuzzy component of the evolutionary fuzzy system gives the capability to incorporate linguistic values in the modification rules so the final derived rules are more comprehensible for the designer. On the other hand, the evolutionary computation methods are known to be a strong search algorithm. So the final derived rules are more accurate and understandable.

1.7 Thesis outline

This thesis proposes the use of an intelligent decision support system for design of brushless DC motors. Section 2 provides the background knowledge necessary for the rest of the thesis. Section 2.1 provides a brief description about the conventional design methodology of BLDC motors. The methodology proposed in this thesis is based on four domains: intelligent decision support systems, fuzzy logic, evolutionary algorithms, and evolutionary fuzzy systems. The basic notions regarding the concept of intelligent decision support system is given in section 2.2. Fuzzy systems and evolutionary computation are two main concepts of soft computing domain

which are explained in section 2.3 and 2.4 respectively. Section 2.5 introduces evolutionary fuzzy systems which comes from the fusion of evolutionary and fuzzy domains.

Section 3 provides the literature review of decision support systems applied to the design of electromagnetic devices. Intelligent methods applied in the field of electrical machine design can be categorized under the concepts of knowledge based expert systems and soft computing approaches. Section 3.1 presents the application of Knowledge Based Expert Systems in the design of electromagnetic devices. Section 3.2 presents the application of Soft Computing approaches in the design of electromagnetic devices.

Section 4 presents the methodology proposed for the intelligent decision support system. The knowledge base structure of the IDSS is presented in section 4.1. The Evolutionary Fuzzy System which is used as the optimization method of the IDSS is described in section 4.2.

Section 5 presents a sample application of the system on the design of a sample BLDC motor. In section 5.1 the modification rule derivation subsystem is explained which extracts modification rules from a casebase of BLDC motor designs using the methodology explained in section 4.2. The knowledge based system is developed in section 5.2, which includes the modification rules derived by the EFS, besides the design and test rules. The developed KBS is applied on a sample case and the final results are presented and compared to the conventional design method in section 5.3.

Section 6 concludes the thesis, providing the advantages of the system compared to other available methods. Also, some possible directions for future work are mentioned in this section.

2 Background

2.1 Design of BLDC motors

In every design problem, the values for a number of variable parameters need to be determined. This value determination is subject to a number of constraints which represents the feasibility and the quality of the proposed model. The design process of BLDC motors consists of two major steps: selection of the material and computation of geometric properties.

First, the specification of the requested design is given to the designer. These specifications are the constraints that the proposed model should satisfy. In the simplest case, these parameters include the available input voltage and the desired output power and speed. The maximum allowed physical properties of the motor are also given. Using this information the designer can initiate the design process.

2.1.1 Selection of the material

The proper material for the permanent magnet (PM) and steel is to be selected first. The designer does not always have full control over this step since manufacturing limitations (*e.g.*, punchability of steel sheets) and availability of materials influence the range of materials that can be used. Cost is another important concern throughout the whole design process. The final product cost is representative of the construction and material costs.

Motor yoke is used to attach the permanent magnet to the shaft and to keep the conductors in place. Soft metallic material, such as silicon iron electrical steel, is mainly used for motor yoke fabrication. The steel can be in a solid, laminated or pressed soft iron forms. Physical (*e.g.*, lamination stacking factor) and magnetic (*e.g.*, saturated flux density, hysteresis energy loss,

permeability) properties are the main criteria used for steel selection. Other operation parameters such as size, frequency of operation and the length of operation periods also have effect on the type of steel recommended for each motor [34]. For example, the pressed soft iron forms provide flexibility of design and dramatically reduce the amount of scrap, but the laminated steel has lower total core loss in a wider range of operation frequencies.

Possible materials for the permanent magnet are considered based on their physical (*e.g.*, density) and magnetic (*e.g.*, average recoil permeability) properties. Stress effects and applicability under high temperature circumstances are the other important properties of PMs [34]. Table 1 represents heuristic knowledge that can be used for PM selection in BLDC motors.

Table 1. Heuristic knowledge for permanent magnet selection and the recommended alternative material for different DC motor sizes [34].

Application	Recommended material	Primary reason for selection	Alternative material	Condition or reason favoring selection of alternative material
Small DC motor	Bonded Ferrite	<ul style="list-style-type: none"> - Shape favors fabrication - Adequate magnetic energy - Low cost 	<ul style="list-style-type: none"> - Bonded NdFeB - Sintered ferrite 	<ul style="list-style-type: none"> - Higher magnetic energy
Large DC motor	SmCo (Samarium Cobalt)	<ul style="list-style-type: none"> - Maximum energy per unit volume - Resistant to demagnetization and corrosion - Expensive material 	NdFeB	<ul style="list-style-type: none"> - Higher magnetic energy - Less expensive - Low temperature operation - Corrosion prone

2.1.2 Computation of geometric properties

The geometric parameters of the motor including rotor radius size, stator yoke width, tooth width, and magnet length, are calculated using conventional design method [36]. In order to calculate some of the geometric parameters, the magnetic field distribution inside the motor

needs to be modeled. The modeling can be done using the magnetic circuit or numerical field computation (finite element) approaches. Although the finite element method produces more accurate results, the magnetic circuit technique is much faster and produces results that are acceptable in most engineering applications. Meanwhile, the electrical parameters (*e.g.*, number of turns per slot) are to be determined. The cross-sectional view of a BLDC motor is presented in Figure 1.

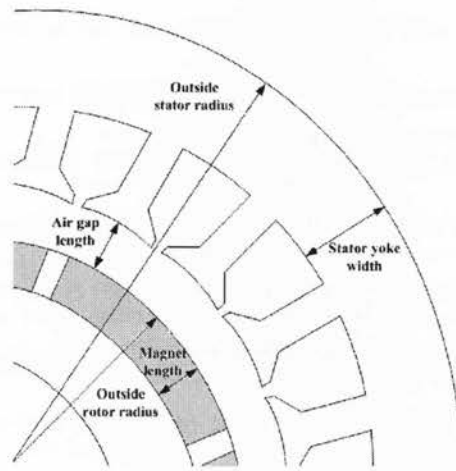


Figure 1. Cross-sectional view of a BLDC motor showing some of the geometrical parameters

When all the parameters have been decided upon, the quality of the design is measured using performance indices such as motor constant and efficiency index. The motor constant index, K_m , is a performance criterion which is proportional with the amount of torque produced in a motor and inversely proportional with the cost of the torque production, *i.e.*, energy dissipated in coils and is defined as [35]:

$$K_m = \frac{T}{\sqrt{I^2 R}} \quad (1)$$

where T is torque; I is the total current in slots, and R is the total slot resistance. Efficiency is the other fundamentally important criterion defined as [83]:

$$\eta = \frac{T\omega_m}{T\omega_m + P_r + P_{cl} + P_s} \times 100\% \quad (2)$$

where ω_m is mechanical speed, and P_r , P_{cl} , P_s are ohmic motor loss, core loss, and stray loss respectively. A simplified version of the conventional BLDC motor design process is shown in Figure 2.

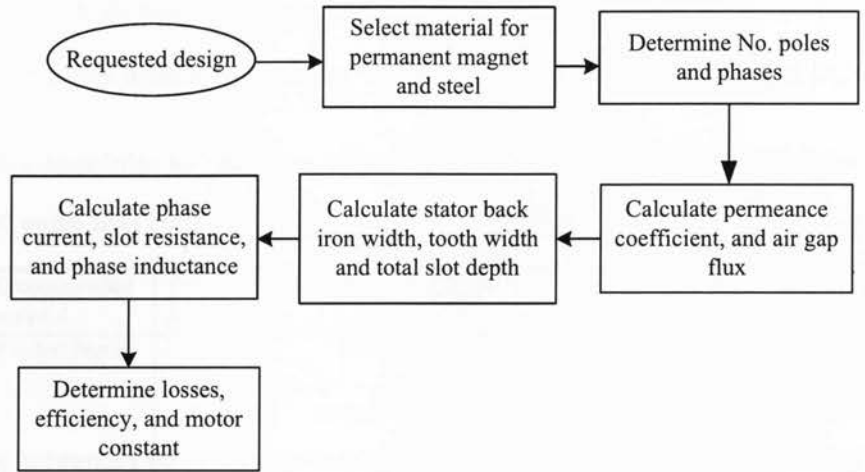


Figure 2. BLDC motor design steps

2.2 Intelligent decision support systems

Decision making could be considered as a process of selecting between alternative choices for the purpose of attaining a goal or goals. Simon proposed three phases of intelligence, design, and choice as a model for decision making [73]. In the intelligence phase, the environment is being searched to identify a problem situation. When the problem is identified, the possible problem models are generated, developed and analyzed in the design phase. During the choice phase,

different solutions for the model are evaluated using the what-if, goal-seeking and sensitivity analysis, so that a solution to the problem can be recommended. Note that the decision making is an iterative process in which the decision maker may go back and forth between the different phases of the process.

A Decision Support System (DSS) is a computer based information system which is designed to support a decision maker to solve a particular problem in an attempt to relax cognitive, temporal, spatial and/or economic limits on the decision maker [24,79]. The system will help the decision maker to be more innovative and productive, the solution to be more agile and reputable and the stakeholders to be more satisfied [24]. The DSS may support one or more than one of the previously mentioned decision phases. It might help the decision maker to collect the relevant data during the intelligence phase, better understand the important parts of the problem during the design phase, or better identifies the correlation between different proposed solutions during the choice phase [79]. Note that in all these situations it is the decision maker who makes the final and critical decisions.

Structurally, a DSS is composed of the following components: (1) a user interface, which is responsible for communication between the user and the system. It is usually composed of a language system which accepts the command from the user and a presentation language which includes all the messages that the system can emit. (2) An information and knowledge part which is a repository of all the information and knowledge that the system has maintained for the problem solving purpose. (3) A problem processing system, which is the active component of the system. The problem processing component uses the other components to solve the problem during the decision making process [10,24].

The concept of Intelligent Decision Support Systems (IDSS) was first proposed under the concept of Knowledge Based Decision Support Systems (KB-DSS) [40]. The KB-DSS was defined as the second generation of DSS which has the ability to acquire, maintain, and present knowledge in various ways. It also has the ability to reason from the acquired knowledge to solve the problems, and interact intelligently with the decision maker.

The concept of KB-DSS was later extended to incorporate other artificial intelligence techniques into a DSS, for the purpose of developing a system which provides human like behavior. An IDSS can be defined as an extended DSS which has the ability to: (a) learn from experience; (b) respond to new information without human intervention; (c) understand the relative importance of different elements in a decision making environment; (d) deal with complex and uncertain situations; (e) apply the acquired knowledge as a recommendation to the decision maker. [3,18,63].

2.3 Fuzzy system

A fuzzy system is a rule-based system which is based on the concept of fuzzy logic instead of Boolean logic. Fuzzy logic is a form of multi-valued logic which tries to resemble human reasoning method in the way that each logical statement can be partially true or false. This approach is an extension of the conventional Boolean logic where every statement could be either true or false. The concept of fuzzy logic was first proposed by Lotfi A. Zadeh in his seminal paper about fuzzy sets [86].

Fuzzy set is a special kind of set that has fuzzy boundaries instead of crisp boundaries used in conventional sets. More precisely, the boundary which defines if a member belongs or does not belong to the set changes gradually. These “fuzzy” boundaries help the system to better model

the imprecision and uncertainty that exists in human abstract thoughts [44]. These fuzzy boundaries are defined through membership functions.

Membership functions (or characteristic functions) are defined on the universe of discourse (or space of objects). Each membership function takes an element from universe of discourse as an input and returns a value between $[0,1]$ which represents the membership grade of that element. In this approach, zero indicates that the element does not belong to the set and one indicates that the element is a member of the set. In classical binary logic the membership function can only return values 0 and 1, but in fuzzy logic intermediate values are also taken into account to represent partial fulfillment of the condition. The application of a membership function on an element from the universe of discourse is usually referred to as fuzzification. Membership functions usually have smooth shapes such as triangular, trapezoidal or bell-shaped [60].

The concept of linguistic variables was also developed by Lotfi A. Zadeh [85] to make a more human friendly medium for interaction between humans and complex systems. Each linguistic variable is composed of a name, a set of linguistic values and a set of membership functions defining each linguistic value. For example, the linguistic variable *Height* could have linguistic values *Short*, *Medium* and *Tall* where the membership functions are represented in Figure 3.

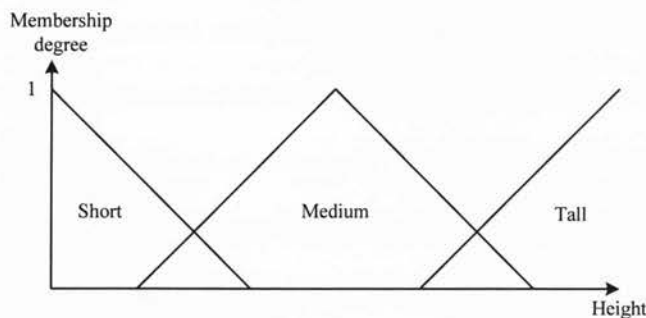


Figure 3. Membership functions showing three linguistic values of *Height*

IF-THEN rules are one of the most successfully used methods for representation and integration of human knowledge into a computational system. The rules have the form IF (antecedent)-THEN (consequent), where the satisfaction of the antecedent gives rise to the execution of the consequent. In a simple fuzzy IF-THEN rule, the antecedent and the consequent are propositional statements of the form " x is A ", where x is a linguistic variable defined on the universe of discourse and A is a linguistic value. The propositions in the antecedent could be connected with each other using logical operators. The most common logical operators are $\{AND, OR, NOT\}$. The *AND* operator is usually modeled using the triangular norm (T-norm) function, and the *OR* operator is modeled with the T-conorm function.

Mamdani [53] and TSK [76] are two of the most well-known types of fuzzy rule based systems. In the Mamdani model, the consequent of each rule is a fuzzy propositional statement, while the consequent of a TSK rule is a function of the input variables. Comparing these methods, Mamdani type system is more interpretable and provides a better mechanism for integration of the expert's knowledge into the system, while TSK system is more accurate and derives a set of more compact rules [16].

Implementation of a Mamdani fuzzy rule based system can be broken down into two phases. First, the basic fuzzy system characteristics (such as reasoning method, fuzzy operators, and defuzzification method) need to be selected. This phase is usually done by the system designer based on his experience on problem characteristics [61]. Secondly, the KB (Knowledge Base) of the system should be generated. The KB stores the knowledge that exists about the problem in the form of fuzzy rules. The KB consists of a DB (Data Base) and a RB (Rule base) [16]. DB stores the membership function parameters associated to each linguistic variable, while RB defines the specific composition of the antecedent and the consequent of each fuzzy rule.

The KB of a fuzzy system can be derived manually by a human expert or automatically using a learning method. Automatic learning methods are a good candidate for the problem of deriving the KB of a fuzzy system, because the search space of the problem is huge and there exists little or no a priori knowledge about the problem.

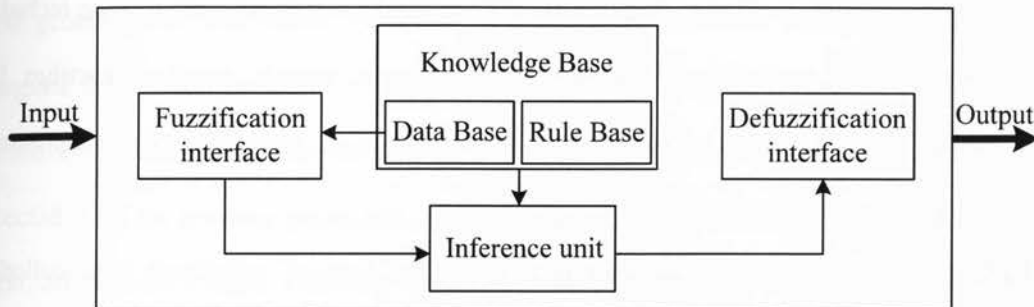


Figure 4. Fuzzy inference system

The fuzzy inference system is constituted of the following components: a fuzzification interface, an inference unit and a defuzzification interface [16]. The inference system uses the knowledge stored in the KB to make fuzzy inferences. This structure is shown in Figure 4. The fuzzification interface gets the crisp input from the universe of discourse and returns a fuzzy value which represents the membership degree of the crisp input in the fuzzy sets defined on the universe of discourse of that input. The fuzzy values derived by the fuzzification method are fed as an input into the inference unit. The inference unit deduces the fuzzy output from the fuzzy input value received from the fuzzification interface and the fuzzy relationships defined in the RB and DB of the KB. In the defuzzification interface, the fuzzy sets derived from the inference unit are aggregated into a crisp value which is the final output of the fuzzy system. For more information on the foundation of fuzzy systems, the reader is referred to [87].

2.4 Evolutionary Computation

Evolutionary Computation (EC) [29,39] refers to a set of population based optimization methods, in which the repeated application of genetic operators (inspired from biological evolution) on the candidate solutions finds the solution for the optimization problem. Although EA is the simplified model of its biological counterpart, it is a powerful search and optimization technique. There are different variations of evolutionary techniques namely genetic algorithm [39], evolutionary strategies [65,71], evolutionary programming [23], gene expression programming [22], and genetic programming [48].

All EC methods are based on the same principles. First a set of individuals (also called the population) is randomly generated. Each individual represents a potential solution for the user defined optimization problem. There are different ways to represent this population. The classical genetic algorithm uses a binary string for each member of the population. In the problems that the goal is to find an optimum value for the objective function, these binary strings encode a value which represents the objective value that the system is looking for. The way that each member of the population is encoded has an impact on the way that operators are applied on the individuals in the next steps of the algorithm. Therefore, the main factor that differentiates between various evolutionary algorithms is the way that each member represents the individuals of the population. For example, genetic algorithm uses binary strings, while genetic programming uses trees to encode the individuals.

Next, a number of individuals are selected based on a criterion known as *fitness function*. The fitness function is usually a scalar function which returns the quality of each individual (solution) in solving the goal optimization problem of the system. More precisely, the fitness function plays the role of environment in distinguishing between good and bad individuals. Individuals with

higher fitness values are selected from the previous generation and put into an intermediate parent population. This selection can either be stochastic, *i.e.* one individual can be selected more than one time based on a probability function, or deterministic.

The selected set of individuals is then used as parents for the production phase to produce the next generation of individuals. Here, the individuals are combined together in an attempt to integrate the advantageous part of each individual into the offspring (this operation is known as *crossover*). During the crossover process in genetic algorithm, two individuals are randomly selected from the previous generation, and a position is selected randomly inside the first string. Next, all the binary digits following the selected position are swapped between the two strings. The two entirely new generated individuals are moved to the next step. It also worth mentioning, that the crossover operation takes place with a probability known as the crossover probability. This probability determines the probability that a member of the intermediate parent population undergoes the crossover operation.

The new individuals might also undergo some random alterations (known as *mutation*). In genetic algorithm, an individual is randomly selected with a probability known as mutation probability, and a bit is randomly selected inside the individual. The randomly selected bit gets flipped from 1 to 0 or vice versa to generate a new offspring. Mutation was mainly developed for the problem of local minima. Mutation change individuals randomly so the population would not converge to a local minimum.

At the end, the new generated individuals are copied into the next generation. The next generation goes under the same process to produce succeeding generations. This process

continues until a maximum number of generations or an acceptable level of fitness for the best individual is reached [7]. This process is shown in Figure 5.

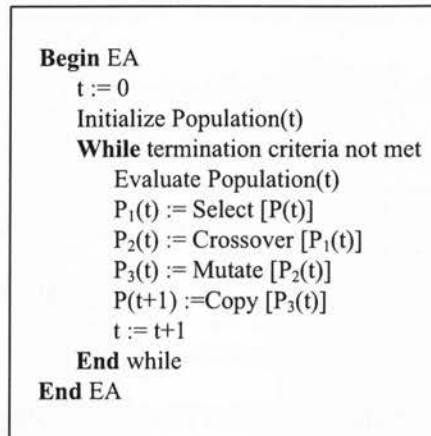


Figure 5. Structure of an evolutionary algorithm

EC is based on the fact that in each generation a number of individuals which have the above average fitness (*i.e.* those solutions which got better results on solving the defined optimization problem) are selected. These individuals are also combined to produce the next generation. In the next generation, individuals which inherited the goodness of their parents have higher chance of surviving to the next generations. Therefore, the evolutionary computation method has the ability to gather information about the initially unknown search space and guide the further generations to focus on promising sections of the search space.

To design an evolutionary computation method two parts need to be specified by the designer. First the genetic representation should be decided. The genetic representation defines the way that potential solutions of the problem are encoded. The representation should be able to encode possible solutions to the problem and at the same time be evolvable. Differences in the genetic representations are usually the main difference between evolutionary techniques. Linear binary

representation (used in GA) and tree-like representation (used in GP) are among the most common representations used.

Fitness function is the next main component of each evolutionary system. The fitness function is usually an expression which defines the quality of each individual in the population. In evolutionary computation the fitness function is used to rank the individuals against other individuals of the population.

2.5 Evolutionary fuzzy system

Determination of the system parameters is a crucial component of any fuzzy system design process. The basic structural parameters of the fuzzy system which deal with the inference mechanism are usually determined by an expert, *e.g.*, reasoning method and defuzzification mechanism. An automatic mechanism is more suitable for derivation of operational parameters, *e.g.*, Membership function values, for the following reasons: (a) System experts generally have less accurate assumption for the proper values of the operational parameters, and even if they have, the complexity of the problem increases exponentially with the number of attributes and fuzzy sets of each attribute. (b) Derivation of the operational parameters can easily be encoded as a search problem, where different operational parameters of the fuzzy system constitute the search space [61].

Evolutionary fuzzy methods are categorized under the concept of soft computing methods. Soft computing methods were designed to work in the environments with partial and uncertain information. These methods, in contrast to hard computing methods, are more flexible and need less computation power [16]. Neuro Fuzzy Systems, Genetic Neural Networks, and Genetic Bayesian Networks are some common examples of the hybrid soft computing methods.

Evolutionary methods are well-known as strong search methods which can do well in complex search environments, and fuzzy systems can provide flexible inference mechanisms in uncertain and imprecise environments. A possible combination of these two methods is possible through the concept of Evolutionary Fuzzy System (EFS). EFS is a fuzzy system which uses evolutionary methods to learn and tune different parameters of the fuzzy system. In the design of conventional fuzzy systems, the expert designer should design different components of the fuzzy system, but in the EFSs, evolutionary methods are used as search mechanism to design a part or the entire KB of the fuzzy system [16].

In order to use evolutionary methods as a search mechanism to find the KB of a fuzzy system, different components of the KB should be represented in a genotype space so that the genetic operators can be applied to them. As mentioned before, the KB of a fuzzy system is composed of a Rule Base (RB) and a Data Base (DB). The designer should decide which components of the KB are subject to the optimization process by the evolutionary method. Among different learning methods that have been proposed for KB derivation, the simplest ones use evolutionary computation to derive the RB from a predefined DB or vice versa. These methods are computationally efficient, but they ignore the mutual relationship that exists between components of the KB [17]. On the other hand, methods which derive the RB and the DB at the same time are computationally expensive, as these methods need to deal with a much bigger search space. Besides, a fitness function which defines the objective of the evolutionary process should be determined. This objective could be a KB which better solves the given problem when embedded into a fuzzy inference system.

3 Decision Support Systems for Design – An Overview

Design of an artifact can be seen as a process of specifying the structural specifications of the artifact. This specification should satisfy a number of design constraints and functional requirements. This process can be divided into the following steps [75]:

1. Problem identification: the problem and the available resources are defined.
2. Specification generation: the performance specifications of the goal artifact are identified.
3. Concept generation: some possible alternative solutions for the artifact are generated.
4. Analysis: the performance specifications of the generated alternatives are calculated.
5. Evaluation: the calculated performance are compared with the given performance specifications. If no solution satisfies the performance specifications go back to step 3.

The loop between steps 3 and 5 continues until a satisfactory design is generated.

Computers have initially been used as analysis tools in the design process. As computers have huge computational power, they were assumed to be more suited to do the burden of calculations in the analysis step. In this viewpoint, the designer does the actual design and uses the computer as a fast calculator to perform a diagnosis on the proposed design. Although this view provides a good balance between human and computer usage, it ignores the searching and pattern matching capabilities of the computers. Application of Knowledge Based Expert Systems (KBES) in the design process was one of the first attempts to use computers as an intelligent tool to help humans.

3.1 Application of knowledge based expert systems in the design of electromagnetic devices

Expert System (ES) is a branch of Artificial Intelligence (AI). ESs are defined as intelligent computer programs that use knowledge and inference procedures to solve problems which need significant human expertise for their solutions [21]. An expert system is composed of a knowledge base and an inference engine. The knowledge base stores the expert's knowledge in the problem domain which can be represented in a number of ways (such as rules and objects). The inference engine uses the knowledge stored in the knowledge base to make conclusions and give them as responses to the user's queries for expertise [27]. In general, a knowledge based system is suitable for applications which are too complex to be solved with mathematical formulations or too difficult to be solved using optimization methods [57].

KBESs have been applied for many engineering design problems. CONGEN [75] was one of the first applications of a KBES as a framework for conceptual design of buildings. The Knowledge base is divided into several layers, separating the domain knowledge from domain independent knowledge. The evaluator, analyzer, constraint manager, synthesizer and the geometric modeling constitutes the knowledge independent knowledge sources, while the objects, plans and constraints of a specific problem constitute the domain knowledge sources. Design of pneumatic systems [72], design of air-cylinders [12] and design of small computers [11] are some other applications of KBESs in engineering design tasks. Besides, KBESs have also been applied to the design of electrical machines.

Garrett et al. [26] were among the first to propose the application of knowledge based systems in the design of electrical machines. They proposed an object-oriented KBS (called Encore) for the

design of transformers and inductors. Encore automated all the steps for the design of an electrical machine. The system used “generate and test strategy” in which a rough design was generated first and then the heuristic knowledge was used to improve the initial design. All the mechanisms for generation, checking and pruning of the designs were implemented as rules. Objects in the knowledge base stored the properties of different components of the electrical machines used throughout the design process. Groschwitz et al. [30] proposed the application of an expert system in the design of transformers. The authors implemented a system which chose the optimum magnetic components from a given database. Criteria such as power rating, dc output voltage, temperature rating, importance of a low-profile core and isolation class were used during the design process.

Lowther [52] developed a hierarchical view as the structure of the knowledge base. He represented the structure of the electrical machine in a set of views, where the higher layers give more abstract views and the lower ones give more details about the design. He proposed Case Based Reasoning (CBR) [46] as the starting point for each design. CBR systems are well suited for problems that the underlying principles of the problem cannot be easily decomposed into linguistic rules or the involved principles are not well understood [49]. In this approach, the specifications of a new design are compared with a database of previously stored designs. The design which best matches the current design is used as rough design for the newly proposed design. Vo et al. [81] also studied the applications of a CBR approach for the design of electromagnetic devices. They proposed a system which had the ability to make structural changes to the model if the given specifications are “over-constrained”.

Kwong et al. [49] described a case based reasoning approach for concurrent product and process design of low power transformers. Under the current methodology, the design engineer first

designs the transformer and then the process engineer makes the process for the design. In the proposed approach, a CBR system is used to estimate the activities during the process design of low power transformers, *e.g.* estimation of assembly time and selection of winding machine, based on the previous cases stored in the case base of the CBR system. Therefore, the process engineer can work concurrently with the design engineer. The case base of the system is composed of cases stored using indexes, such as primary and secondary voltage, which are used in a similarity measure for later retrieval of the cases. The retrieved cases are adapted by changing the number of primary windings, and then the best one is selected based on the distance to the given solution.

Saldanha et al. [69] studied the application of an Algebraic Constraint System (ACS) as part of knowledge based system approach for design of electromagnetic devices. The ACS was developed to model and manipulate analytical or mathematical equations that are involved in the design process. The system has the capability to parse an equation into executable code, apply an equation, and solve the equation for each variable in an equation. Therefore, the designer can concentrate on generating an accurate model for the electromagnetic device as the code to solve the equations for each variable is generated automatically. The system works by building a network where each node represents an equation and the links between the nodes represents the variables. During the design process, as the user enters values for variables, the ACS checks the consistency of the network with the newly entered values. In the case of inconsistency the user is notified to modify the value of some design variables.

Rong et al. [67] described a CBR approach for the design of electromagnetic devices. The case-based engineering design system (CBEDS) that they proposed stores the structural knowledge of the devices in a semantic network. Besides, the mathematical knowledge is organized in an

equation networks using the concept of algebraic constraint systems. The system uses these knowledge types to relate a new set of specifications to a stored case in the case base, and do the modifications on the retrieved case.

Sassine et al. [70] investigated a knowledge-based system based on the concept of blackboard architecture to design electromagnetic devices. The proposed system uses the blackboard as a mean for coordination and communication between different design tools. Authors classified the knowledge needed to complete a design into three categories, namely, knowledge storage, design knowledge architecture, and control knowledge. Knowledge storage defines the facts describing the current designs and how these facts can represent the design space. Design knowledge architecture classifies the design knowledge as device and operational knowledge, where the device knowledge stores the physical properties of a device and the operational knowledge maintains the relational, behavioral and procedural knowledge of the design tools. Finally, the control knowledge identifies the sequence of actions needed to take to focus the search during design process.

Lowther [50] investigated the effects of different design parameter changes on the performance of final electromagnetic system. He studied four paradigms for this purpose namely, semantic networks, response surfaces, interval mathematics and sensitivity analysis. Semantic network can provide qualitative relationships between the design parameters and the performance criteria, while the response surface can provide the qualitative and quantitative measures between the input and the output parameters. Interval mathematics and sensitivity analysis are two methods to measure the effects of an error during the manufacturing process on the performance of the final device. These methods can provide feedbacks on the feasibility of the proposed design and can reduce the cost of faulty designs in this way.

3.2 Application of soft computing approaches in the design of electromagnetic devices

Knowledge based approaches need to have access to an expert in the domain of interest for the process of knowledge acquisition. The expert should have experience and expertise in the application domain and be available for the knowledge acquisition sessions with the knowledge engineer. The knowledge engineer interviews the domain expert to extract the heuristic knowledge of the expert, formalize the extracted knowledge, and represents this knowledge in the knowledge base of the expert system. There exist some problems in the knowledge acquisition problem a number of which includes: a) unavailability of human expert b) inaccuracy and inconsistency in the knowledge extracted from expert c) experts having lots of tacit knowledge which is hard to describe in a formal format d) the extracted knowledge is constant and cannot adapt itself to the changing environment. Due to the problems that exist in the elicitation of knowledge from human experts, knowledge acquisition from human experts was considered as a major problem in the development of knowledge based systems. These problems are usually referred to as knowledge acquisition bottlenecks [41].

To overcome the problems of knowledge elicitation from humans, some researchers proposed the concept of automatic knowledge acquisition [31]. Automated knowledge acquisition is based on machine learning methods. In this approach, an inductive learning algorithm is used to generate knowledge from the data. In other words, the system tries to convert the implicit knowledge that exists in the set of examples of data into an explicit intentional description.

Soft Computing (SC) methods are a group of artificial intelligence methods that have been applied to the design problem. Compared to conventional symbolic computing methods, SC

methods provide more synthesis capabilities, *e.g.* evolutionary methods can deduce conclusions that are not necessarily stored in the knowledge base, or they can deal with the less precise inputs, *e.g.* fuzzy systems can guide a less precise input to a more optimal solution [78]. Evolutionary computation [9], fuzzy system [20], and particle swarm optimization method [4] are some of the most well-known soft computing methods that have been applied to the problem of designing electrical machines.

Chiampi et al. [14] presented a fuzzy logic approach for the problem of multiobjective optimization. The design of electromagnetic devices usually categorizes under the concept of multiobjective optimization where several conflicting goals, *e.g.* increasing force and decreasing the joule loss, are competing with each other. Traditional approach for dealing with multiobjective problems is based on definition of a global performance function which tries to merge all the objectives in a single scalar function. Converting all the objectives into one scale and finding the appropriate weights for different objectives are some of the problems of this approach. The authors have proposed the use of fuzzy logic for this problem, where a membership function was defined for each objective function and let a scalar optimization algorithm to find the optimum results. The system was applied to optimize a DC electromagnet and Superconducting Magnetic Energy Storage (SMES) and the results were promising. Park et al. [59] also applied the concept of fuzzy decision making to the optimization of an induction motor design problem.

Arkadan et al. [5] presented a fuzzy logic environment to predict the performance characteristics of different electromagnetic devices for variant design parameters. The system had the capability to optimize the performance of permanent magnet generators by reducing the weight, minimizing total losses, and lowering total harmonic distortions. Francois et al. [25] introduced a

fuzzy expert system for designing electrical appearances. The proposed system permits imprecision and uncertainty to be included in the rules intended for the preliminary design stages.

Lowther [51] studied the application of Artificial Neural Networks (ANN) as an assistant for the numerical processes used in the electromagnetic device design. He investigated the application of ANN for the mesh generation process, and application of Self Organizing Feature Maps to predict the field structure.

Renyuan et al. [66] introduced a new intelligent simulated annealing algorithm for the optimal design of electromagnetic devices. The proposed algorithm combines fuzzy logic with simulated annealing algorithm in an attempt to identify infeasible solutions before objective function computation. In this approach, a number of initial simulated annealing points are randomly chosen to infer fuzzy relationship between the point's variable values and the feasibility of the point. This relationship is later applied to remove the need for objective function computation for the points that the fuzzy relationship system detects them as infeasible. The system was applied to the design of a BLDC motor and it reduced the computation time by 60 percent.

Guimaraes et al. [33] mentioned that in most cases the membership functions and the T-norm aggregation functions used as a part of fuzzy systems are not differentiable. This fact prevents the use of deterministic optimization methods for solving these problems. Authors proposed the use of differentiable membership functions and aggregation functions as a part of an optimization system that utilizes both deterministic and stochastic methods in a hybrid approach. The presented results showed that the hybrid methodology needs less computational power compared to pure stochastic methods.

Bianchi et al. [9] presented an approach for the design of brushless DC motors using genetic algorithm. Minimum cost was proposed as the objective function of the system, while other constraints on geometrical, electrical and thermal specifications of the design were defined as penalty functions. The penalty functions were introduced to ensure that unacceptable designs get eliminated after some generations. Authors showed that their system got better results compared to the climbing hill method. Also they showed that the quality of the climbing hill method is greatly affected by the quality of the initial design, while the goodness of the starting point does not affect the final result of the genetic algorithm method.

Rashid et al. [64] studied different methods to build a neurofuzzy system for designing electromagnetic devices when the number of variable parameters is high. Curse of dimensionality or model complexity is one of the major problems when dealing with models with a great number of parameters. Clustering the search space, reducing the number of input parameters by approximation, and partitioning the search space into even subsets are some of the approaches proposed to deal with the curse of dimensionality problem. These approaches were applied on a seven variable problem for designing a loud speaker magnet assembly unit, and the presented results showed the effectiveness of the presented approaches.

Guimaraes et al. [32] proposed an object-oriented structure which stores specifications of different computational intelligence techniques for optimization of electromagnetic devices. The goal of each computational intelligence method was to predict the output of the objective function during the design of the device, because the computational cost is usually high when a numerical method is used. Specifications of two artificial neural networks (multilayer perceptron, and radial basis function) and two neurofuzzy networks (Sugeno model and

Tsukamoto model) were stored in the proposed structure. The system had promising results when applied on the design of a superconducting magnetic energy storage problem.

Conventional knowledge based approaches are knowledge intensive and logically sound. On the other hand, evolutionary computation approaches have more similarities with the human invention process. Human invention process is not a deterministic process. In other words, a new design is noteworthy if it cannot be logically deduced from what was previously known. This process also exists in the evolutionary methods as a solution is not guided by mathematical logic [47]. One of the strong points of evolutionary computation methods is the existence of a population of contradictory and inconsistent solutions which does not exist in logically sound systems. This population provides the diversity of solutions which is one of the preconditions for the success of the evolutionary process.

Hui et al. [42] developed an Improved Genetic Algorithm (IGA) for the single and double objective design problems of power transformers. Authors have proposed a new encoding scheme, a stochastic crossover operator, an adaptive mutation operator and a new scaled fitness function compared to the conventional genetic algorithm. The system was applied to design a S9-1000/10 kV power transformer, considering the total material cost and operating cost as the objectives of the design process. Final results showed the improvement of the system over the conventional design method.

Preis et al. [62] compared three high-order deterministic optimization techniques (steepest-descent, conjugate gradient, quasi-Newton) with one stochastic optimization technique ($\mu + \lambda$) in designing electromagnetic devices and concluded that the stochastic method had competitive

speed of convergence, and superior generality and robustness compared to the deterministic methods.

Upadhyay et al. [80] developed an intelligent CAD system based on the concept of genetic algorithm to design brushless DC motors. The proposed system takes efficiency as the objective function while temperature-rise and weight are considered as constraint conditions. In this approach, airgap, airgap flux density, slot electric loading, magnet-fraction and the slot-fraction were considered as design variables. Authors showed that the intelligent system improved efficiency and phase-inductance compared to the conventional design method.

Mirzaeian et al. [56] proposed a new optimization mechanism based on a genetic-fuzzy system. The system was applied on the design of a Switched Reluctance Motor (SRM) with two objectives: high efficiency and low torque ripple. The proposed system uses a probability function to select the chromosomes with better performances for the next generation. The probability function combines objective functions with the fuzzy weights, where each fuzzy weight shows the goodness of a chromosome for a specific objective function. Authors applied the proposed system on a test case and showed that the optimization system improved the performance while decreasing the torque ripple, weight and size at the same time. Choi et al. [15] also applied a SC method to determine the optimal shape of a switched reluctance motor. Their method was based on fuzzy logic, where three membership functions for the torque profile of the SRM at variant rotor angles were defined. The final goal of the system is that the motor keeps the torque performance at higher speeds. Simulation results showed that although the optimal derived motor could not outperform the conventional design motor at lower speeds, but the optimal motor got better results at higher speeds and on average compared to the conventional motor.

We mentioned some of the advantages of the SC approaches over the knowledge based approaches in the design problems. On the other hand, conventional knowledge based approaches have some advantages over SC methods, *e.g.* genetic algorithm (GA), some of which are as follows [2]:

- Integration of expert's knowledge: GA is a stochastic and blind search method that does not accept any direction during the search process. However, there exists a large amount of a prior directional heuristic knowledge that can be integrated into the system. Accommodation of this knowledge in the KBES can significantly reduce the computational cost.
- Search space: Depending on the initial assumptions, the motor design process can include numerous input and output parameters increasing the size of the GA search space, hence increasing the computational cost and time, as well as increasing the possibility of getting trapped in local minima. Whereas a KBES uses a systematic approach and is directional where the design task is achieved by iterating among the sub-groups in a partitioned rule-based.
- Transparency: The computation method of the KBES systems is more understandable to the user; this will improve the reliability of the final results. Moreover, the transparency of the underlying system will make it easier to modify the system.

Having said this, it seems that there should be a tradeoff which covers the advantages of both the soft computing and the conventional knowledge based approaches. The conventional knowledge based expert system is not a good solution because of the knowledge acquisition bottleneck problem and the problems with adaption of the system with the changing requirements. On the other hand, a full automatic system is not a good solution because of the problems such as transparency and integration of expert's knowledge. We think that a semi-automatic system

which uses both the searching potentials of the computers and the knowledge of the expert in problem domain is the best way to use the advantages of both systems. In this approach, the automatic data mining system is used to extract knowledge from the set of examples and then propose the derived knowledge to the expert. The expert then combines this knowledge with his own expertise and makes the final decision.

4 Methodology

4.1 IDSS structure

The proposed intelligent decision support system (IDSS) includes an evolutionary fuzzy system (EFS) and a knowledge based expert system (KBES). The overall structure of the system and a possible interaction session of the user and the system are shown in figure 6. The design process starts when the user gives the set of requirements as input to the IDSS. Then EFS derives a set of modification rules which could be later used to improve the performance of an unsatisfactory design. The set of derived rules are presented to the user. The user might accept, reject, or modify the rules and embed the approved rules into the knowledge base of the KBES. Next, the iterative process starts in the KBES and at the end s set of developed designs are given as outputs of the system to the user.

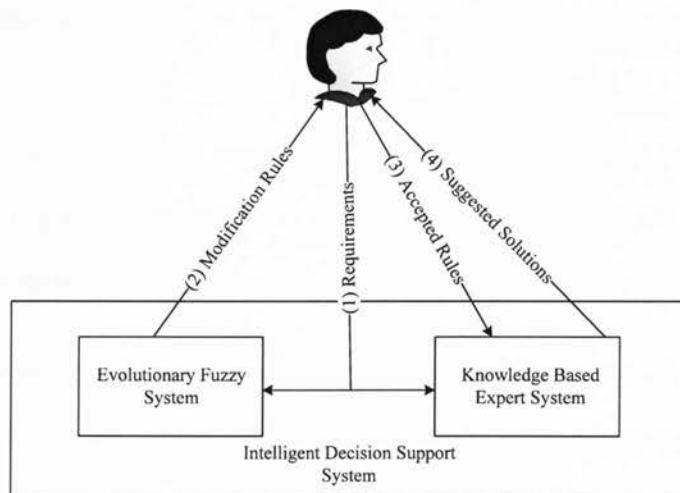


Figure 6. User interaction with the IDSS

4.2 KBS architecture

The iterative process in the KBES of the IDSS starts when an initial design is generated using the conventional design method. The performance of the developed design is compared to the given set of requirements in the test section. If the developed design satisfies the requirements the design process is stopped. Otherwise, the process goes to the modify section where a number of modifications are suggested to improve the performance of the unsatisfactory design. The iteration between the test and modify sections continue until no more modifications could be suggested or the proposed improvements does not improve the performance of a design. This process is shown in figure 7. In this figure, the dashed boxes represent different types of knowledge which exist during the design process.

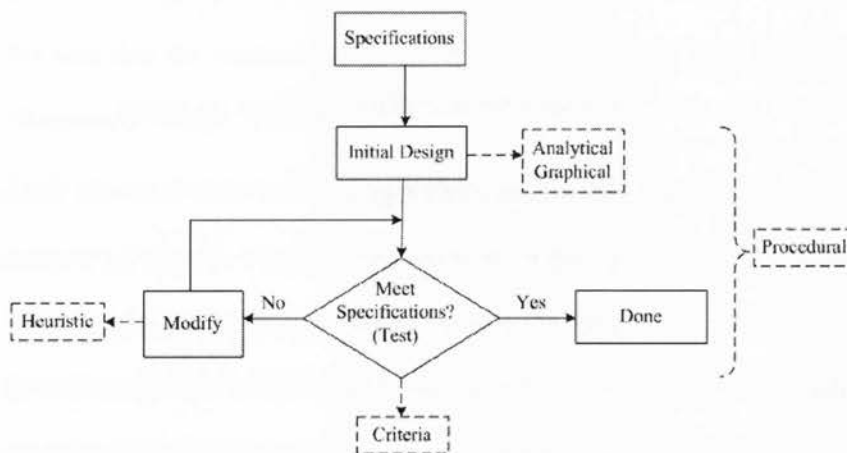


Figure 7. Design steps

Different types of knowledge used during the design process are stored in the knowledge base of the KBES using three paradigms of frames, rules, and procedural attachments. We will show that the iterative motor design process can be well implemented in a KBS framework. The overall

structure of our knowledge base is illustrated in Figure 8. Each part has a specific responsibility which will be explained next.

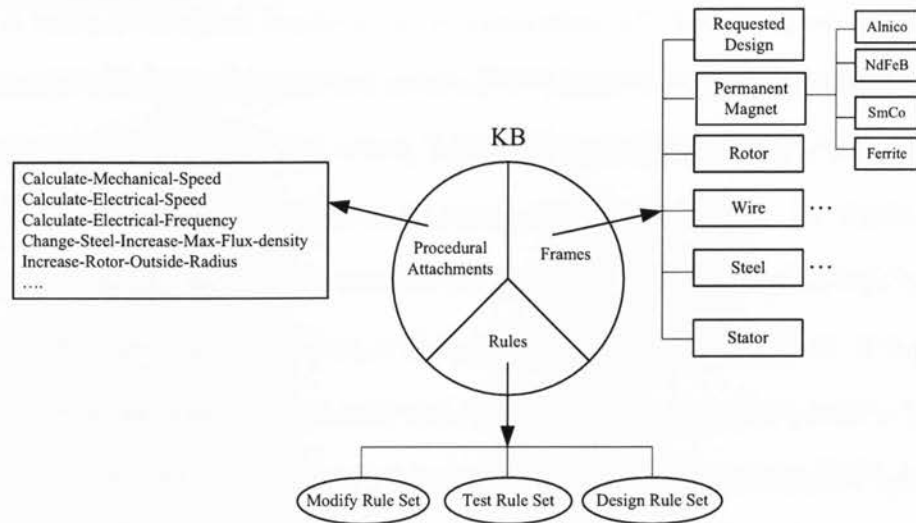


Figure 8. Knowledge Base (KB) structure of the system

- **Frames:** have a hierarchical structure and together with their slots and facets are used to represent the BLDC motors and to store the specifications of different elements involved in the motor design process and to define relationships. As can be seen in Figure 8, the motor parts along with the materials used for motor construction are represented as frames. The materials involved in the process have a hierarchical structure (e.g. Alnico is an instance of the PM) that can be well implemented with the inheritance present in frame structure. Requested design specifications are held in the requested design frame. This frame contains the input parameters and the requested performance of the final designs.
- **Rules:** have an “IF condition THEN action” structure and can be used to represent the design algorithm. Rules provide a mechanism to access, to control and to change data structures (i.e.

frames, their attributes and relations) toward the goal of finding a set of satisfactory designs. The conventional design algorithm explained in section 2.1 is also implemented by means of rule. This is done by partitioning the rule base into three main sets, *i.e.*, initial design, test and modifications. The BLDC design process starts with the activation of the initial design rule set and then proceeds with successive iterations between the test and modification rule sets until no more modifications can be made. The functionality of these rule sets are as follows:

- Initial design rule set: uses the conventional motor design methodology to reach an initial design of the motor [36]. The set of equations used for motor design are given in Appendix A. A sample design rule is shown in Table 2. When the conditions are met, *i.e.*, the required design parameters are known, the rule is executed by calling a procedural attachment which calculates the value of back-iron-width of the motor. We note that the required design parameters are collected from the “Motor” and the “Requested-Design” frames.
- Test rule set: compares the specifications of the developed designs with that of the requested design to indicate if they meet the desired performance that are given in the requested design frame. If the design criteria are met then no action is to be taken and the exiting design is put in a satisfactory designed queue. Otherwise, appropriate flags are set to indicate which specification is under-performed.
- Modify rule set: proposes modifications to improve the performance of flagged designs based on the available expertise, *i.e.*, modification rules. These rules are derived using the evolutionary fuzzy system. The modified designs are then considered for evaluation by test rule set.

- Procedural attachments: represent the analytical expressions that the system uses for the design process. Frames and rules lack the ability to perform numerical calculations, so procedural attachments compensate for the lack of computational ability inherent to frames and rules. Procedures are attached to their related frames and are activated by rules.

Table 2. A sample design rule

DESIGN RULE TO CALCULATE BACK IRON WIDTH	DESCRIPTION
(defrule Design::27-Calculate-Back-Iron-Width (object (is-a REQUESTED-DESIGN) (maximum-steel-flux-density ?steel-flux) (steel-lamination-stacking-factor ?lam-factor) (motor-axial-length ?axial-len))	<i>Get the values of</i> <i>Maximum-steel-flux-density B_{\max}</i> <i>Steel-lamination-stacking-factor K_{st}</i> <i>Motor axial length L</i>
?motor <- (object (is-a MOTOR) (Air-Gap-Flux ?air-flux)) =>	<i>From the Requested Design, and the value of</i> <i>Air-gap-flux ϕ_g</i> <i>From the motor that the design request has been initiated.</i>
(send ?motor Design::Calculate- Back-Iron-Width ?axial-len ?air-flux ?steel-flux ?lam-factor))	<i>Then</i> <i>Send a message with all the collected values to the "Calculate-Back-Iron-Width" procedure.</i>
(defmessage-handler MOTOR Calculate- Back-Iron-Width (?axial-len ?air-flux ?steel-flux ?lam-factor) (bind ?self:back-iron-width (/ ?air-flux (* 2.0 ?axial-len ?steel-flux ?lam-factor))))	<i>Procedural attachment to calculate the back iron width:</i> <i>Calculate the back iron width using the given values</i> <i>and the following equation [6]: $w_{bi} = \frac{\phi_g}{2B_{\max}k_{st}L}$</i>

4.3 Evolutionary fuzzy system

4.3.1 Fuzzy IF-THEN rules

Rules are one of the common means to represent the mapping function of a predictive system. Rules have the form IF (antecedent)-THEN (consequent). The antecedent of a rule includes a logical expression among the predicting attributes, and the consequent assigns one of the possible values to the goal attribute. If the predicting attribute values of a record in the dataset satisfy the antecedent of a rule, then the class in the consequent is assigned to that example.

There has always been a tradeoff between accuracy and interoperability of a predictive system. Fuzzy logic is a good candidate to be integrated into a rule based system, as the resulting system's rules are more comprehensible for humans. Moreover, the system can better manage the uncertainty of the underlying dataset.

To better understand the concept of fuzzy rule based systems, consider a dataset S (shown in Figure9) with M records, where each record $Record_i$ has m predicting attributes and one goal attribute G .

	Predicting Attributes						Goal Attribute
	P_1	P_2	P_j			P_m	
Record ₁	x_{11}	x_{12}	$\cdot \cdot \cdot$	x_{1j}	$\cdot \cdot \cdot$	x_{1m}	x_{1G}
Record ₂	x_{21}	x_{22}	$\cdot \cdot \cdot$	x_{2j}	$\cdot \cdot \cdot$	x_{2m}	x_{2G}
Record _i	\cdot	\cdot	\cdot	\cdot	\cdot	\cdot	\cdot
	\cdot	\cdot	$\cdot \cdot \cdot$	\cdot	$\cdot \cdot \cdot$	\cdot	\cdot
	\cdot	\cdot	\cdot	\cdot	\cdot	\cdot	\cdot
Record _i	x_{i1}	x_{i2}	$\cdot \cdot \cdot$	x_{ij}	$\cdot \cdot \cdot$	x_{im}	x_{iG}
Record _M	\cdot	\cdot	\cdot	\cdot	\cdot	\cdot	\cdot
	\cdot	\cdot	$\cdot \cdot \cdot$	\cdot	$\cdot \cdot \cdot$	\cdot	\cdot
	\cdot	\cdot	\cdot	\cdot	\cdot	\cdot	\cdot
Record _M	x_{M1}	x_{M2}	$\cdot \cdot \cdot$	x_{Mj}	$\cdot \cdot \cdot$	x_{Mm}	x_{MG}

Figure 9. Dataset S

Let us assume that each predicting attribute P_j has been divided into K fuzzy sets $\{F_{j1}, F_{j2}, \dots, F_{jK}\}$, and each fuzzy set is defined through the μ_{jk} membership function. More precisely, $\mu_{jk}(x_{ij})$ represents the degree which $Record_i$ belongs to fuzzy set F_{jk} . In this notation, a fuzzy rule has the form:

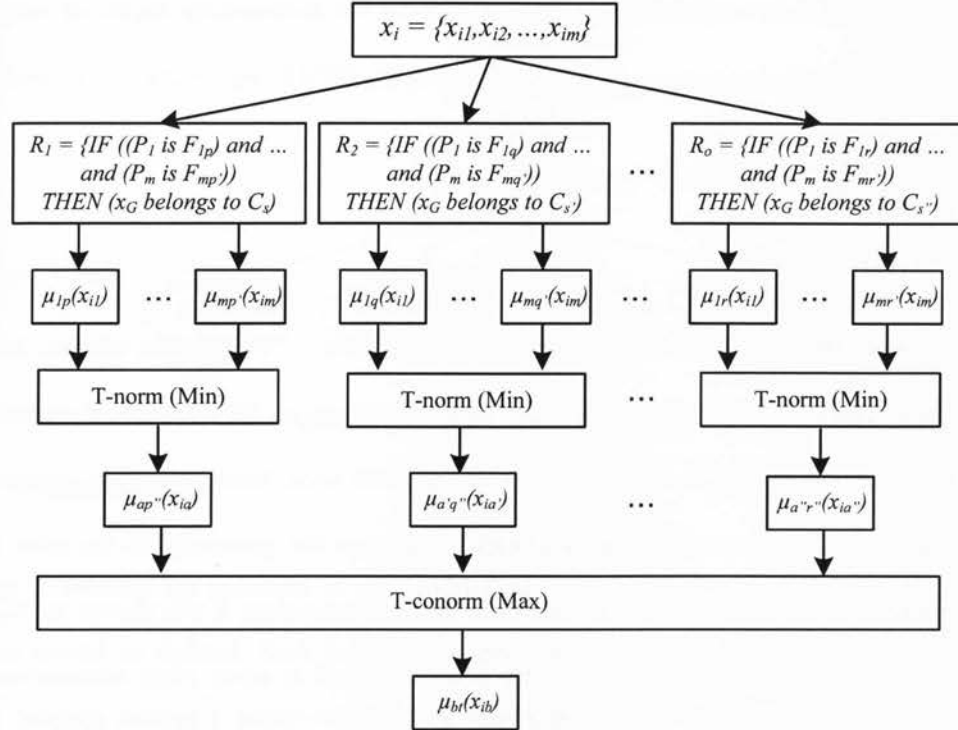
IF $((P_1 \text{ is } F_{1p}) \text{ and } (P_j \text{ is } F_{jp}) \text{ and } \dots \text{ and } (P_m \text{ is } F_{mp}))$

THEN $(x_G \text{ belongs to } C_s)$

$s = 1, 2, \dots, n ; p = 1, 2, \dots, K$

The fuzzy classification rule based system is composed of a Knowledge Base (KB) and a fuzzy inference engine [16]. The Rule Base (RB) and the Data Base (DB) constitute the KB. As their name suggest, the RB contains all the linguistic rules and the DB keeps the definition of the linguistic variables and the membership functions corresponding to each linguistic value in the fuzzy system.

The fuzzy reasoning method is the mechanism which uses the information presented in the KB to assign the class label to an unclassified input record. In our system, we used the fuzzy reasoning method proposed in [43] to calculate the output of the fuzzy rule based system. Assume that the input record $x_i = \{x_{i1}, x_{i2}, \dots, x_{im}\}$ and a Rule Base $R = \{R_1, R_2, \dots, R_o\}$ are given. The input vector's goal attribute x_{iG} is classified as the output of the single winning fuzzy rule from the rule base which has the highest compatibility degree with the input record. This process is shown in Figure 10. In this figure, the record belongs to the class which is indicated in the consequent of the winning rule.



$$p, q, r, t = 1, 2, \dots, K; a, b = 1, 2, \dots, m; s = 1, 2, \dots, n$$

Figure 10. The fuzzy reasoning method

We use an evolutionary system to derive operational parameters of the fuzzy system. More precisely, an evolutionary system is used to induce the antecedent and consequent of each fuzzy rule. In order to derive fuzzy rules, each fuzzy rule should be encoded as an individual of the evolutionary optimization method.

4.3.2 Rule presentation

We followed the Pittsburgh approach for rule representation [74]. In this approach the function set contains the fuzzy logic operators $\{AND, NOT\}$. We used the standard fuzzy operators for

{*AND*,*NOT*}[84]. More precisely, if $\mu_A(x)$ represents the membership degree of element x in fuzzy set A , then the fuzzy operators are defined as follows:

$$\mu_A(x) \text{ AND } \mu_B(x) = \min(\mu_A(x), \mu_B(x)) \quad (3)$$

$$\text{NOT } \mu_A(x) = 1 - \mu_A(x) \quad (4)$$

In our system the antecedent of the fuzzy rules are in form of conjunction clauses. In order to show fuzzy rules as individuals, we propose some restrictions on the structure of the individuals. These restrictions are: (a) The root of each tree is an *AND* node. Each *AND* node can have one or more than one *Terminal* node as its child(ren). Although the generality of the rules could be increased by integration of other connectives in the rule antecedent, it was shown in [82] that the rules with only conjunctive connectives are generic enough to cover other connectives. (b) The second level of the tree includes *Terminal* nodes. Terminals of the system are defined using the following rules:

$$\langle \text{terminal} \rangle ::= \text{NOT } \langle \text{literal} \rangle \quad (5)$$

$$\langle \text{terminal} \rangle ::= \langle \text{literal} \rangle \quad (6)$$

$$\langle \text{literal} \rangle ::= \langle \text{op} \rangle = \langle \text{linguistic value} \rangle \quad (7)$$

$$\langle \text{linguistic value} \rangle ::= \{ \text{Low}, \text{Med}, \text{High} \} \quad (8)$$

$$\langle \text{op} \rangle ::= \{ \text{Att}_1 \dots \text{Att}_N \} \quad (9)$$

(c) No more than one child of an *AND* node can refer to a terminal with the same attribute as the first operand and the = as the operator. For example, two terminals of *Temp=Low* and *Temp=High* cannot be the children of the same *AND* node [55]. The other node which should be included in the structure of an individual is the goal node which represents the consequent of each rule.

A sample rule of the system is shown in Figure 11. This rule is equal to: **IF** $((Att_3=Low) \text{ AND } (Att_2=Med) \text{ AND } (Att_4=High))$ **THEN** $(Goal=Class)$

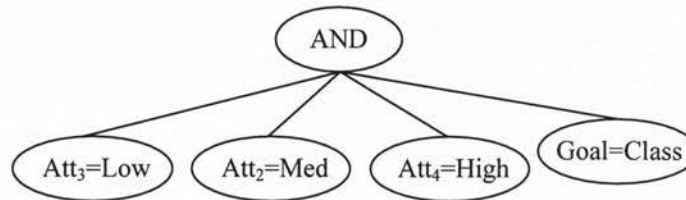


Figure 11. A sample rule of the system

4.3.3 Fitness function

In order to evaluate the goodness of each individual for solving the objective function, a fitness function should be defined. Each individual is given as an input to the fitness function and the fitness function returns a scalar value which shows the quality of that individual. Individuals with higher fitness value are better suited for solving the problem.

In our system, each individual represents a modification rule, so the fitness function should evaluate the quality of different modification rules. Roughly speaking, a modification rule is good if it can improve the performance of a developed design when the rule is applied to the case. There are two criteria which could be considered when evaluating the quality of a modification rule. 1) The amount of improvement. 2) The frequency of improvement.

To make the case more clear, consider the following example. Assume two modification rules which were designed to improve the efficiency of a motor. To compare the quality of these two rules, we apply each one on all the cases of an available casebase of previous designs. Then we measure the amount that each one of the rules has improved or degraded the quality of each case in the casebase of previous designs. These values could be added up to constitute a scalar value

which represents the quality of each modification rule in improving the efficiency of a BLDC motor. So the fitness function can be defined as:

$$fitness(r_j) = \sum_{i=1}^m A(r_j, c_i) \times F(r_j, c_i) \quad (10)$$

Where m is the number of cases in the casebase, r_j represents a modification rule of the system which is going to be evaluated, c_i is a case in the casebase, F is a scalar function which returns the results of application of the rule (r_j) on the case (c_i), *i.e.*, the difference in performance of the motor after the modification compared to the performance before the modification, and A returns the degree that the antecedent of c_i is satisfied by r_j .

It also worth to mention that, when a modification rule produces a result which is inconsistent, *e.g.* the slot current density is higher than the allowed value or the rotor inside radius is smaller than a minimum value, then a predefined constant value is removed from the fitness value of the modification rule as a penalty. The major steps of the system are shown in Figure 12. In this figure, the main loop of the system stops when a maximum number of generations is reached.

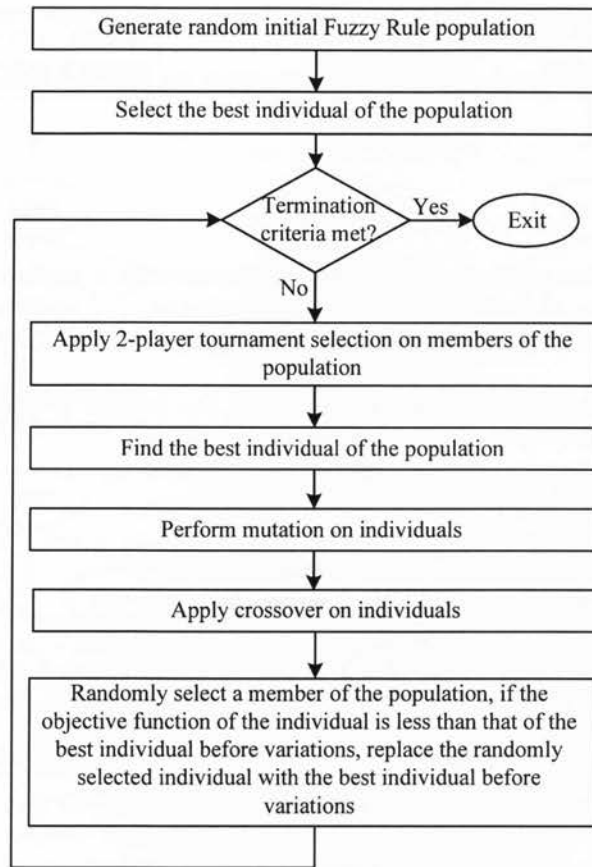


Figure 12. Steps of the evolutionary rule derivation system

4.3.4 Selection

We used 2-player tournament selection [7] in which the higher objective function value wins and is chosen for reproduction. The tournaments are played as follows: For each member of the population, one tournament is held. The second player for each tournament is randomly chosen from the population (sampled independently with replacement, *i.e.* being chosen as a second player in one tournament does not alter the probability of being chosen as a second player in

another tournament, and an individual may be chosen to play with itself). The winners of the tournaments are collected into a “parent pool” to which the genetic operators are applied.

4.3.5 Mutation

Each individual in the population goes under mutation with a probability known as mutation probability. In this process, a *Terminal* node is randomly selected in the individual and its value is swapped with a randomly selected terminal value. The process of mutation is shown in Figure 13.

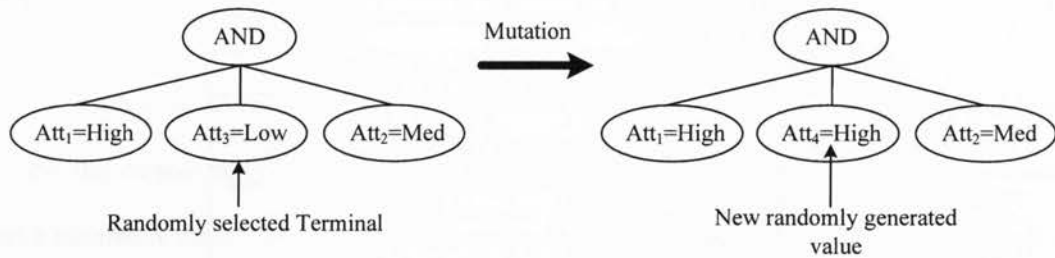


Figure 13. Mutation

4.3.6 Crossover

We used a modified version of cut and splice method [38] for crossover. The method works as follows: Two *Terminal* nodes are randomly selected in parent one and parent two. Next, all the *Terminal* nodes beyond the two selected *Terminal* nodes are swapped between the rules in two parents. After crossover, the child which has higher fitness value is moved to the next generation. This process is shown in Figure 14.

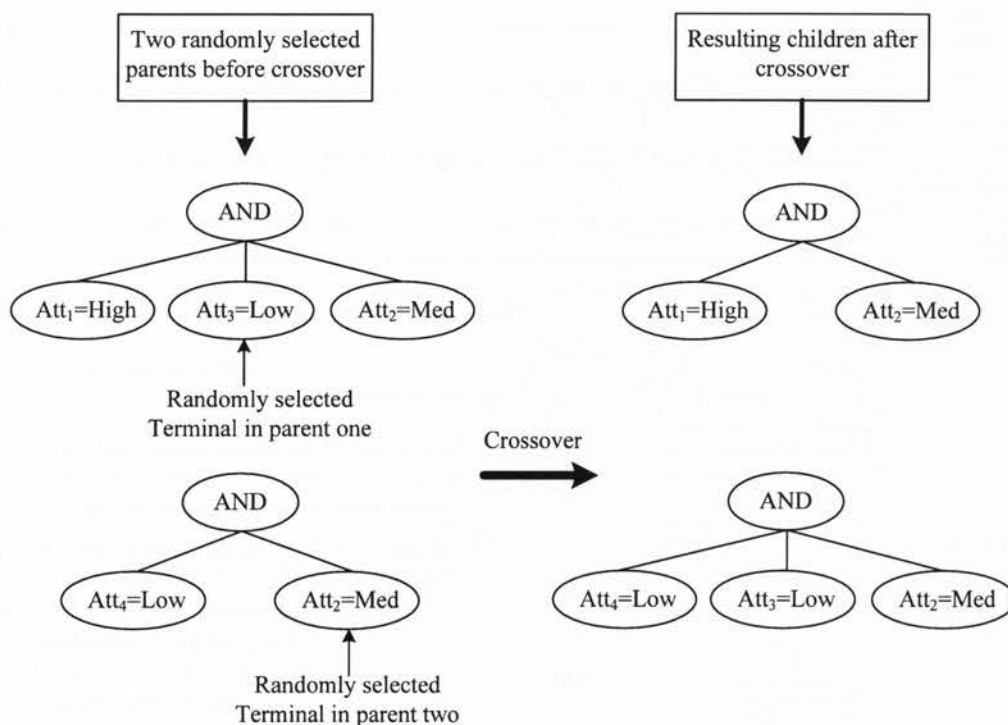


Figure 14. Crossover

4.3.7 Membership function

Each terminal variable can be assigned to a linguistic value which is defined through the membership functions. Note that each one of these attributes can be assigned to one of the linguistic values of $\{Low, Med, High\}$. This process is shown in Figure 15.

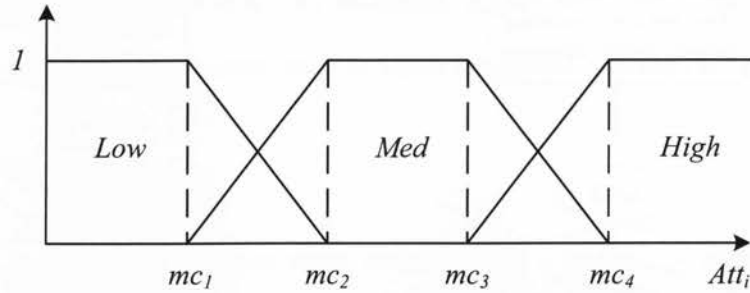


Figure 15. Membership functions of predicting attributes

The values for mc_1 , mc_2 , mc_3 and mc_4 are calculated to be of equal size from each other. In other words, the value for the range of each attribute is divided to five equal values and each one is assigned to one of the variables for the membership function of a terminal variable. Note that we could have also used an automatic method for derivation of membership function variables such as the one proposed in [1]. In this approach, a search algorithm is used to adapt the membership function parameters associated with each linguistic variable. Although approaches based on search algorithms consider the interdependence that exists between the rule base and data base of a fuzzy system, these approaches are computationally much more expensive because they have to deal with a much bigger search space.

4.3.8 Design algorithm

To design a BLDC motor, first the user gives the set of input parameters and the requirements to the IDSS. Then, the EFS derives a set of modification. These rules will be later used to modify the unsatisfactory designs. Next, the set of derived rules are given to the expert user. The user can accept, reject, or modify the given rules and embed the accepted ones into the knowledge base of the knowledge based expert system. Next, the iterative approach starts where an initial

design is first obtained using the conventional design methodology. Using a successive test/modification process, the design requirements are measured, and modifications, if any, are recommended accordingly. Iterations between modification and test will continue until no more improvements can be made [26,68]. This process can be broken into the following steps:

1. The given specifications are entered into the evolutionary fuzzy system to derive a set of modification rules.
2. The user embeds the set of accepted modification rules into the KBES.
3. An initial design is developed using the conventional design method.
4. The characteristics of the developed design are compared with the desired design, and the unsatisfactory aspects are marked to be modified later.
5. Modifications are suggested based on the deficiencies found in the existing design.
6. New designs are developed based on the suggested modifications.
7. The process will end if no more improvements can be done otherwise step 3 and forward will be repeated.

5 Results and Discussion

5.1 Modification rule derivation subsystem

The specifications of a case study for design of a BLDC motor is described in section 5.1.1. As mentioned in section 4.2.1, fuzzy rules can be extracted from a predefined dataset. Section 5.1.2 explains the casebase (or dataset) which was used to derive fuzzy modification rules. The structure of the modification rules is explained in section 5.1.3 and the modification rules derived based on the case study are represented.

5.1.1 Motor specifications

Among different configurations available for radial flux BLDC motors, the inner rotor and outer stator configuration was selected here. There are a variety of possibilities for the rotor and the stator structure. The traditional radial arc surface mounted magnet is chosen for rotor structure. The slotted stator is chosen for the stator structure because it has better heat conduction and higher permanence coefficient compared to the slotless configuration [35]. This configuration is depicted in Figure 1.

During the design process, the values for the design parameters should be determined by the design methodology. In general, the designer should fix the values for some of the variable parameters at the beginning of the design process and determine the value for the rest of the parameters during the design process. Most of the times, the designer has some ideas about the total size of the motor, the available input voltage and current, and the desired power output at some rated speed [36]. Based on these assumptions the design specifications of the motor are given in Table. 3. Note that most of the parameters which deal with geometrical and material

specifications are initial assumptions and their values would change as the design algorithm proceeds. Among geometrical parameters, outside stator radius and motor axial length specify the overall size of the motor so their values would be fixed during the process. Maximum back emf specifies the available input voltage for the motor so it will be constant. Torque and speed are the required specifications given by the user at the beginning of the design and would not change during the process.

Among the specifications, the number of phases is set to be three. Three-phase motors have good utilization of copper, iron, magnet, insulating material and silicon, when measured by the amount of these materials needed for a given output power. Although, motors with higher number of phases have better utilization, this utilization is canceled by the increase in number of transistors and leads needed for the control module, which increases the cost and reduces reliability. Three-phase motors have very good starting characteristics and low torque ripple [37]. Therefore, we fixed the number of phases to three for the whole process.

Table 3. Design specifications

Parameter	Value
Proposed steel	24-gauge silicon iron electrical steel
Proposed magnet	Sintered Alnico-8B
Torque (N.m)	1.994
Speed (rpm)	1000.0
Maximum back emf (V)	30.0
Number of slots	24.0
Number of phases	3.0
Maximum slot current density (A/m^2)	9000000.0
Number of magnet poles	8.0
Air gap length (m)	0.0026
Magnet length (m)	0.0025
Outside stator radius (m)	0.0635

Motor axial length (m)	0.0635
Steel lamination stacking factor	0.85
Conductor packing factor	0.59
Magnet fraction	0.9
Shoe depth fraction	0.1
Required motor constant	0.5
Required efficiency	90%
Required weight (Kg)	2.5

5.1.2 Casebase

The casebase used for derivation of modification rules is generated using the parameters given by the user. The set of variable parameters are selected from the set of input parameters mentioned in the previous section. The program takes the values that the user has entered for the variable parameters and changes each one through a predefined range to make the casebase. In other words, to make a new case for the casebase, the value for only one input parameter is changed within the range specified in the range section. Next, the performance characteristics of the new case are calculated. At the end, the input and the performance parameters are added as a new case to the casebase. The range of each variable parameter used for casebase generation is mentioned in the Table 4. For all the parameters, except the ratio parameter, the value of the input parameter is used for case generation. In the case of ratio, three values of {0.4, 0.5, 0.6} will always be used for casebase generation.

Note that the range specified for each parameter is chosen by the designer. For example, consider the “proposed steel type” parameter. The mentioned range for this parameter, selects two other steels types, where one has higher and another has lower steel gauges compared to the one specified by the designer. The designer can apply other types of variations on the proposed steel type, for example, designer can choose different steel material as the variation domain.

The choice of the range for each parameter depends on the range of materials available or manufacturing processes feasible. For example, the choice for the number of slots or the number of magnets range depends on the machinery available for the manufacturing of the stator and rotor. In other words, the search space is build considering the designs possible with the available machinery and material.

Table 4. Parameter range for casebase generation

Parameter	Range
Proposed steel type	Two other steel types, one with a higher and one with a lower gauge
Magnet length	Input value ± 0.003 (mm)
Magnet fraction	Input value ± 0.025
Proposed magnet type	Two other magnet types, one with a higher and one with a lower magnet remanence
Number of slots	Use Table 5 to select two number of slots, where one is bigger and one is smaller than the proposed value
Number of magnets	Use Table 5 to select two number of magnets, where one is bigger and one is smaller than the proposed value
Rotor outside radius/stator outside radius (Ratio)	Use three values of {0.4,0.5,0.6}

Table 5 shows the possible slot/pole combinations for three phase motors that can be used effectively. The table has listed all possible combinations for slot numbers from 3 to 48.

Table 5. slot/pole combinations for three phase motors [71]

Slots	3	6	9	12	15	18	21	24	27	30	33	36	39	42	45	48
Poles	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
			6	8	10	6	8	8	6	8	8	6	8	8	6	8
			8	10		8	14	10	8	10	10	8	10	10	8	10
			12	18		12	16	16	10	20	14	10	14	14	10	14
						14		20	12	22	20	12	16	16	12	16
						16			18	26	22	14	26	26	14	20
									20		26	16	28	28	16	32
									22		28	22	32	32	20	34
									24			24	34	34	28	38
												26			30	40
												28			32	
												30			34	
												32			38	
															40	

Each case in the casebase will have the mentioned seven attributes as the input attribute. Among these attributes, “proposed steel type” and “proposed magnet type” are nominal attributes, and the rest are numerical attributes. There are also three output parameters representing the weight, efficiency, and motor constant of the case.

With the given range for different attributes there should be 2187 cases in the casebase. But after the case production phase, the number of cases is usually slightly less than the specified amount. This reduction in number happens because of inconsistent cases. More specifically, some combinations of the input parameters are inconsistent as they violate the mechanical or electrical constraints that exist on a BLDC motor. We remove those cases which are not consistent with the constraints from the casebase. We implemented the following constraints in our system:

1. Rotor inside radius > 3 (mm)

2. Conductor slot depth > 3 (mm)
3. Peak conductor current density < maximum conductor current density

5.1.3 Modification rules structure

As we mentioned in section 4.2.2, each individual of the evolutionary population encodes the antecedent and consequent of a fuzzy modification rule. It seems that there should be another node included in the antecedent of our fuzzy modification rules, to represent the unsatisfactory aspect of the design. This node was not included as a part of the individual, because we used binary classification [45] in our system. In this approach, during each run of the system, all the rules of the system are trying to find a solution for only one of the problems of the BLDC motor. Therefore, the rule derivation system should run three times, because we have three problems of weight, efficiency, and motor constant that we are trying to find a solution for. Although binary classification increases the running time of the system, it prevents mating to happen between individuals that are trying to solve different problems. This approach also simplifies the structure of each individual [55]. We use binary classification in our system therefore there is no need to encode the problem node in the antecedent of our rules.

We ran our system to derive rules for three problems that a BLDC motor might have. These problems are:

1. Weight: This shows that the given motor has higher weight than the requested weight given in the specifications. The weight is calculated through the following formulas:

$$W_{ste} = [\pi(R_{so}^2 - R_{si}^2 + (R_{ri} + W_{bi})^2 - R_{ri}^2) - N_s A_s] L K_{st} \rho_{ste} \quad (11)$$

$$W_{mag} = [\pi(R_{ro}^2 - (R_{ri} + W_{bi})^2)]L\rho_{mag} \quad (12)$$

$$W_{wir} = n_s N_s L \rho_{wir} \quad (13)$$

$$Weight = W_{ste} + W_{mag} + W_{wir} \quad (14)$$

Where, W_{ste} = Weight of steel, R_{so} = Outside stator radius, R_{si} = Inside stator radius, R_{ri} = Inside rotor radius, W_{bi} = Back iron width, N_s = Number of slots, A_s = Conductor area, L = Motor axial length, K_{st} = Lamination stacking factor, ρ_{ste} = Steel mass density, W_{mag} = Weight of magnet, R_{ro} = Outside rotor radius, ρ_{mag} = Magnet mass density, W_{wir} = Weight of wire, n_s = Number of turns per slot, ρ_{wir} = Weight of wire per meter, $Weight$ = Total weight of the motor. We did not include the weight of the end turn wires in the total weight of the motor, because the length of the end turn wires depend on the wiring method used.

2. Efficiency: This shows that the efficiency of the motor is less than the requested efficiency given in specifications. We used equation (1) to calculate the efficiency of the motor.
3. Motor constant: This shows that the motor constant of the motor is less than the one given in specifications. We used the following formula to calculate the motor constant of a motor [35]:

$$K_m = B_g R_{ro} \sqrt{K_{wb} L N_m A_s / \rho} \quad (15)$$

Where B_g = the air gap flux density, K_{wb} = bare wire slot fill, N_m = number of magnet poles, A_s = Conductor area, ρ = resistivity of the wire.

5.1.3.1 Antecedent of fuzzy modification rules

The antecedent of each fuzzy rule is consisted of the conjunction of fuzzy propositions. Each fuzzy proposition is consisted of the assignment of a linguistic value to a predicting attribute. The parameters listed in Table 6 were used as the predicting attributes that the system can use in the antecedent of the modification rules. The fuzzy propositions are encoded as terminals in the rule structure of EFS using the method mentioned in section 4.2.2.

Table 6. Predicting attributes used in the antecedent of modification rules

Parameter	
Proposed steel type	Magnet length
Number of slots	Proposed magnet type
Magnet fraction	Number of magnets
Rotor outside radius/stator outside radius (Ratio)	

Each attribute in the antecedent of the rule can be assigned to one linguistic value to make a fuzzy proposition. Three linguistic values of $\{Low, Med, High\}$ are defined through the membership functions for each attribute. These membership functions are shown in Figure 16.

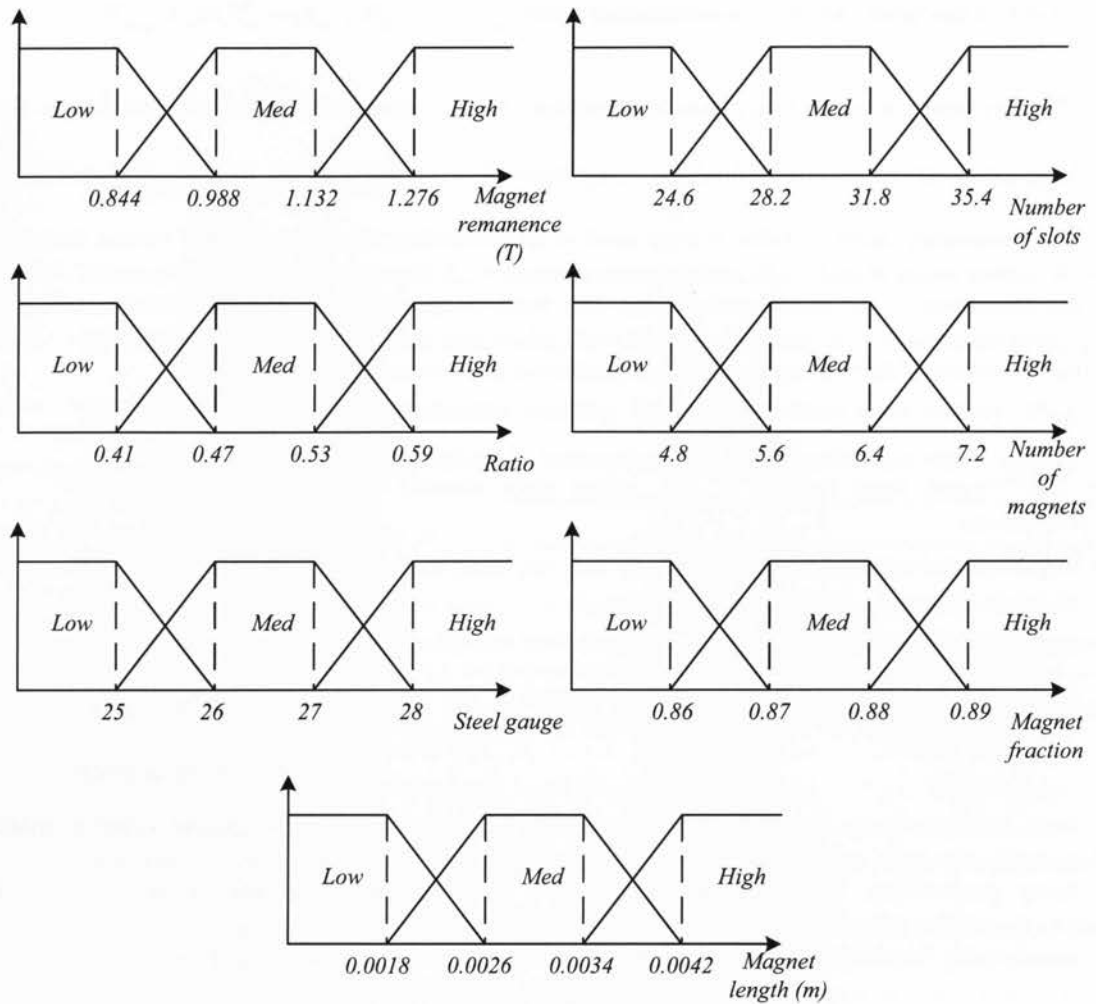


Figure 16. Membership function definitions for predicting attributes

We evaluated the quality of a BLDC motor from three perspectives, namely, weight, efficiency and motor constant. The quality measure is embedded as a scalar function into the EFS, so the system can use it to rank modification rules.

5.1.3.2 Consequent of fuzzy modification rules

There are several actions that the rule can take in the consequent of a modification rule. These actions are listed in Table 7. Regarding the actions which change the number of magnets or the number of slots Table 5 was used.

Table 7. The actions in the consequent of fuzzy modification rules

Action	
Air gap increase	Number of magnets increase
Air gap decrease	Number of magnet decrease
Magnet length increase	Change steel, lower steel gauge
Magnet length decrease	Change steel, higher steel gauge
Ratio increase	Change magnet, magnet remanence increase
Ratio decrease	Change magnet, magnet remanence decrease
Number of slots increase	Magnet fraction increase
Number of slots decrease	Magnet fraction decrease

Some of the mentioned actions are not very clear when it comes to applying them on an unsatisfactory design. For example, consider “air gap decrease” be the consequent of a sample modification rule. In the classical KBS the designer should define a constant value that would be decreased from the air gap length in all cases when the mentioned rule is fired. The advantage of the fuzzy approach is that, the system can decrease the value based on the amount that the antecedent of the rule is satisfied. More precisely, in a fuzzy system, the antecedent of each rule is satisfied to a degree in the range $[0, 1]$. The system can use this degree to decrease the air gap length proportionally. The following formula was used in the system to calculate the value that a certain variable would change as the result of an activated fuzzy rule:

$$Adaption_value = C \times act \quad (16)$$

The designer should assign a constant value for each numeric attribute that can be changed in the consequent of a fuzzy modification rule. Here, C represents the constant value for each numeric attribute, and act is the degree that the antecedent of the fuzzy rule was satisfied. The resulting $Adaption_value$ shows the absolute value that the numeric attribute should be changed.

5.1.3.3 Evolutionary System

We ran the evolutionary system with the set of parameters specified in Table 8 to derive modification rules from the casebase. The evolutionary system used for rule derivation was explained in section 4.2. We ran our system on a dual-processor 2.13 GHz machine with 2.0 GB of memory. Each run of the system for derivation of one modification rule took a processing time of 5 minutes.

Table 8. Evolutionary system parameters

Parameter	Value
Population size	250
Maximum generations	100
Initial number of terminals per rule	2
Maximum number of terminals per rule	5
Crossover rate	0.9
Mutation rate	0.05
Reproduction rate	0.1

It worth mentioning that during the rule derivation phase, we consider the whole casebase as the training set and we do not divide the casebase into the test and training sets. The reason lies in the special type of fitness function that we defined for our system. Although the fitness function gives a good comparison mechanism between two modification rules, the output value of the

fitness function is not very informative for the user because of its cumulative nature. Hence they are not reported as a measure for accuracy of the rules. The conventional way of deriving rules from the training set and evaluating the accuracy on the test set is not used here. Instead, the accuracy and effectiveness of the rules are evaluated when the rules are embedded as a part in the knowledge based system.

After running the evolutionary system the modification rules were derived. As we mentioned before, each run of the system would produce one modification rule, so the user can derive as many rules as needed by running the system several times. When each rule is extracted from the dataset, the user can take three actions. First, user can move the rule as it is to the set of modification rules in the knowledge based system. Second, user can make some modifications on the antecedent or consequent of the rule based on his expertise. Finally, the user might reject the validity of the rule and do not use it during the design process. Therefore, the user has complete control on the design process, meaning that all the modification rules are transparent to the user and the user can change them based on his experience. The list of modification rules derived from the casebase and the specifications given in section 5.1.1 are shown in Table 9.

Table 9. Some of the derived modification rules

Problem	Rule
Weight	IF (magnet remanence is not low) THEN change the magnet with another magnet which has lower remanence
	IF (number of magnets in not high) THEN increase number of magnets
	IF (magnet length is not low) THEN decrease magnet fraction
	IF (magnet remanence is low) THEN change the magnet with another magnet which has higher remanence

Efficiency	IF (number of magnets is not low) AND (rotor ratio is not high) THEN decrease number of magnets
	IF (number of magnets is not low) THEN change the steel with another steel which has lower gauge
Motor Constant	IF (number of slots is not low) THEN decrease number of slots
	IF (number of magnets is not low) THEN decrease air gap length
	IF (rotor ratio is high) THEN decrease rotor ratio

As you can see, there exist some contradictory rules among the modification rules derived for different problems. For example, the second rule derived for the weight problem suggests increasing the number of magnets to reduce the overall weight of the machine while the second rule induced for the efficiency problem suggests decreasing the number of magnets to increase the efficiency of the machine. The knowledge base system will tradeoff between these contradictory rules to find an optimal point where all the constraints are satisfied as much as possible.

5.2 Knowledge base system implementation

The KBS is developed in FuzzyCLIPS. CLIPS [6] (C Language Integrated Product System) is an expert system shell developed by NASA, which is widely used to develop rule and/or object based expert systems. CLIPS supports three paradigms of rules, objects and procedures for knowledge presentation. CLIPS has portability among different platforms, can be integrated with other programming languages, and has an interactive development environment.

FuzzyClips [58] is a fuzzy expert system shell, developed by the National research Council of Canada (NRC) as an extension to CLIPS. In addition to the capabilities available in the CLIPS environment, FuzzyCLIPS can deal with fuzzy concepts and reasoning. Therefore, the user can freely mix fuzzy and exact facts to make fuzzy rules. FuzzyCLIPS has two inexact concepts namely fuzziness which is implemented through fuzzy sets and relations, and uncertainty which is handled by certainty factors for rules and facts.

The KB is composed of three parts of frames, rules and procedural attachments. The complete description of each part is given in section 4.1. The rule section is composed of Design, Test, and Modification rules. Design rules were extracted from the conventional design method [36]. Complete list of design equations used are presented in Appendix A and B.

Seven rules were generated for the test section. Three of the rules were developed to check weight, efficiency, and the motor constant of a motor to see if they are satisfactory or not. One rule would delete identical designs in the KB, one rule checks to see if there exists a satisfactory design in the KB (a design which satisfies the mentioned three test rules) to stop system, another rule would delete off track designs, and the last rule in the test section was developed to change the control of the system from the test module to modification module.

The EFS is responsible for generating the modification rules. Some of the derived modification rules are given in Table 9. These rules deal with the problem of excessive weight and insufficient efficiency and motor constant. Among the modification rules, the first rule converts all the unsatisfactory designs developed into facts. The reason was that in the FuzzyCLIPS environment, only fuzzy facts can be placed in the right hand side of the rules. Therefore, one fact was generated for each instance that existed in the KB. These facts were used inside the

modification rules. The last rule in the modification section changes the control from the modification module to the test module, so the iteration between test and modification continues. The complete list of all rules designed for the system is given in appendix C.

The sample application of a modification rule on a motor is shown in Figure 17. Assume that *Motor* is an unsatisfactory design. The system will match the antecedent of the available modification rules with the specifications of the unsatisfactory *Motor* to see which rule can be fired. Assume that the *Modification Rule* is a fuzzy rule which was satisfied. First, the fuzzy value for the predicting parameters in the antecedent of the modification rule are calculated. Here the value for the rotor ratio is calculated and the number of magnets is read from the information stored in the *Motor* specifications. Next, the gathered information are applied on the fuzzy membership functions to calculate the membership degree of each element. As it is shown, the calculated rotor ratio value satisfies the condition of *rotor ratio is not low* to the degree of 0.5 and the extracted number of magnets value satisfies the condition of *number of magnets is not high* to the degree of 1. Next, the min operator is applied on the resulting values to calculate the satisfaction degree of the antecedent of the modification rule. The satisfaction degree would be 0.5 in the explained example. This value is later used in the consequent of the fuzzy rule. The consequent of the modification rule states that the value for the rotor ratio should be decrease. In order to decrease the value of the rotor ratio, equation (16) is used in the following order:

$$\begin{aligned} \text{New ratio} &= \text{Previous ratio} - C \times 0.5 \\ &= 0.44 - 0.01 \times 0.5 \\ &= 0.435 \end{aligned}$$

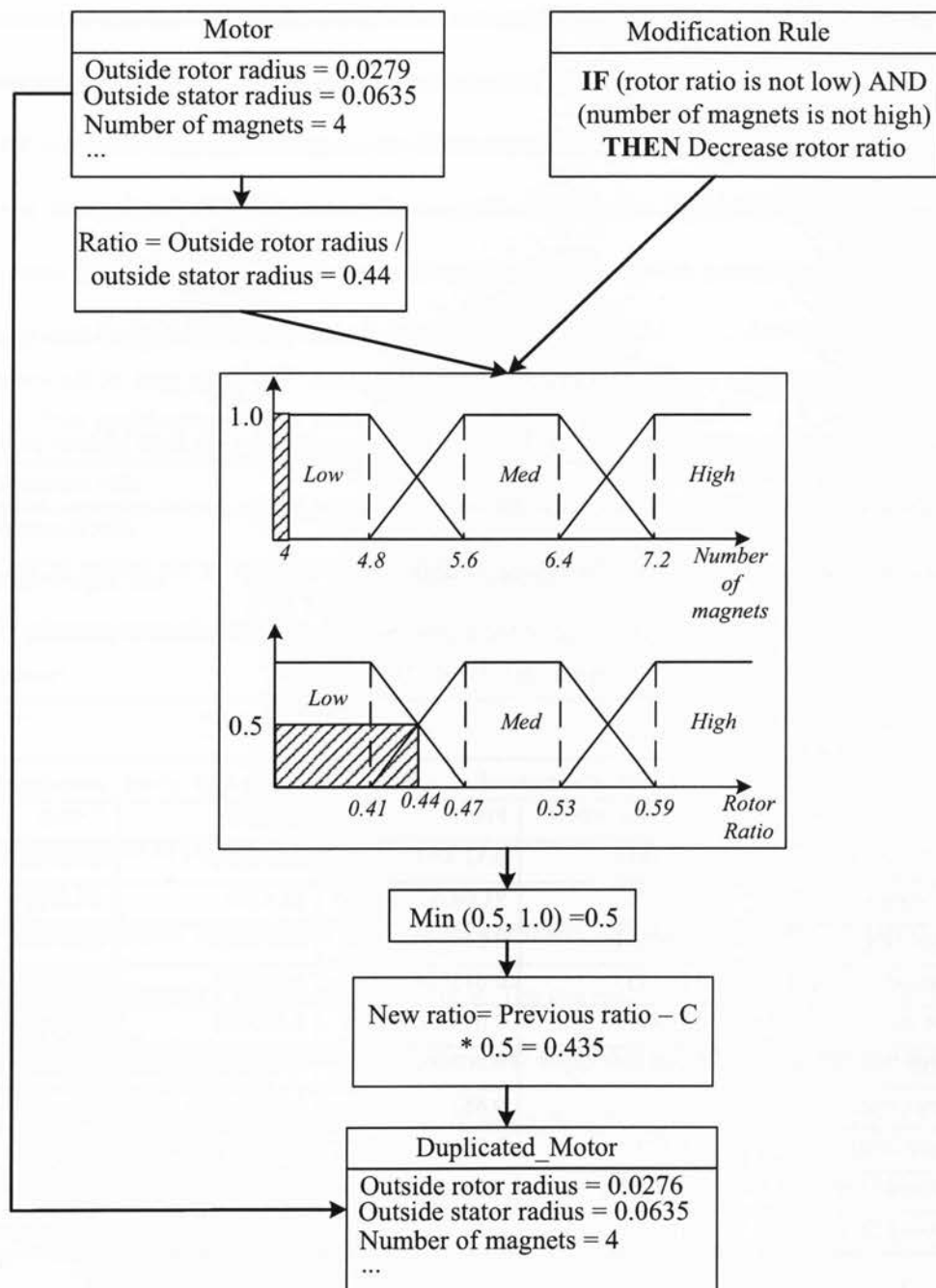


Figure 17. Application of a fuzzy modification rule on an unsatisfactory design

Where, the previous ratio equals 0.44 which was initially calculated to evaluate the antecedent of the modification rule. C is a constant value set by the designer which we set to be equal 0.01 here. The new value for the rotor ratio is calculated and assigned to a duplicate motor which was copied from the initial motor.

5.3 Results

Table 10 illustrates the results obtained by the KBS system. The main goal of the system is to maximize the motor constant parameter. Efficiency and weight have the next highest priority. All the designs developed by our knowledge based system have higher motor constant compared to the conventional design. We ran the system to derive three designs as the output of the system, where each design tries to improve one of the performance indices as much as possible.

Table 10. Results

MOTOR SPECIFICATIONS	DESIGNS			
	CONVENTIONAL	FIRST	SECOND	THIRD
Motor constant	0.513614	0.523094	0.796281	0.867846
Efficiency (%)	86.8608	91.0426	93.6190	89.6569
Weight (Kg)	2.264621	2.244910	2.711602	2.914547
Phase inductance (H)	0.013835	0.013214	0.004131	0.003425
Phase resistance (Ω)	0.708252	0.675322	0.287425	0.241494
Air gap flux (Wb)	0.000384	0.000390	0.000668	0.000747
Magnet fraction	0.9	0.882	0.882	0.9
Air gap length (m)	0.0026	0.0025	0.0025	0.0025
Proposed Magnet	Alnico-8B	Alnico-8B	Sm ₂ Co ₁₇	Neomax-35
Proposed Steel	Silicon-iron-24-1	Silicon-iron-26-1	Silicon-iron-26-1	Silicon-iron-24-1

In the first design, an increase in efficiency and motor constant and a decrease in the weight have been achieved by decreasing the magnet fraction and the air gap length. In the second design, the

magnetic material has been substituted with a stronger one. The 24 guage (0.025 thick) silicon iron electrical steel has been substituted with thinner 26 guage (0.0185 thick), which has reduced the total core loss and increased the total efficiency of the motor. In the last design, the magnet has been changed, which has increased the motor constant of the design but increased the weight at the same time. A knowledge driven decision can be made by the system user to select any of the suggested designs that better suits their application.

Table 11. Two modification rules in conventional and fuzzy KBSs

Modification rule	
Conventional rule	Fuzzy rule
IF (meet-weight no) THEN (change magnet – lower remanence)	IF (meet-weight no) AND (magnet remanence is not low) THEN (change magnet – lower remanence)

The mentioned fuzzy rule approach has the following advantages over the conventional KBS approach proposed in [2]:

1. More accurate suggestions: our system considers the status of each developed design when suggesting a modification rule for it. The conventional method follows the “one fits all” methodology where one modification was suggested for all motors. For example, two modification rules developed for modifying a design which does not satisfy the weight constraint are shown in Table 11. As you can see, the fuzzy rule is more informative, because it advises a new magnet with lower magnetic remanence only if the magnet remanence of the current magnet is not low. Although it may seem trivia in the first glance, but it is a very important point which should be considered when generating the modification rules.

2. More precise adaption: integration of fuzzy logic with our systems increases the accuracy of the modification rules compared with the conventional one. As shown in Figure 17, the consequent of each fuzzy rule is executed proportional with the degree that the antecedent of the rule was satisfied. This proportional execution increases the accuracy of each rule, and makes the rule more adaptable to each case.
3. Adaptable rules: In our system, modification rules are derived for each design case separately, while in the conventional system there is a predefined set of rules which are used in all the cases.
4. Less need for maintenance: One of the main problems with the classical approach of KBS is the need for maintenance. Our system derives the modification rules automatically, so there is no need for a knowledge engineer to maintain the rules. It should be also mentioned that both systems are decision support systems, meaning that the motor expert is involved in all stages of designing a motor.

6 Conclusion

Design of BLDC motors involves an iterative approach which starts with an initial design and then iterates between test and modification phases. We developed a system which imitates this approach in a KBS framework. The analytical, graphical, procedural and criteria knowledge were integrated into the KBS using frames, rules and procedures. For a given set of required performances, the system produces several alternate designs that meet the required performances as much as possible, but have different specifications. The designers can select the one that best meets their priorities requirements.

We used an Evolutionary Fuzzy System (EFS) to derive the modification rules from a predefined casebase. The EFS consists of a population of individuals where each individual encodes the antecedent and consequent of a fuzzy modification rule. An evolutionary algorithm evolves the population of fuzzy rules. The system was implemented and the effectiveness of the system was shown by comparing the results with those of the conventional method.

In summary, our system includes the benefits of both knowledge based and soft computing approaches proposed for the design of electrical machines. Knowledge acquisition problem from human experts, which is a common problem of conventional KBS is not an issue in our system because we use an automatic method for derivation of our rules. On the other hand, the base structure of our system is a KBS so transparency and integration of human expert's knowledge is not a drawback of our system.

The proposed system uses evolutionary computation to derive modification rules, and evolutionary computation methods are known to be computationally intensive. This increases the running time of the system, and makes the system unsuitable to be utilized in an interactive

manner. Possible integration of faster algorithms, such as decision trees, could be an interesting future work.

Also, the system was applied for the design of BLDC motors in this thesis. The principles of the same system could be used for the design of other electromagnetic devices or more generally for the design of any device whose design steps could be broken down into the three phases of design, test, and modify. An interesting future work could be the application of the same principles for the design of other devices.

Appendix A. Design equations for the radial flux BLDC

motor [36]

List of the equations used during the design phase is listed below:

Parameter	Description	
$\omega_m = (\pi/30)S_r$	Mechanical speed	(A.1)
$\omega_e = (N_m/2)\omega_m$	Electrical speed	(A.2)
$f_e = \omega_e/(2\pi)$	Fundamental electrical frequency	(A.3)
$T = 746P_{hp}/\omega_m$	Torque from horsepower	(A.4)
$N_s = N_{sp}N_{ph}$	No. of slots	(A.5)
$N_{spp} = N_{sp}/N_m$	No. of slots per pole per phase	(A.6)
$N_{sm} = N_{spp}N_{ph}$	No. of slots per pole	(A.7)
$\alpha_{cp} = \text{int}(N_{spp})/N_{spp}$	Coil-pole fraction	(A.8)
$\theta_p = 2\pi/N_m$	Angular pole pitch	(A.9)
$\theta_s = 2\pi/N_s$	Angular slot pitch	(A.10)

$$\theta_{se} = \pi / N_{sm}$$

Slot pitch (A.11)

$$R_{si} = R_{ro} + g$$

Inside stator radius (A.12)

$$\tau_p = R_{si} \theta_p$$

Pole pitch (A.13)

$$\tau_c = \alpha_{cp} \tau_p$$

Coil pitch (A.14)

$$\tau_s = R_{si} \theta_s$$

Slot pitch at air gap (A.15)

$$w_t = \tau_s - w_s$$

Tooth width at air gap (A.16)

$$k_d = \frac{\sin(N_{spp} \theta_{se} / 2)}{N_{spp} \sin(\theta_{se} / 2)}$$

Distribution factor (A.17)

$$k_p = \alpha_{cp}$$

Pitch factor (A.18)

$$k_s = 1 - \theta_{se} / (2\pi)$$

Skew factor (A.19)

$$C_\phi = \frac{2\alpha_m}{1 + \alpha_m}$$

Flux concentration factor (A.20)

$$P_c = l_m / (g C_\phi)$$

Permeance coefficient (A.21)

$$k_{ml} = 1 + \frac{4l_m}{\pi \mu_R \alpha_m \tau_p} \ln \left[1 + \pi \frac{g}{(1 - \alpha_m) \tau_p} \right]$$

Magnet leakage factor (A.22)

$$g_c = g + l_m / \mu_R$$

Effective air gap for (A.23)

Carter coefficient

$$k_c = \left[1 - \frac{1}{\frac{\tau_s}{w_s} \left(5 \frac{g_c}{w_s} + 1 \right)} \right]^{-1}$$

Carter coefficient (A.24)

$$A_g = \frac{\tau_p L (1 + \alpha_m)}{2}$$

Air gap area (A.25)

$$B_g = \frac{C_\phi}{1 + \mu_R k_c k_{ml} / P_c} B_r$$

Air gap flux density (A.26)

$$\phi_g = B_g A_g$$

Air gap flux (A.27)

$$w_{bi} = \frac{\phi_g}{2 B_{\max} k_{st} L}$$

Back iron width (A.28)

$$w_{tb} = \frac{2}{N_{sm}} w_{bi}$$

Tooth width (A.29)

$$R_{sb} = R_{so} - w_{bi}$$

Stator back iron radius (A.30)

$$R_{ri} = R_{ro} - l_m - w_{bi}$$

Rotor inside radius (A.31)

$$w_{sb} = R_{sb} \theta_s - w_{tb}$$

Slot bottom width (A.32)

$$w_{si} = (R_{si} + \alpha_{sd} w_{tb}) \theta_s - w_{tb}$$

Slot width inside shoes (A.33)

$$\alpha_s = \frac{w_{si}}{w_{si} + w_{tb}}$$

Slot fraction inside shoes (A.34)

$$d_s = R_{sb} - R_{ro} - g$$

Total slot depth (A.35)

$$d_3 = d_2 - \alpha_{sd} w_{tb}$$

Conductor slot depth (A.36)

$$d_1 + d_2 = \alpha_{sd} w_{tb}$$

$$\text{Shoe depth} \quad (\text{A.37})$$

$$A_s = d_3 [\theta_s (R_{sb} - d_3 / 2) - w_{tb}]$$

$$\text{Conductor area} \quad (\text{A.38})$$

$$n_s = \text{int} \left(\frac{E_{\max}}{N_m k_d k_p k_s B_g L R_{ro} N_{spp} \omega_m} \right)$$

$$\text{No. of turns per slot} \quad (\text{A.39})$$

$$e_{\max} = N_m k_d k_p k_s B_g L R_{ro} N_{spp} n_s \omega_m$$

$$\text{Peak back emf} \quad (\text{A.40})$$

$$I_s = \frac{T}{N_m k_d k_p k_s B_g L R_{ro} N_{spp}}$$

$$\text{Peak slot current} \quad (\text{A.41})$$

$$I_{ph} = \frac{I_s}{N_{ph} n_s}$$

$$\text{Phase current} \quad (\text{A.42})$$

$$J_c = \frac{I_s}{k_{cp} A_s}$$

$$\text{Peak conductor current} \quad (\text{A.43})$$

density

$$|B_s|_{\max} = \frac{\mu_0 I_s}{w_s}$$

$$\text{Peak slot flux density} \quad (\text{A.44})$$

$$R_s = \frac{\rho n_s^2 L}{k_{cp} A_s}$$

$$\text{Slot resistance} \quad (\text{A.45})$$

$$R_e = \frac{\rho n_s^2 \pi \tau_c}{2 k_{cp} A_s}$$

$$\text{End turn resistance} \quad (\text{A.46})$$

$$R_{ph} = N_{sp} (R_s + R_e)$$

$$\text{Phase resistance} \quad (\text{A.47})$$

$$L_g = \frac{n_s^2 \mu_R \mu_0 L \tau_c k_d}{4(l_m + \mu_R k_c g)}$$

$$\text{Air gap inductance} \quad (\text{A.48})$$

$$L_s = n_s^2 \left[\frac{\mu_0 d_3^2 L}{3 A_s} + \frac{\mu_0 d_2 L}{(w_s + w_{si})/2} + \frac{\mu_0 d_1 L}{w_s} \right]$$

$$\text{Slot leakage inductance} \quad (\text{A.49})$$

$$L_e = \frac{n_s^2 \mu_0 \tau_c}{8} \ln \left(\frac{\tau_c^2 \pi}{4 A_s} \right) \quad \text{End turn inductance} \quad (\text{A.50})$$

$$L_{ph} = N_{sp} (L_g + L_s + L_e) \quad \text{Phase inductance} \quad (\text{A.51})$$

$$P_r = N_{ph} I_{ph}^2 R_{ph} \quad \text{Ohmic power loss} \quad (\text{A.52})$$

$$V_{st} = [\pi (R_{so}^2 - R_{si}^2) - N_s A_s] L k_{st} \quad \text{Stator steel volume} \quad (\text{A.53})$$

$$P_{cl} = \rho_{bi} V_{st} \Gamma(B_{\max}, f_e) \quad \text{Core loss} \quad (\text{A.54})$$

Appendix B. Determination of hysteresis and Eddy current coefficients for calculation of the total core loss of the motor [83]

The total core loss of the motor can be divided into the hysteresis and eddy losses. These losses can be formulated as:

$$P_h = \left(\frac{W}{D10^7} \right) f_n B^x \quad \text{Hysteresis loss} \quad (\text{B.1})$$

$$P_e = \left(\frac{1.6545d^2W}{D10^{16}} \right) f^2 B^2 \lambda \quad \text{Eddy loss} \quad (\text{B.2})$$

$$P = P_h + P_e \quad \text{Total core loss} \quad (\text{B.3})$$

Where, W = weight of core, f = frequency, d = thickness of core lamination, B = flux density, D = density of core material, n = hysteresis loss coefficient, λ = eddy loss coefficient, x = hysteresis loss exponent for B. For two frequencies f_1 and f_2 and two flux densities B_1 and B_2 the following equations can be derived:

$$P_1 = \left(\frac{W}{D10^7} \right) f_1 B_1^x + \left(\frac{1.6545d^2W}{D10^{16}} \right) f_1^2 B_1^2 \lambda \quad (\text{B.4})$$

$$P_2 = \left(\frac{W}{D10^7} \right) f_2 B_2^x + \left(\frac{1.6545d^2W}{D10^{16}} \right) f_2^2 B_2^2 \lambda \quad (\text{B.5})$$

$$P_3 = \left(\frac{W}{D10^7} \right) f_1 B_2^x + \left(\frac{1.6545d^2W}{D10^{16}} \right) f_1^2 B_2^2 \lambda \quad (\text{B.6})$$

Solving equations (B.4) through (B.6) will result in the following equations for the coefficients and hysteresis exponents. Here, we assume that $a=f_2/f_1$.

$$x = \frac{\log \left[\frac{B_2^2 (P_2 - a^2 P_3)}{B_2^2 a P_1 (1 - a) + B_1^2 (P_2 - a P_3)} \right]}{\log(B_2 / B_1)} \quad (\text{B.7})$$

$$\lambda = \frac{P_2 B_1^x - a P_1 B_2^x}{\left(\frac{1.6545d^2W}{D10^{16}} \right) f_2^2 / a (a B_2^2 B_1^x - B_1^2 B_2^x)} \quad (\text{B.8})$$

$$n = \frac{P_3 - f_1^2 \lambda \left(\frac{1.6545d^2W}{D10^{16}} \right) B_2^2}{f_1 \left(\frac{W}{D10^7} \right) B_2^x} \quad (\text{B.9})$$

Appendix C. List of rules in the KB of the system

Number	Rule
1	Design::1-Calculate-Mechanical-Speed
2	Design::2-Calculate-Electrical-Speed
3	Design::3-Calculate-Electrical-Frequency
4	Design::4-Calculate-Number-Of-Slots
5	Design::5-Calculate-Number-Of-Slots-Per-Pole-Per-Phase
6	Design::6-Calculate-Number-Of-Slots-Per-Pole
7	Design::7-Calculate- Coil-Pole-Fraction
8	Design::8-Calculate-Angular-Pole-Pitch
9	Design::9-Calculate-Angular-Slot-Pitch
10	Design::10-Calculate-Slot-Pitch
11	Design::11-Calculate-Inside-Stator-Radius
12	Design::12-Calculate-Pole-Pitch
13	Design::13-Calculate-Coil-Pitch
14	Design::14-Calculate-Slot-Pitch-At-Air-Gap
15	Design::15-Calculate-Tooth-Width-At-Air-Gap
16	Design::16-Calculate-Distribution-Factor
17	Design::17-Calculate-Pitch-Factor
18	Design::18-Calculate-Skew-Factor
19	Design::19-Calculate-Flux-Concentration-Factor
20	Design::20-Calculate-Permeance-Coefficient
21	Design::21-Calculate-Magnet-Leakage-Factor
22	Design::22-Calculate-Effective-Air-Gap
23	Design::23-Calculate-Carter-Coefficient
24	Design::24-Calculate-Air-Gap-Area
25	Design::25-Calculate-Air-Gap-Flux-Density
26	Design::26-Calculate-Air-Gap-Flux
27	Design::27-Calculate-Back-Iron-Width
28	Design::28-Calculate-Tooth-Width
29	Design::29-Calculate-Stator-Back-Iron-Radius
30	Design::30-Calculate-Rotor-Inside-Radius
31	Design::31-Calculate-Slot-Bottom-Width
32	Design::32-Calculate-Slot-Width-Inside-Shoes

33	Design::33-Calculate-Slot-Fraction-Inside-Shoes
34	Design::34-Calculate-Total-Slot-Depth
35	Design::35-Calculate-Conductor-Slot-Depth
36	Design::36-Calculate-Shoe-Depth
37	Design::37-Calculate-Conductor-Area
38	Design::38-Calculate-No-Of-Turns-Per-Slot
39	Design::39-Calculate-Peak-Back-Emf
40	Design::40-Calculate-Peak-Slot-Current
41	Design::41-Calculate-Phase-Current
42	Design::42-Calculate-Peak-Conductor-Current-Density
43	Design::43-Calculate-Peak-Slot-Flux-Density
44	Design::44-Calculate-Slot-Resistance
45	Design::45-Calculate-End-Turn-Resistance
46	Design::46-Calculate-Phase-Resistance
47	Design::47-Calculate-Air-Gap-Inductance
48	Design::48-Calculate-Slot-Leakage-Inductance
49	Design::49-Calculate-End-Turn-Inductance
50	Design::50-Calculate-Phase-Inductance
51	Design::51-Calculate-Steel-Volume
52	Design::52-Calculate-Ohmic-Power-Loss
53	Design::53-Calculate-Core-Loss
54	Design::54-Calculate-Efficiency
55	Design::55-Calculate-Motor-Constant
56	Design::56-Calculate-Wire-Gauge
57	Design::57-Calculate-Steel-Volume-Weight
58	Design::58-Calculate-Magnet-Volume
59	Design::59-Calculate-Wire-Weight
60	Design::60-Calculate-Weight
61	Design::61-Copy-Input-Parameteres
62	Test::1-Test-Efficiency
63	Test::2-Test-Weight
64	Test::3- Test-Motor-Constant
65	Test::4-Design-Satisfactory
66	Test::5-Delete-Identical-Designs
67	Test::6-Delete-Off-Track-Designs

68	Test::7-Deactivate-Test-Activate-Modify
69	Modify::1-Convert-Instance-to-Fact
70	Modify::2-Motor-Constant-Problem-NumberofMagnetInc
71	Modify::3-Motor-Constant-Problem-MagnetRemInc
72	Modify::4-Motor-Constant-Problem-NumberofSlotDecMagnet
73	Modify::5-Motor-Constant-Problem-AirgapDec
74	Modify::6-Motor-Constant-Problem-RotoRatioDec
75	Modify::7-Motor-Constant-Problem-MagnetFracDec
76	Modify::8-Efficiency-Problem-MagnetRemInc
77	Modify::9-Efficiency-Problem-NumberofMagnetDec
78	Modify::10-Efficiency-Problem-SteelcorelossDec
79	Modify::11-Efficiency-Problem-NumberofSlotDec
80	Modify::12-Efficiency-Problem-RotoRatioDec
81	Modify::13-Weight-Problem-MagnetRemDec
82	Modify::14-Weight-Problem-NumberofMagnetInc
83	Modify::15-Weight-Problem-MagnetFracDec
84	Modify::16-Weight-Problem-RotoRatioDec
85	Modify::17-Weight-Problem-MagnetLenDec
86	Modify::18-Weight-Problem-AirgapInc
87	Modify::19-Weight-Problem-ChangeSteel
88	Modify::20-Delete-Off-Track-Designs

References

- [1] V. Akbarzadeh, A. Sadeghian, and M.V. dos Santos, "Inducing Relational Fuzzy Classification Rules by means of Cooperative Coevolution," in *Foundations of Computational Intelligence Vol 4: Bio-Inspired Data Mining Theoretical Foundations and Applications, Studies in Computational Intelligence, Chapter 6*. Springer Verlag, Germany, 2009.
- [2] V. Akbarzadeh, and A. Sadeghian, "A Knowledge Based Decision Support Architecture for Designing Brushless DC Motors," in *Proc. of the 14th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2009)*, 2009.
- [3] J.S. Albus and A.M. Meystel, *Engineering of Mind: An Introduction to the Science of Intelligent Systems*, John Wiley and Sons, 2001.
- [4] A. A. Arkadan, M. N. ElBsat and M. A. Mneimneh, "Particle Swarm Design Optimization of ALA Rotor SynRM for Traction Applications," *IEEE Trans. Magn.*, vol. 45, pp. 956-959, 2009.
- [5] A. A. Arkadan, M. Mneimneh and N. Al-Aawar, "R-FL-C Model for Design Optimization of PM Generators," *IEEE Trans. Magn.*, vol. 43, pp. 1649-1652, 2007.
- [6] Artificial Intelligence Section, *CLIPS Reference Guide Volume I Basic Programming Guide, CLIPS Version 6.3*, December 2007.
- [7] W. Banzhaf, *Genetic Programming: An Introduction on the Automatic Evolution of Computer Programs and its Applications*. Morgan Kaufmann, 1998.

- [8] W. Banzhaf, J. Koza, C. Ryan, L. Spector and C. Jacob, "Genetic programming," *IEEE Intelligent Systems and their Applications*, vol. 15, pp. 74-84, 2000.
- [9] N. Bianchi and S. Bolognani, "Brushless DC motor design: An optimisation procedure based on genetic algorithms," in *Electrical Machines and Drives, 1997 Eighth International Conference on (Conf. Publ. no. 444)*, pp. 16-20, 1997.
- [10] R. H. Bonczek, C. W. Holsapple and A. B. Whinston, "The evolving roles of models in decision support systems," *Decision Sciences*, vol. 11, pp. 337-356, 1980.
- [11] D. C. Brown and B. Chandrasekaran, *Design Problem Solving: Knowledge Structures and Control Strategies*. 1989.
- [12] D. C. Brown and B. Chandrasekaran, "Expert systems for a class of mechanical design activity," in *Knowledge Engineering in Computer-Aided Design: Proceedings of the IFIP WG 5.2 Working Conference on Knowledge Engineering in Computer-Aided Design*, pp. 259, 1985.
- [13] F. Burstein, *Handbook on Decision Support Systems: Basic Themes*. Springer-Verlag New York Inc, 2007.
- [14] M. Chiampi, C. Ragusa and M. Repetto, "Fuzzy approach for multiobjective optimization in magnetics," *IEEE Trans. Magn.*, vol. 32, pp. 1234-1237, 1996.
- [15] C. Choi, D. Lee and K. Park, "Fuzzy Design of a Switched Reluctance Motor Based on the Torque Profile Optimization," *IEEE Trans. Magn.*, vol. 36, pp. 3548-3550, 2000.

- [16] O. Cordón, F. Herrera, F. Hoffmann and L. Magdalena, *Genetic Fuzzy Systems: Evolutionary Tuning and Learning of Fuzzy Knowledge Bases*. World Scientific Pub Co Inc, 2001.
- [17] O. Cordon, F. Herrera and P. Villar, "Generating the knowledge base of a fuzzy rule-based system by the genetic learning of the data base," *IEEE Trans. Fuzzy Syst.*, vol. 9, pp. 667-674, 2001.
- [18] T. Efraim, E. A. Jay, T. P. Liang and R. V. McCarthy, "Decision support systems and intelligent systems," *London: Prentice-Hall International*, vol. 890, pp. 46-418, 1998.
- [19] A. Emadi, *Energy-Efficient Electric Motors*. CRC Press, 2005.
- [20] M. Enokizono and Y. Tsuchida, "Optimal design by boundary element method with fuzzy inference," *IEEE Trans. Magn.*, vol. 30, pp. 3447-3450, 1994.
- [21] E.A. Feigenbaum, "Knowledge Engineering for the 1980s," *Computing Science Dept, Stanford University*, 1981.
- [22] C. Ferreira, "Gene expression programming: A new adaptive algorithm for solving problems," *Complex Systems*, vol. 13, pp. 87-130, 2001.
- [23] L.J. Fogel, "Autonomous automata," *Industrial Research*, vol. 4, pp. 14-19, 1962.
- [24] B. Frada, *Handbook on decision support systems 1: Basic themes (International handbooks on information systems)*, Springer, 2008.

- [25] F. Francois and J. Bignon, "Integration of fuzzy techniques in a CAD/CAM system," *IEEE Trans. Magn.*, vol. 31, pp. 1996-1999, 1995.
- [26] J. H. Garret Jr and A. Jain, "A knowledge-based system for designing transformers and inductors," in *Artificial Intelligence Applications, 1988., Proceedings of the Fourth Conference on*, pp. 96-101, 1988.
- [27] J. C. Giarratano and G. Riley, *Expert Systems: Principles and Programming*. PWS Publishing Co. Boston, MA, USA, 1994.
- [28] J. F. Gieras and M. Wing, *Permanent Magnet Motor Technology: Design and Applications*. CRC, 2002.
- [29] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA 1989.
- [30] H. Groschwitz and J. Holtz, "An expert system on transformer design for switched mode power supplies," in *Industrial Electronics, Control and Instrumentation, 1994. IECON'94., 20th International Conference on*, 1994.
- [31] T. R. Gruber, "Acquiring strategic knowledge from experts," 1990.
- [32] F. G. Guimarães, P. H. L. Barros and J. A. Ramírez, "An object-oriented library based on computational intelligence techniques for optimization in electromagnetic," *IEEE Trans. Magn.*, vol. 39, pp. 2121-2124, 2003.

- [33] F. G. Guimaraes, F. Campelo, R. R. Saldanha and J. A. Ramirez, "A hybrid methodology for fuzzy optimization of electromagnetic devices," *IEEE Trans. Magn.*, vol. 41, pp. 1744-1747, 2005.
- [34] A. Handbook, "prepared under the direction of the ASM international Handbook committee," *Metals Handbook*, 1990.
- [35] D. C. Hanselman, *Brushless Permanent Magnet Motor Design*, 2nd ed., Magna Physics Publishing, 2006.
- [36] D. C. Hanselman, *Brushless Permanent Magnet Motor Design*. New York: McGraw-Hill, 1994.
- [37] J. R. Hendershot and T. J. E. Miller, *Design of Brushless Permanent-Magnet Motors*. Magna Pysics Pub., 1994.
- [38] F. Hoffmann and G. Pfister, "Evolutionary design of a fuzzy knowledge base for a mobile robot," *International Journal of Approximate Reasoning*, vol. 17, pp. 447-469, 1997.
- [39] J. H. Holland, "Adaptation in natural and artificial systems." *Ann Arbor MI: University of Michigan Press*, 1975.
- [40] C. W. Holsapple, and A. B. Whinston, *Decision Support Systems: A Knowledge-Based Approach*. West Publishing Company Minneapolis/St. Paul, 1996.
- [41] S. H. Huang, "Dimensionality reduction in automatic knowledge acquisition: A simple greedy search approach," *IEEE Computer Society*, 2003.

- [42] L. Hui, H. Li, H. Bei and Y. Shunchang, "Application research based on improved genetic algorithm for optimum design of power transformers," in *Electrical Machines and Systems, 2001. ICEMS 2001. Proceedings of the Fifth International Conference on*, 2001.
- [43] H. Ishibuchi, T. Morisawa and T. Nakashima, "Voting schemes for fuzzy-rule-based classification systems," in *Fuzzy Systems, proceedings of the Fifth IEEE International Conference on*, 1996.
- [44] J. S. R. Jang, C. T. Sun and E. Mizutani, *Neuro-Fuzzy and Soft Computing*. Prentice Hall Upper Saddle River, 1997.
- [45] J. K. Kishore, L. M. Patnaik, V. Mani and V. K. Agrawal, "Application of genetic programming for multicategory pattern classification," *IEEE Transactions on Evolutionary Computation*, vol. 4, pp. 242-258, 2000.
- [46] J. Kolodner, *Case-Based Reasoning*. Morgan Kaufmann Publishers Inc. San Francisco, CA, USA, 1993.
- [47] J. R. Koza, "Human-competitive machine invention by means of genetic programming," *AI EDAM*, vol. 22, pp. 185-193, 2008.
- [48] J. R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. The MIT press, 1992.
- [49] C. K. Kwong and S. M. Tam, "Case-based reasoning approach to concurrent design of low power transformers," *Journal of Materials Processing Tech.*, vol. 128, pp. 136-141, 2002.

- [50] D. A. Lowther, "Automating the design of low frequency electromagnetic devices—a sensitive issue," *COMPEL: The International Journal for Computation and Mathematics in Electrical and Electronic Engineering*, vol. 22, 2003.
- [51] D. A. Lowther, "Accelerating the computer-based simulation and design of electromagnetic devices by applying knowledge," *IEEE Trans. Magn.*, vol. 38, pp. 1105-1108, 2002.
- [52] D. A. Lowther, "Knowledge based systems in electrical machine design," in *Electrical Machines and Drives, 1993. Sixth International Conference on (Conf. Publ. no. 376)*, pp. 328-333, 1993.
- [53] E. H. Mamdani, "Application of fuzzy algorithms for control of simple dynamic plant," *Proc.Iee*, vol. 121, pp. 1585-1588, 1974.
- [54] A.M. McCosh, and B.A. Correa-Perez, "The Optimization of What?," *Gupta, J.G. Forgonne, and M. Mora, Intelligent Decision-making Support Systems: Foundations, Applications and Challenges*, Springer-Verlag, pp. 475-494, 2006.
- [55] R. R. F. Mendes, F. de B Voznika, A. A. Freitas and J. C. Nievola, "Discovering fuzzy classification rules with genetic programming and co-evolution," *Lecture Notes in Computer Science*, pp. 314-325, 2001.
- [56] B. Mirzaeian, M. Moallem, V. Tahani and C. Lucas, "Multiobjective optimization method based on a genetic algorithm for switched reluctance motor design," *IEEE Trans. Magn.*, vol. 38, pp. 1524-1527, 2002.

- [57] A. S. Nookabadi and J. E. Middle, "A knowledge-based system as a design aid for quality assurance information systems," *International Journal of Quality and Reliability Management*, vol. 18, pp. 657-671, 2001.
- [58] B. Orchard, "FuzzyCLIPS Version 6.10 d User's Guide," *Integrated Reasoning Group, Institute for Information Technology, National Research Council*, 2004.
- [59] J. T. Park, C. G. Lee, M. K. Kim and H. K. Jung, "Application of Fuzzy Decision to Optimization of Induction Motor Design," *IEEE Transactions on Magnetics*, vol. 33, pp. 1939-1942, 1997.
- [60] C. A. Pena-Reyes, *Coevolutionary Fuzzy Modeling*, 2002.
- [61] C. A. Pena-Reyes, and M. Sipper, "Fuzzy CoCo: A cooperative-coevolutionary approach to fuzzy modeling," *IEEE Transactions on Fuzzy Systems*, vol. 9, pp. 727-737, 2001.
- [62] K. Preis, O. Biro, M. Friedrich, A. Gottvald and C. Magele, "Comparison of different optimization strategies in the design of electromagnetic devices," *IEEE Trans. Magn.*, vol. 27, pp. 4154-4157, 1991.
- [63] N. C. Proudlove, S. Vadera and K. A. H. Kobbacy, "Intelligent management systems in operations: a review," *Journal of the Operational Research Society*, pp. 682-699, 1998.
- [64] K. Rashid, J. A. Ramirez and E. M. Freeman, "Optimization of electromagnetic devices using computational intelligence techniques," *IEEE Trans. Magn.*, vol. 35, pp. 3727-3729, 1999.

- [65] I. Rechenberg, "Evolution strategy: optimization of technical systems according to the principles of biological evolution," *Frommann-Holzboog*, Stuttgart, 1973.
- [66] T. Renyuan, S. Jianzhong, L. Yan and C. Xiang, "Optimization of electromagnetic devices by using intelligent simulated annealing algorithm," *IEEE Trans. Magn.*, vol. 34, pp. 2992-2995, 1998.
- [67] R. W. Rong and D. A. Lowther, "Storage and retrieval of solutions in the design of electromagnetic devices," *IEEE Trans. Magn.*, vol. 30, pp. 3648-3651, 1994.
- [68] A. Sadeghian and J. D. Lavers, "Implementation of knowledge-based system for iron core inductor design," *IEEE Trans. Magn.*, vol. 40, pp. 3495-3504, 2004.
- [69] C. M. Saldanha and D. A. Lowther, "Knowledge-based computation of electromagnetic device parameters," *IEEE Trans. Magn.*, vol. 24, pp. 334-337, 1988.
- [70] R. M. Sassine and D. A. Lowther, "A knowledge-based environment for linking electromagnetic design tools," *IEEE Trans. Magn.*, vol. 30, pp. 3652-3655, 1994.
- [71] H. P. Schwefel, *Evolution and Optimum Seeking*. Wiley-Interscience, 1995.
- [72] H. Y. Shin and J. W. Lee, "An expert system for pneumatic design," *AI EDAM*, vol. 12, pp. 3-11, 1998.
- [73] H. Simon, *The new science of management decisions*, Rev. ed. Englewood Cliffs, NJ: Prentice-Hall, 1982.

- [74] S. F. Smith, "Flexible learning of problem solving heuristics through adaptive search," in *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*, 1983.
- [75] D. Sriram, G. Stephanopoulos, R. Logcher, D. Gossard, N. Groleau, D. Serrano and D. Navinchandra, "Knowledge-Based System Applications in Engineering Design: Research at MIT." *AI Magazine*, vol. 10, pp. 79-96, 1989.
- [76] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Trans. Syst. Man Cybern.*, vol. 15, pp. 116-132, 1985.
- [77] H. A. Toliyat and G. B. Kliman, *Handbook of Electric Motors*. CRC, 2004.
- [78] T. Tomiyama, "Intelligent computer-aided design systems: Past 20 years and future 20 years," *AI EDAM*, vol. 21, pp. 27-29, 2007.
- [79] E. Turban, *Decision Support and Expert Systems: Management Support Systems*. Prentice Hall PTR Upper Saddle River, NJ, USA, 1993.
- [80] P. R. Upadhyay and K. R. Rajagopal, "Genetic algorithm based design optimization of a permanent magnet brushless dc motor," *J. Appl. Phys.*, vol. 97, pp. 10Q516, 2005.
- [81] Q. Vo and D. A. Lowther, "A paradigm for the non-routine design of electromagnetic devices using a case based reasoning approach," *IEEE Trans. Magn.*, vol. 36, pp. 1669-1672, 2000.
- [82] L. X. Wang, *Adaptive Fuzzy Systems and Control: Design and Stability Analysis*. Prentice-Hall, Inc. Upper Saddle River, NJ, USA, 1994.

- [83] W. H. Yeadon and A. W. Yeadon, *Handbook of Small Electric Motors*. McGraw-Hill Professional, 2001.
- [84] L. A. Zadeh, "Knowledge representation in fuzzy logic," *IEEE Trans. Knowled. Data Eng.*, vol. 1, pp. 89-100, 1989.
- [85] L. A. Zadeh, "Outline of a new approach to the analysis of complex systems and decision processes," *IEEE Trans. Syst. Man Cybern.*, vol. 3, pp. 28-44, 1973.
- [86] L. A. Zadeh, "Fuzzy Sets," *Information and Control*, vol. 8, pp. 338-353, 1965.
- [87] H. J. Zimmermann, *Fuzzy Set Theory--and its Applications*. Kluwer Academic Pub, 2001.