INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED'09
24 - 27 AUGUST 2009, STANFORD UNIVERSITY, STANFORD, CA, USA

TOWARD A SOFT COMPUTING INTEGRATED INTELLIGENT DESIGN FRAMEWORK

Xuan F. Zha
Extension Systems International, USA

ABSTRACT
In this paper, a novel soft computing integrated intelligent framework is proposed for virtual product design and prototyping based upon the hybrid intelligence and soft computing techniques. A comprehensive evolutionary neuro-fuzzy model (EFNN), which is a five-layer fuzzy (rule-based) neural network that combines fuzzy logic, neural networks, and genetic algorithms, is developed to support modeling, analysis and evaluation, and optimization tasks in the complex design process. The developed system (EFNN-HIDS) provides a unified, integrated intelligent environment for virtual prototyping, design and simulation, and customization of complex engineering systems and products. A case study for mass customization design of workstation tables is provided to verify and illustrate the proposed model and framework.

Keywords: Hybrid intelligent design system, soft computing, computational intelligence, evolutionary neuro-fuzzy model, virtual design and prototyping, mass customization, simulation, integration

1 INTRODUCTION

Over the past two decades, artificial intelligence (AI) techniques have emerged as a main contender for design techniques. Many attempts have been made in applying AI techniques in particular computational intelligence (CI) such as fuzzy logic, neural networks, and genetic algorithms to accomplish some design tasks [3,13,19]. While these individual AI techniques have produced encouraging results, design problem is too complex to be solved by using a single AI technique alone. This is because each AI technique has particular strengths and weaknesses that make it suited for particular problems but not for others [11]. For instance, while neural networks are appropriate for recognizing patterns, but they are generally not good at explaining how they reach their decisions. Similarly, fuzzy systems can explain decisions well but they cannot automatically acquire the rules used to make decisions.

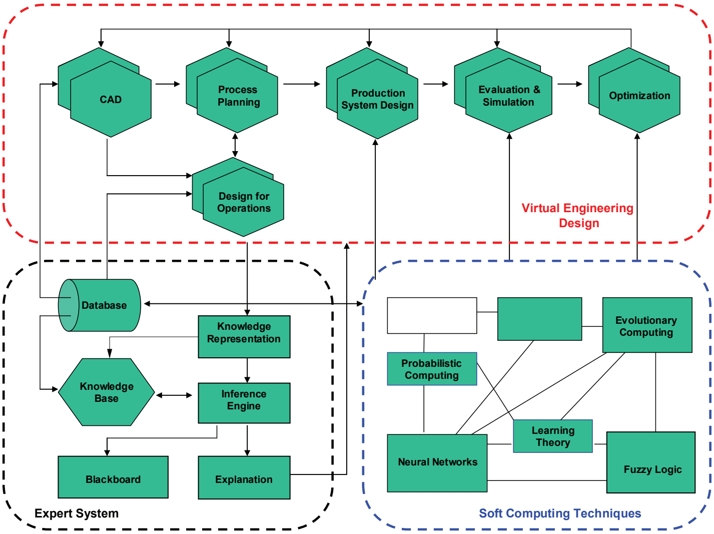
Soft Computing (SC) and hybrid Intelligent Systems (HIS) are rapidly growing in importance and visibility in engineering, in which two or more above individual AI techniques are combined to overcome the limitations of each individual technique, or different intelligent modules are used to solve problems collectively with each solving the parts at which it is best [6,11,4,18,19]. In fact, the SC and HIS techniques have now been applied in a wide variety of real-world complex problems, including planning and scheduling, representation and reasoning, intelligent interface, database systems, process control and fault diagnosis, prediction of economic data, etc. However, there is relatively little work in applying such powerful techniques to complex engineering design problems, especially in virtual product design and prototyping. To overcome the limitations of conventional, individual intelligent design techniques, there is now a growing need in the intelligent design community that complex design problems require hybrid solutions [19,33,37].

Our previous work proposed a neuro-fuzzy hybrid scheme for complex human-machine systems design and simulation [17,28]. Other related work on integrated intelligent systems for engineering design and manufacturing is discussed in [33,19,36]. This paper aims to develop a unified integrated framework for virtual product design and prototyping based on hybrid intelligent systems methodology. In particular, an evolutionary fuzzy neural network model (EFNN) is proposed to integrate individual neural networks, fuzzy logic and genetic algorithms techniques in a unified framework and environment. The "unified" hybrid intelligent approach capitalizes on the advantages of each of these individual intelligent techniques but offsets their disadvantages. This work is not intended to replace the conventional individual design techniques but to complement them and build even more powerful computational models so that good solutions could be attained or a much wider range of design situations can be adequately dealt with.

The remainder of this paper is organized as follows. Section 2 proposes a soft computing integrated intelligent framework for design of complex engineering systems. Section3 proposes an evolutionary fuzzy neural network model (EFNN). Section 4 discusses several approaches to generation of the EFNN model through machine learning. Section 5 outlines the implementation of the EFNN-based hybrid intelligent design system. Sections 6 and 7 present the application of the EFNN model and case study for design and simulation using the developed model and system. Section 8 summarizes and concludes the paper.

2 SOFT COMPUTING INTEGRATED INTELLIGENT DESIGN FRAMEWORK

In this work, a soft computing integrated intelligent framework is proposed for engineering design, as shown in Fig.1, involving an integration of knowledge-based systems, soft computing techniques [27], and virtual engineering design techniques. Soft computing techniques facilitate the use of fuzzy logic, neuro-computing, evolutionary computing and probabilistic computing in combination, leading to the concept of hybrid intelligent systems (HIS).



Blank rectangles in the block stand for other unknown soft computing techniques

Fig.1: Soft computing based integrated intelligent design framework

The concurrent integrated design of products (and processes) entails the integration of the following processes or procedures: product design, process planning, design for operation, task assignment and line balancing, equipment selection, production system design/layout, evaluation and simulation, etc. Virtual prototyping is becoming an emerging technology that allows engineers to visualize multi-dimensional properties of a new product at its design stage without actually producing a prototype [2,8,10]. In virtual engineering design, a machine is virtually assembled; its performance is simulated using a simulation package and tested with new variable inputs. This allows manufacturers to reduce or bypass multiple physical prototype stages and achieve a significantly higher level of efficiency in product design by saving time and costs from the development cycle. The key issue in virtual product design and development is to accurately correlate the simulation output with the input conditions and after that to evaluate or predict the machine's behavior and performance under new conditions.

3 AN EVOLUTIONARY NEURO-FUZZY HYBRID MODEL

3.1 The Neuro-Fuzzy Network Model

The structure of a general fuzzy neural network system consists of components of a traditional fuzzy logic system and the neural network structure [17,10,24]. It is basically a five-layer fuzzy (rule-based) neural network. In accordance with the common neural network notation in [23], a node in any layer of the network has its input. The node performs a certain designated transformation or operation on its input weighted by their respective weight values and generates an output which is a function of its input. The output weighted by weight values will in turn become inputs to the relevant nodes in the immediate next layer according to the node connections in the network structure. The input layer of the FNN has $p+1$ nodes, where p is the number of significant inputs. The extra input is a bias, which has a constant activation value of 1. The bias neurons provide adjustable thresholds for each neuron connected to them. The fuzzification layer is used to build the antecedent of the fuzzy rules. The fuzzy rule is represented as "if x_1 is A_1, …, and x_i is A_i, then y is B" where A_i is a linguistic term. The output of the fuzzification layer characterizes the possible distribution of the antecedent clause "x_i is A_i". The inference layer has h rule nodes. Each rule node represents one fuzzy rule, which describes the characteristics of a sub-set of input-output data. The number of fuzzy rules required depends on the number of appropriate subsets in the input-output space. The inference layer calculates the certainty of each compound proposition "if x_1 is A_1,…, and x_i is A_i". This indicates how well the prerequisites of each fuzzy rule are satisfied. The defuzzification layer performs the rule evaluation. The number of output nodes corresponds to the number of output required in the model.

Fuzzy neural networks (FNN) can be trained using various supervising or self-organizing training methods [23], such as the back propagation method [5], the Kohonen's feature maps method [7], and the conjugate gradient method [9]. The standard back propagation [12,15,11] remains the most widely used supervised training and learning method for neural networks. There are also many other alternative approaches based on logic programming, different or hybrid neural networks and genetic algorithms (GAs) that can be deployed for training and learning FNN's to complement the traditional algorithms. Here, the extended hybrid dual cross-mapping neural network (HDCMNN) [22] and the GAs are used for FNN's training and learning. The extended HDCMNN model is an integration of two individual network models: the above FNN and the Hopfield network (HNN) [22]. These two networks perform different but complementary tasks, i.e., the output weight values of the FNN serve as the energy function of the HNN. The cross mapping is spontaneously implemented through the dynamic changes of the network. Compared with the previous work, this approach has self-modifying and self-learning functions for evolution, within which only one network is needed to be trained for accomplishing the evaluation, rectification/ modification and optimization tasks in the design process. The evolution of FNN by using the GAs will be discussed in Section 3.2 below.

3.2 The Evolutionary Neuro-Fuzzy Hybrid Model

The evolution of the above FNN model can be considered as in two aspects: 1) evolution of the fuzzy knowledge base, i.e., evolution of the fuzzy rule bases, and 2) design evolution of the fuzzy neural network architecture. The key processes of the evolutionary approach to the design of FNN models can be illustrated in Fig.2. When the GA is integrated into FNN to evolve a population of fuzzy variables in fuzzy rule, it requires a pre-specified number of rules, which cannot be met in many real applications. Section 4 will discuss the generation of FNNs.

Evolutionary Fuzzy Knowledge Base

GAs can be used to find or tune knowledge bases (rules) for expert systems, and expert systems can provide heuristics to improve the performance of genetic algorithms. Specifically, GAs can be applied to expert networks to find better membership values or certainty factors, and/or they can be used to optimize the parameters in problem-solving techniques. Thus, GAs may be used for knowledge base synthesizing or rule-finding and tuning for FNN models. In synthesizing fuzzy knowledge-based nets (fuzzy rule bases), three features may be adjusted: the membership functions defined over the input and output variables, the rule base, and the weights connecting rules to the outputs. As a result, there are two direct ways for integrating FNN with GA: 1) the combination of GA and FNN; and 2) the hybridization of GA and FNN.

GAs can be used to find the optimum solutions for the parameters (e.g. design objectives and custom optimization criteria, see Section 6). The best solutions are recorded, and then the FNN is employed to simplify and resolve problem's complexity by substituting existing associative relations with a fuzzy rule system. Redesign is performed by searching the optimum solution under the same criteria but using the simplified fuzzy structure. In the hybridization of GA and FNN, given that a rule is coded as a single chromosome, the rules are then packed end-to-end to form the chromosome. Thus, a GA can be used to evolve a population of chromosomes, and yield a most-fit chromosome (a rule), which represents the relationship between the output and input fuzzy variables in rules. Fixed or static rules representing a priori knowledge are excluded from manipulation by the GA operators (crossover, mutation) but are included in the evaluation of a chromosome's fitness. In a simple GA, the length of the chromosome is fixed and thus the number of rules (R_1, R_2, …, R_n) packed in a chromosome is fixed, as follows,

$$R_1, \ldots, R_i, \ldots, R_n$$
$$|$$
$$110,\ldots,111,\ldots,010$$
$$|$$
$$\text{If } X_1=A_1 \text{ and } X_2=A_2 \text{ and } \ldots \text{ Then } H_1=C_1$$
$$\ldots, \ldots$$

The number of rules must be specified prior to training. A trail and error procedure can be used to estimate the number of rules required by running the GA several times with an arbitrarily large number of rules and examining the number of duplicated rules and cut connections. If the fuzzy knowledge based system is under saturated (i.e., contains disconnected or duplicated rules), then the number of rules will be decreased.

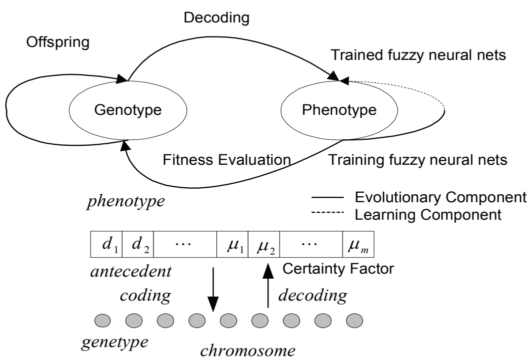


Fig.2: Evolutionary (genetic) design of fuzzy neural networks

Evolutionary Design of Neuro-Fuzzy Networks

With the given sets of constraints, the design and evolution of FNN models for specific applications is a process of trial and error relying mostly on the past experience with similar applications. Moreover, the performance of the neural network models on particular problems is critically dependent, among other things, on the choice of primitives, network architecture (e.g. topology and type), and the algorithms. These factors make the evolutionary design process of the neural network model difficult. In addition, the lack of sound design principles constitutes a major hurdle in the development of large-scale neural network models and systems for a wide variety of practical problems. Motivated by above, it is therefore necessary to investigate efficient design and evolution approaches for the fuzzy neural network models. The design and evolution of FNN architecture is a challenging multi-criteria optimization problem for particular classes of applications (e.g. modeling, planning, and simulation, visual pattern classification, inductive and deductive inference, etc.). Evolutionary (genetic) algorithms offer an attractive and relatively efficient, randomized opportunistic approach to search for near-optimal solutions to the neural network architecture.

4 GENERATION OF THE NEURO-FUZZY NETWORK MODEL

Fuzzy rules in the knowledge base are acquired using the learning approach. The generation of the neuro-fuzzy model can be implemented through many machine learning approaches. In this work, the approaches based on data mining (e.g., the cluster-based partition), logic programming (e.g., the inductive logic programming ILP [34]), and hybrid symbolic-connectionist network (fsc-Net) [35]) are deployed for generation of the FNN.

4.1 Cluster-based Partition

To obtain the significant recurrent inputs in the discrete time models, the extraneous inputs from the time history of the input/output data have to be eliminated. There are many methods to partition the input-output space into subsets [30], like the genetic algorithm based partition [31], the grid partition, and the cluster partition [32]. A simplified cluster-based partition approach based on the "effect" of input-output data is used in this work to identify the significant inputs. The concept of "effectiveness" is based on the fuzzy logic curves proposed in [29]. From the variation of the "effect" of an input variable, the number of partitions required for mapping could be estimated. The identified significant inputs determine the number of input neurons in the input layer of the FNN.

4.2 Inductive Logic Programming (ILP)

Inductive logic programming (ILP) is a well known machine learning technique in learning concepts from relational data [34]. Nevertheless, ILP systems are not robust enough to noisy or unseen data in real world domains. Furthermore, in multiclass problems, if the example is not matched with any learned rules, it cannot be classified. To alleviate the above restriction, in this paper, we used a novel hybrid learning method called the first-order logical fuzzy neural network (FOL-FNN) to enable fuzzy neural networks to handle first-order logic programs directly. The FOL-FNN method is based on feed forward neural networks and integrates inductive learning from examples and background knowledge. In addition, a multiple-instance learning (MIL) method is used to determine the appropriate variable substitution in FOL-FNN learning.

4.3 Hybrid Symbolic-Connectionist Network

Fuzzy rules in the knowledge base are acquired using the learning approach with the fuzzy symbolic-connectionist network (fsc-Net) [35]. fsc-Net is a hybrid symbolic -connectionist network that utilizes fuzzy logic as its means to perform uncertainty management. The learning algorithm is a mechanism that supports automatic construction of the network topology. The connectionist network encodes the learned knowledge. By incorporating symbolic structures into the network itself, it is possible to represent both structured as well as unstructured variables and rules. Rules can not only be added as domain specific knowledge, but also can be extracted after learning or refined during learning. Finally, fsc-Net supports the direct incorporation of fuzzy variable membership functions through the user, or by automatic learning of the appropriate membership functions for any given inputs.

5 SYSTEM IMPLEMENTATION

The proof-of-concept implementation of the EFNN-based hybrid intelligent design system (EFNN-HIDS) is developed. EFNN-HIDS is basically a combination of various independent and self-contained intelligent modules or subsystems coordinated by a control mechanism to perform all the sub-tasks in the design process. Each of the intelligent modules itself is a HIS of individual AI techniques, working as a unified system to perform the sub-task for which it is designed. It allows the user to concentrate on the design problem without having to worry about the selection and fine-tuning of intelligent (optimization) algorithms. The current EFNN-HIDS implementation as shown in Fig.3 integrates the following sub-systems such as: an FNN modeling and mapping system, a GA optimization system, a GA-FNN interface, a fuzzy design tool, a fuzzy rule acquisition system, and a CAD system. Other Matlab tools are also integrated. The fuzzy rule acquisition system is actually a knowledge learning system. The self-developed (both standalone and web-enabled) RAPID Assembly system [16,17] is used as the CAD system for test purpose. It was originally developed for virtual assembly prototyping and planning with modules such as geometric modeling, assembly design, assembly sequence planning, assembly system construction, simulation and evaluation, and ergonomic database (e.g. anthropometric data). It is noted that commercial CAD systems with appropriate APIs can also be supported.

We are currently implementing an ontology-based open engineering platform (SWOEP) that can deliver a new sustainable approach for ICT support in engineering of complex products [39,40]. The platform will be accomplished by a tightly integrated combination of the above EFNN model and the Semantic Web (SW) technologies. This combination allows smart manipulation of semantic data for optimized product engineering processes that involve multidisciplinary and dynamic development environments. SWOEP will enable product developers to specify their products parametrically and evolutionarily. The major result of this is a product ontology that can represent a parametric product, an instance of which can be customized (optimized) for a specific user (client/buyer), over the web. The SWOEP platform will provide a generic tool that can be customized and validated through different application scenarios or business contexts.

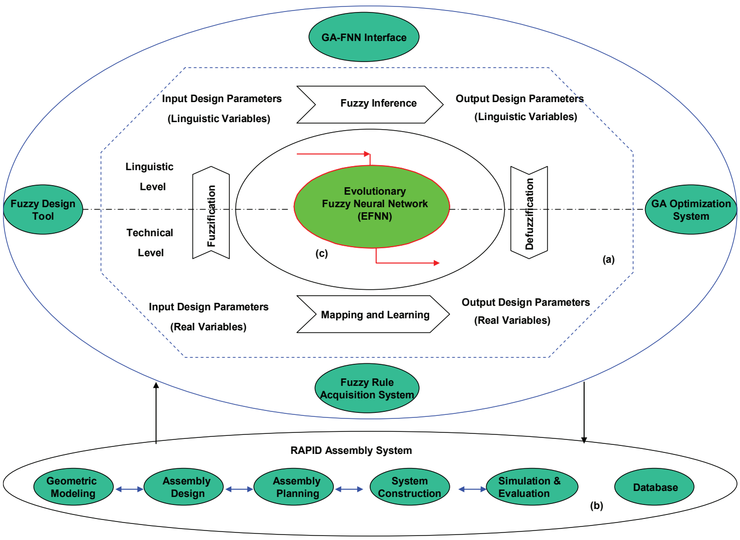


Fig.3: EFNN-HIDS design system implementation environment

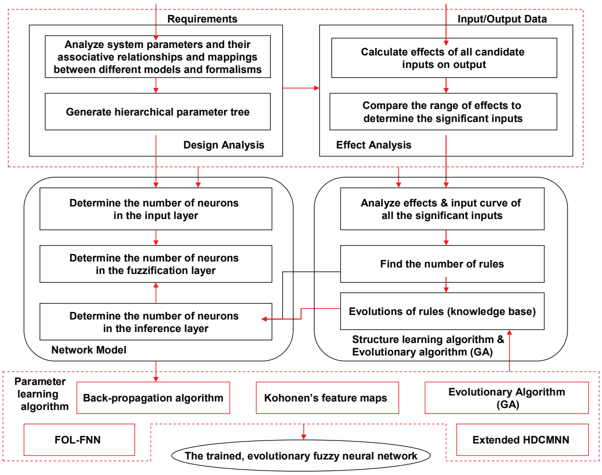


Fig.4: The EFNN approach to engineering design

6 APPLICATIONS OF THE EVOLUTIONARY NEURO-FUZZY MODEL

6.1 Overview

A design problem could be defined in terms of design parameters and their associative relationships and mappings between different models and formalisms [25]. In most cases, the available design knowledge may be vague and expressed with formalisms other than analytical relations. Fig.4 shows the steps for building the EFNN model for design. The process of application involves:

(1) Design analysis and/or effect analysis, including customer requirements, constraints, effects of candidate inputs on outputs, etc.
(2) Resolving the significant inputs through an analysis of their effects with respect to the output and/or generating hierarchical (design) parameters tree.
(3) The number of fuzzy rules along with the FNN structure can then be determined based on the partitioning of the space for the significant inputs or the generated hierarchical parameter tree.
(4) Once the FNN structure has been determined, it can be trained and evolved using the training and learning algorithms (e.g., genetic algorithms, back propagation algorithm) in Sections 3 and 4.

6.2 Application Scenarios of the EFNN Model in Design

The EFNN model can be applied to any design stages, such as creation of design concepts, design selection and evaluation, optimization, and simulation. The following three application scenarios are given.

Scenario 1

Design problem can generally be defined as: given a set of (fuzzy) functional requirements (FR) to generate and select a family of design solutions (DS) that can satisfy the input customer requirements and design constraints (DC). The relationships between requirements, constraints, and final solutions can be represented as fuzzy matrices M (FR×DS), M (DC×DS), and M (DS×DS) in the fuzzy model. The EFNN model is used to solve the design problem formulated above, which means that implement mappings, e.g., the mapping from a fuzzy set of requirements to a crisp set of design alternatives [1], i.e., M (FR×DS), see Section 7. After analyzing various requirements and possible design solutions (e.g. tables) and discussing with marketing groups, fuzzy rules regarding the characteristics of design can be identified. The fuzzy knowledge base in the fuzzy design model is composed by two components: a linguistic terms base and a fuzzy design rule base. The EFNN model supports fuzzy models with N multiple inputs and M multiple outputs (MIMO). The fuzzy design knowledge base expressed by k (k=1,2,…K) MIMO-type heuristic fuzzy rules. The global design relations aggregate all fuzzy rules, which combine fuzzification and inference stages.

Scenario 2

The EFNN model addresses design as a parametric optimization problem, in which the optimal design solution is extracted by varying a set of design parameters. Some of these design parameters are only used for defining other design parameters through associative relations, constraints and rules that are critical for the design performance (performance variables). Specifically, the FNN is used to substitute existing associative relations, constraints, and rules in design; GA is used to find the optimum design solution according to the design objectives and evaluation criteria. The GA results in an optimal solution and a set of 'elite' values for variable primary and dependent design parameters or performance variables [26]. This recorded set of values is used for training the FNN that associates the primary performance variables with dependent performance variables without any intermediate design parameters or associations. The initial design problem is considered to be evolved into a simplified fuzzy digraph with two-levels. Formally, these relations can be represented by fuzzy matrices. Then, an "effect" variable is used to identify the significant inputs from the input-output data [29, 14]. The variations of the "effects" of the significant inputs are used to determine the number of fuzzy rules and the structure of the network.

Scenario 3

The EFNN model is used to evaluate and predict product performance for virtual prototyping. Conventional methods like multiple regression models suffer from several deficiencies and limitations in expressing ability (an explicit model cannot always express the relationships between the outputs and the input factors) and accuracy (when the data involved exhibit irregular pattern), which make them inadequate for virtual product design and prototyping in today's manufacturing requirements [10,24]. It is, therefore, desirable to have a new evaluating or predicting method that can overcome these deficiencies and limitations. In this case, the EFNN model implements fuzzy input-output relations, constraints, and rules for evaluation and prediction in design.

7 CASE STUDY

Three case studies corresponding to three application scenarios above have been carried out in this study. Due to space limits, only one case study on workstation table customization design is given to verify the EFNN model and illustrate the EFNN-HIDS system. This case is corresponding to Scenarios 1 & 2 and it is inspired from [1]. Details are discussed below. The essence of mass customization (MC) is that customers' requirements are precisely satisfied without increasing costs, regardless of how unique these requirements may be. The design for mass customization problem can be defined as: given a set of fuzzy (functional) requirements to generate and select a family of design solutions that can satisfy the design requirements and constraints. More specifically, it is assumed that the design has x generic functional requirements (FR), y design constraints (DC) and z design solutions (DS) generated. Each design solution must satisfy a certain set of functional requirements and design constraints. The relationships between requirements, constraints, and final solutions are represented as the following fuzzy matrices: $M_{x\times z}$(FR×DS), $M_{y\times z}$ (DC×DS), and $M_{z\times z}$(DS×DS).

To design various classes of tables (i.e., variants in a table family) for mass customization, the developed EFNN model and system are used to map a fuzzy set of requirements to a crisp set of table design alternatives. After analyzing various tables and discussing with marketing groups, an EFNN model can be generated using the approaches and techniques discussed in Section 4. Fig.5 is the diagram to illustrate design rules involved in the design. The number of membership function nodes is set for each input variable after studying the distribution patterns of the data samples. After training and evolution, the EFNN model can evaluate and select a customized workstation table based on fuzzy customer requirements. In the defined EFNN model, the input layer has 17 neurons, one neuron for every requirement. Each degree of membership is assigned to a separated neuron. The output layer is designed to have 11 neurons, one for every class of table. For example, the design for a customized table may be stated as: Design a very comfortable table that can be used more or less in operation and should cost about \$100. Thus, the input functional requirements and constraints can be derived as: FR={fr_1}={table must be very comfortable}, DC={dc_1, dc_2}, dc_1=use in operation; and dc_2=cost about \$100. The following sample rule can be derived:

Rule X: IF the desired table must be highly adjustable
& cost is an important consideration
& used mostly in a workstation
& used for typing
THEN the table is an office table.

Based on this rule, the membership function values can be derived: fr_1=highly adjustable, μ_{fr_1} =0.9; dc_1=the cost is an important consideration, μ_{dc_1} =0.8; dc_2=mostly to be used in workstation, μ_{dc_2} =1.0; dc_3=use for typing, μ_{dc_3} =0.5. Thus, in the EFNN model, the functional requirements and design constraints for the workstation table are represented as: INPUT= (0.9, 1.0, 0, 0, 0, 0, 1.0, 0, 0, 0, 0.8, 1.0, 0.5, 0, 0, 0, 1.0). The first value corresponds to f_{r1} (highly adjustable), the second to f_{r2} (ability to move the table), and so on. The design output for the workstation table is represented as: OUTPUT= (0.3, 1, 0, 0, 0.6, 0, 0, 0, 0, 0, 0). To experiment with the EFNN-HIDS system, 10 fuzzy rules, one rule for each class of table, has been created, and 10 design alternatives have been generated. From the test results, the system was capable of successfully generating 9 correct solutions. The overall system accuracy rate is better than 90%. Fig.6 shows the developed HIDS-FNN design system for conceptual workstation table customization design.

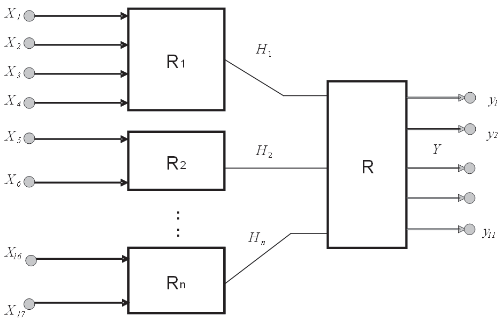


Fig.5: Design and control rules

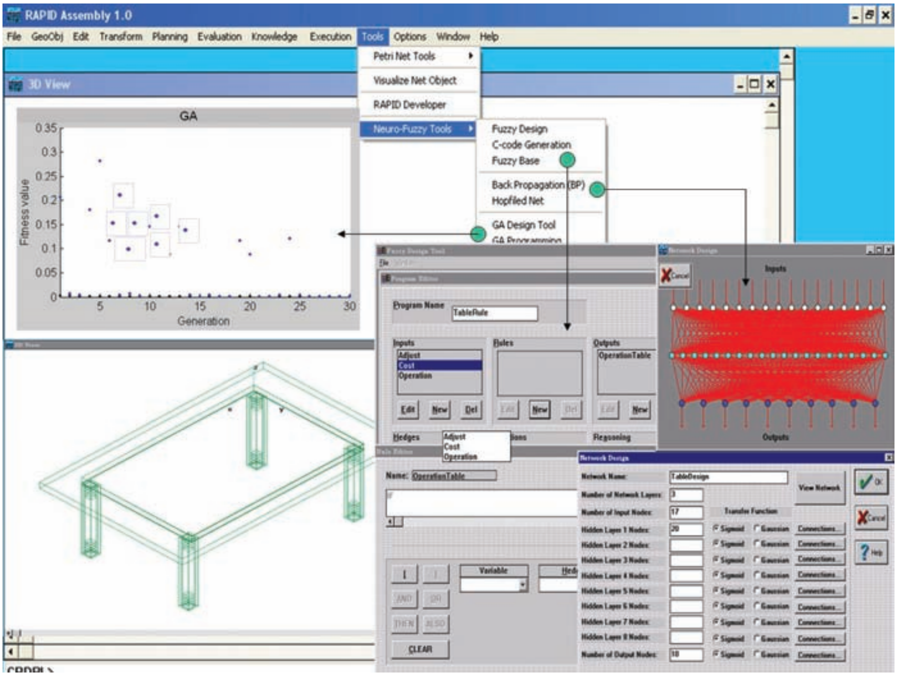


Fig.6: HIDS-EFNN design system used for workstation table customization design

8 SUMMARY AND CONCLUSIONS

This paper developed a soft computing integrated intelligent framework for engineering design, virtual prototyping and simulation. Based on the hybrid intelligent framework, a combination of CAD geometry, fuzzy set and logic on requirements, genetic algorithms on search, and neural network design rules is used to develop an EFNN-HIDS system to create design concepts for different design scenarios. The developed EFNN model and system can be used as a toolkit for virtual, predictive product design and realization. The contributions of this work are summarized as: 1) a unified soft computing integrated intelligent framework and environment for virtual engineering design and simulation, 2) a hybrid integrated intelligent model for supporting computational design and simulation, 3) the evolutionary neuro-fuzzy modeling approach and its applications in virtual product design and prototyping, and 4) the approaches to generation and learning of the evolutionary neuron-fuzzy hybrid design model in which fuzzy design rules delivered by the trained FNN are extracted from elite design solutions; they can be further calibrated rationally in re-design; they can also be used in a collaborative framework among designers with aggregation of existing fuzzy sets under a specific aggregation strategy. In conclusion, the soft computing integrated intelligent design framework presented in this paper can provide a better solution to the problems of virtual product design and prototyping. The developed EFNN model and system are generic and flexible enough to be used in a variety of other applications such as control, machine prognosis and diagnosis. We are currently implementing an ontology-based open engineering platform (SWOEP) that can deliver a new sustainable approach for ICT support in engineering of complex products.

REFERENCES

[1] Bahrami, A. and Dagli, C.H., From Fuzzy Input Requirements to Crisp Design, International Journal of Advanced Manufacturing Technology, 8:52-60, 1993

[2] Bateman, R.E., Bowden, R.G., Gogg, T.J., Harrell, C.R. and Mott, J.R.A., System Improvement Using Simulation, Fifth Edition, Promodel Corporation, 1997
[3] Dagli, C.H., Artificial Neural Networks for Intelligent Manufacturing, Chapman & Hall, 1994
[4] Goonatilake, S. and Khebbal, S., Intelligent Hybrid Systems, John-Wiley & Sons, 1995
[5] Horikawa, S., Furuhasshi, T. and Uchikawa, Y., On Fuzzy Modeling Using Fuzzy Neural Networks with the Back-propagation Algorithms, IEEE Transactions on Neural Networks, 3(5):801-806, 1992
[6] Jang, J.S.R., ANFIS: Adaptive-Network-Based Fuzzy Inference System, IEEE Transactions on System Man Cybernetics, 23: 665–685, 1993
[7] Kohonen, T., Self-organization and Associative Memory, Springer-Verlag, Berlin, 1988
[8] Law, A.M. and Kelton, W.D., Simulation Modeling & Analysis, Second Edition, McGraw-Hill International Editions, 1991
[9] Leonard, J.A. and Kramer, M.A., Improvement of the Back-Propagation Algorithm for Training Neural Networks, Computer Chemical Engineering, 14:337-341, 1991
[10] Li, X, Tay, A., and Ang, C.L., A Fuzzy Neural Network for Virtual Equipment Design, CD-ROM Proceedings of Intelligent Automation in New Millennium, IEEE Sixth International Conference on Control, Automation, Robotics and Vision, December 5-8, 2000, Singapore, Paper No. 443.
[11] Medesker, L.R., Hybrid Intelligent Systems, Kluwer Academic Publishers, 1995
[12] Refenes, A.-P., Neural Networks in the Capital Markets, John Wiley & Sons, Chichester, 1995
[13] Wang, J. and Takefuji, Y., Neural Networks in Design and Manufacturing, World Scientific, Singapore, 1993
[14] Xiang, W., Fok, S.C., and Yap, F.F., A Fuzzy Neural Network Approach to Model Hydraulic Component from Input–Output Data, International Journal of Fluid Power, 2 (1): 37–47, 2001
[15] Yager RR, Zadeh LA (eds), An Introduction to Fuzzy Logic Applications in Intelligent Systems, Kluwer Academic Publishers, 1991
[16] Zha, X.F. and Li, LL, AI-supported Web-based Virtual Prototyping and Planning for Assembly, Proceedings of FAIM 2004, Canada
[17] Zha, X.F., Soft Computing Framework for Intelligent Human–Machine System Design, Simulation and Optimization, Soft Computing, 7:184 -198, 2003
[18] Zha, X.F. A Hybrid Cross-Mapping Neural Network Model for Computational Intelligent Design, International Journal of Knowledge-based and Intelligent Engineering Systems, 8(1): 17-26, 2004
[19] Zha, X.F., Artificial Intelligence and Integrated Intelligent Systems in Product Design and Development, Intelligent Knowledge-based Systems: Business and Technology in New Millennium, Vol. IV: Intelligent Systems, Chapter 1, Cornelius T. Leondes (ed), Kluwer Academic Publishers, USA, 2004
[20] Zha, X.F., Soft Computing in Engineering Design: A Fuzzy Neural Network Model for Virtual Product Design, Advances in Soft Computing Series, Springer Verlag, Germany, 2006
[21] Zarefar, H., and Goulding, J.R., Neural Networks in Design of Products: A Case Study, Intelligent Design and Manufacturing, Kusiak A (ed.), pp. 179–201, John Wiley & Sons, Inc., 1992
[22] Zha, X.F., Soft Computing in Engineering Design: A Hybrid Dual Cross-Mapping Neural Networks, Neural Computing and Its Applications, 14 (3):176-188, 2005
[23] Yu, W. and Li, X.O, Fuzzy Identification using Fuzzy Neural Networks with Stable Learning Algorithms, IEEE Transactions on Fuzzy Systems, 12(3), 2004
[24] Li, X., Ang, C.L., and Gay, R., An Intelligent Hybrid System for Business Forecasting, Intelligent Knowledge-based Systems: Business and Technology in New Millennium, Vol. IV: Intelligent Systems, Chapter 4, Cornelius T. Leondes (ed), Kluwer Academic Publishers, USA, 2005
[25] Suh, N.P., The Principle of Design, New York, Oxford University Press, 1990
[26] Saridakis, K.M., Dentsoras A.J., Evolutionary Neuro-Fuzzy Modeling in Parametric Design, The First I*PROMS Virtual International Conference on Intelligent Production Machines and Systems, IPROMS 2005, 4-15 July 2005
[27] Zadeh, L. A., Fuzzy Logic, Neural Networks and Soft Computing, Communication of ACM, 37(3): 7784, 1994

[28] Zha, X. F., Lim, S. Y. E., Intelligent Design and Planning of Manual Assembly Workstation: A Neuro-Fuzzy Approach, Computers & Industrial Engineering, 44: 611–632, 2003
[29] Lin, Y., Cunnihingham III, G. A. and Coggeshall, S. V., Using Fuzzy Partitions to Create Fuzzy Systems from Input-Output Data and Set the Initial Weights in a Fuzzy Neural Network, IEEE Transactions on Fuzzy systems, 5(4): 614-621, 1997
[30] Jung, C.F., and Lin, C.T., An On-Line Self-Constructing Neural Fuzzy Inference Network and Its Applications, IEEE Transactions on Fuzzy Systems, 6(1):12-32, 1998
[31] Ruspini, E.H., Recent Development in Fuzzy Clustering, Fuzzy Set and Possibility Theory, New York: North Holland, pp.113-147, 1982
[32] Sun, C.T., and Jang, J.S., A Neuro-Fuzzy Classifier and Its Applications, Proc. IEEE Int. Conf. Fuzzy System, San Francisco, CA, March 1, pp.94-98, 1993
[33] Zha, X.F., Howlett, R.J. (ed.), Integrated Intelligent Systems for Engineering Design, IOS Press, 2006
[34] Lerdlamnaochai, T.; Kijsirikul, B, First-order Logical Neural Networks, Fourth International Conference on Hybrid Intelligent Systems, 5-8 Dec. 2004, Page(s): 192 – 197
[35] FSC-Net, Fuzzy Symbolic-Connectionist Network, Universal Problem Solvers, Inc., 1995
[36] Zha, X.F., Integrated and Hybrid Intelligent Systems in Product Design and Development, Guest Editorial: International Journal of Knowledge-based and Intelligent Engineering Systems, 9(2),2005
[37] Saridakis, K.M., Dentsoras A.J.,Soft Computing in Engineering Design: A Review, Advanced Engineering Informatics, 22(2): 202-221, 2008
[38] Chen, C.H., Occea, L.G., Soft Computing for Concurrent Product Design Evaluation under a Blackboard Environment, Proceedings of the Third International Conference on Engineering Design and Automation (EDA '99), Vancouver, Canada, 1999
[39] Bock,C. and Zha, X.F., Ontological Product Modeling for Collaborative Design, NIST, 2007
[40] Liang, V.C., Bock, C. and Zha, X.F., An Ontological Modeling Platform, NIST, 2008

Contact: Dr. Xuan F. Zha
President and Chief Scientist
Extension Systems International, LLC
Gaithersburg, MD 20878
USA
Tel: 1-240-3867186
Fax:1-301-2169704
E-mail: xfzha@ieee.org
URL: http://www.featronix.com/