An Introductory Course on Computational Artificial Intelligence Techniques for Engineering Students

Mario G. Perhinschi

***Abstract*—This paper brings forth the design of an introductory course on computational artificial intelligence techniques for graduate and undergraduate engineering students. The course is aimed at providing students outside the computer science and computer engineering curricula with an affordable and practical introduction to genetic algorithms, fuzzy logic, and artificial neutral networks. At the end of the course, the students are expected to be able to solve engineering problems using the three covered methodologies. The general strategy for designing the course aims at three main targets: readiness, practicality, and applicability. Active and experiential learning methodologies are implemented primarily through three team projects addressing each of the three techniques investigated. Students are directly involved in the selection of project topics to increase their motivation and participation and effectively close the gap between theory and practice in the context of applications that are relevant for student academic and career interests. The course has been received with great interest by students and produced positive educational impact by allowing dissemination of powerful techniques to a broader engineering audience.**

***Index Terms*—Computational Artificial Intelligence Techniques, Education, Fuzzy Logic, Genetic Algorithms, Neural Networks**

I. Introduction

While there are no commonly accepted definitions for intelligence, artificial intelligence (AI), or computational AI, the latter syntagm will be used here to refer to algorithms and methodologies inspired by the biological world and used to solve complex real-world problems for which "traditional" methods do not exist, fail, or are unpractical [1,2].

Over the past few decades, a variety of computational AI techniques have been developed and implemented exhibiting an extreme diversity in terms of source of inspiration, conceptual background, mathematical representation and methodology, and applicability. A nonexhaustive list, starting with the top three most popular categories, includes: evolutionary optimization [3,4], fuzzy logic [5,6], artificial neural networks [7,8], artificial immune systems, swarm intelligence, expert systems, etc.

Computational AI techniques have been extensively used not only in all engineering areas [9-11], but also in medicine [12,13], finances [14], business management [15], environmental sciences [16,17], and the list could continue. This widespread applicability and potential justify the significant early attention given to the area within academia. Today, computational AI techniques are well incorporated into computer science (CS) curricula at both the undergraduate and graduate levels [18,19]. There are general courses required or elective for the computer scientist or engineer, but there are also dedicated programs offering focus on AI and computational AI [20].

Computational AI topics and techniques are broad and interdisciplinary, lack a unified methodology, and involve a large diversity of knowledge, skills, and abilities. As a consequence, the access of engineering students to the field through typical CS courses is most of the time unaffordable and impractical because they are rightly detailed, broad, and specific. It is the author's experience that engineers can implement, use, and benefit from AI techniques, if provided with a critical, yet limited background. It is also important that a direct connection is clearly established to real-life problems that engineering students are familiar with through other courses or research.

Attempts to address the issues related to introducing AI techniques to broader audiences exist [21-24]; however, educational tools to effectively facilitate the adequate dissemination of AI techniques outside CS in a practical and affordable manner are still needed.

In this paper, the design of a course is presented aimed at introducing engineering students to three most popular and powerful computational AI techniques: genetic algorithms (GA), fuzzy logic (FL), and artificial neural networks (ANN). The general course design strategy is to emphasize practicality from an engineering point of view and rely on active [25] and experiential [26] learning methodology. The course objectives and learning outcomes are formulated and supported closely by the course topics, content, and assignments. Special attention is given to the three team projects addressing each of the three main areas covered. The projects are the main tool for active and experiential learning providing the framework for concept clarification,

This paper was submitted for review on August 18th 2017.

M. G. Perhisnchi is a professor with the Department of Mechanical and Aerospace Engineering, West Virginia University, Morgantown, WV, USA (e-mail: Mario.Perhinschi@mail.wvu.edu).

topic synthesis and analysis, and bridging the gap between theory and practice. The projects also represent an instrument for increasing student motivation and interest and exercising sound team work. Finally, the paper presents a brief discussion of the course impact and student feedback.

II. General Strategy for Designing the Course

Three main vectors drive the general strategy for designing the course: readiness, practicality, and applicability.

The three computational AI techniques (GA, FL, and ANN) are introduced from a direct and practical engineering point of view. Therefore, the course attempts to provide a set of ready-to-use tools that are necessary for the successful application of the targeted techniques.

Theoretical aspects and mathematical background are tentatively kept at a basic level with the course focusing on practical implementations, examples, and hands-on assignments. This strategy not only responds to time constraints, but also makes the course accessible, attractive, and beneficial to undergraduate seniors.

Students are involved directly in the selection and formulation of project assignments, such that they become aware of how computational AI techniques can solve the problems they come in contact with. Students are encouraged to choose project topics addressing interesting problems that they have encountered during other courses or their research. This approach enhances active and experiential learning and greatly increases student motivation and effectiveness. It provides them with first-hand opportunity to appreciate the applicability and utility of the AI techniques and stimulates their curiosity and creativity.

While targeting primarily graduate students in the area of dynamic systems, modeling, simulation, and control, the course may constitute a valuable technical elective for any engineering undergraduate program.

Several textbooks are recommended for the different chapters [3-10], but not required. The instructor provides comprehensive handouts with relevant course material.

III. Course Objectives and Learning Outcomes

Consistent with the main goal of the course of facilitating the dissemination of practical use of GA, FL, and ANN, the objectives of the course are the following:

- Introducing students to the basic concepts and main aspects of artificial intelligence techniques, in particular: genetic algorithms, fuzzy logic, and artificial neural networks;
- Introducing students to the structure and development of genetic algorithms;
- Introducing students to fuzzy logic-based algorithms for modeling, simulation, and control of complex dynamic systems;
- Introducing students to the design and development of artificial neural networks for function estimation and approximation;
- Providing opportunity for developing skills and ability to use computational AI techniques for solving practical engineering problems;
- Providing new perspectives on engineering design and training students for team work and independent study and research.

At the end of the course, the students are expected to be able to:

- Identify and formulate engineering problems that require non-classic solution techniques;
- Describe the general structure and list the components of a genetic algorithm;
- Design and implement in Matlab® a genetic algorithm for the optimization of design parameters for a complex engineering application;
- Discuss the differences between fuzzy logic and classical (binary) logic;
- Design and implement in Matlab®/Simulink® a fuzzy logic-based controller for a dynamic system;
- Describe the general structure of an artificial neuron and the main characteristics of other components of an artificial neural network;
- Design and implement in Matlab® a neural network-based estimator for a multi-variable non-linear function applicable to modeling or control of a dynamic system, or for a classification problem

IV. Course Outline and Content

A. *Course Outline*

The course is structured in four chapters including a general introduction and one chapter dedicated to each of the three computational AI techniques addressed. At West Virginia University (WVU), the course extends over 15 weeks with the typical 3-hour weekly classroom contact. Additional time must be scheduled as needed for brain storming sessions for project topic assignments and further support. Usually, 2-3 hours per team and per project are sufficient. The topics discussed during the lecture part of the course are presented next. A tentative weekly schedule of topics covered and assignment due dates is listed in Table I.

Chapter 1: Introduction – What are artificial intelligence techniques and why do we need them to solve engineering problems? (1 week)

1.1. Definitions, main concepts, theories
1.2. Areas of application of AI techniques
1.3. Taxonomy of AI techniques

Chapter 2: Genetic Algorithms (or Evolutionary Algorithms) (5 weeks)

2.1. Definitions, concepts, and areas of application
2.2. General structure of a genetic algorithm
2.3. Main components of a genetic algorithm (genetic representation, individual and population, evaluation function, genetic operators)

2.4. Example application: Parameter optimization for a complex engineering design problem
2.5. Discussion of the team project assignment #1: Parameter optimization for a complex engineering design problem (alternatives are considered to match the specific background and interests of the students)

Chapter 3: Fuzzy Logic (5 weeks)
3.1. Classic binary-valued logic versus fuzzy logic
3.2. Definitions, concepts, and areas of application of fuzzy logic and fuzzy sets
3.3. General structure of a fuzzy logic-based controller
3.4. Design of main components of a fuzzy logic-based controller (linguistic variables and values, membership functions, inference rules)
3.5. Example application: Trajectory tracking command for an autonomous vehicle
3.6. Discussion of the team project assignment #2: Control or modeling of a complex system/process (alternatives are considered to match the specific background and interests of the students)

Chapter 4: Artificial Neural Networks (4 weeks)
4.1. Biological neuron versus artificial neuron
4.2. Definitions, concepts, and areas of application of artificial neural networks
4.3. Artificial neural network topology
4.4. Learning process
4.5. Feedforward and feedback artificial neural networks
4.6. Design of an artificial neural network
4.7. Example applications: Pattern recognition and adaptive control
4.8. Discussion of the team project assignment #3: Approximation of a non-linear function using artificial neural networks (alternatives are considered to match the specific background and interests of the students)

TABLE I
RECOMMENDED COURSE SCHEDULE

Week	**Material Covered**	**Due Dates**	**Notes**
Week #1	Chapter 1		
Week #2	Chapter 2, Sections 2.1 and 2.2	Quiz #1	Discussion of team project assignment #1
Week #3	Chapter 2, Section 2.3	Quiz #2	Discussion of team project assignment #1
Week #4	Chapter 2, Section 2.3		
Week #5	Chapter 2, Sections 2.4 and 2.5	Quiz #3	
Week #6	Discussion of team project #1 results and student presentations	Project #1 report due	
Week #7	Chapter 3, Sections 3.1 and 3.2		Discussion of team project assignment #2
Week #8	Chapter 3, Section 3.3	Quiz #4	Discussion of team project assignment #2
Week #9	Chapter 3, Section 3.4		
Week #10	Chapter 3, Section 3.5		
Week #11	Discussion of team project #2 results and student presentations	Quiz #5 Project #2 report due	
Week #12	Chapter 4, Sections 4.1 – 4.4		Discussion of team project assignment #3
Week #13	Chapter 4, Sections 4.5 – 4.6	Quiz #6	Discussion of team project assignment #2
Week #14	Chapter 4, Sections 4.6 and 4.7		
Week #15	Discussion of team project #3 results and student presentations	Project #3 report due	
Week #16		Final Exam	Optional based on performance on quizzes

B. Genetic Algorithms (GA)

GAs are a class of computational AI techniques developed for parameter optimization. Many important problems in sciences, engineering, and economics can be formulated as parameter optimization problems. GAs iteratively search the solution space relying on analogies to natural biological processes [27,28]. They simulate the evolution of species and individual selection based on Darwin's theories to direct the search towards a global optimum. GAs work simultaneously on a set (population) of potential solutions (individuals) to the problem at hand. A set of design requirements and constraint (DRC) must be defined. They may be expressed mathematically, logically (binary or fuzzy), or in a descriptive manner and can be totally independent from one another. This capability is one of the strong points of GAs. This set of DRC plays the role of the environment. Based on the "survival of the fittest" principle, the degree to which solutions meet the performance requirements and constraints is evaluated and used to select "surviving" individuals that will "reproduce" and generate a new population. Individuals will now undergo alterations similar to the natural genetic mutation and crossover. The next iteration starts with this new population (new set of possible solutions). The process continues till there is no more significant increase in the performance of the best solution or a pre-set maximum number of iterations is reached.

After introducing some genetic nomenclature and terminology, the mechanisms of genetic encoding and species evolution are briefly described in the course. The main components of the GA and their interactions are then presented in detail with numerical examples and exercises interactively solved in the classroom. The main components of the GA (Fig. 1) are listed next with the specific algorithms covered mentioned in parentheses:

- representation of potential solutions referred to as chromosome (binary and real);
- strategy regarding the evolution of individuals and the population (fixed or variable individual and population size);

- procedure to generate the initial set of solutions or initial population (random generation or use of prior knowledge);
- evaluation or fitness function is a metric that allows each individual to be quantitatively evaluated with respect to the desired performance criteria (normalization, scaling, and weighting);
- selection and reproduction procedure, which means a way to generate the next step "better" set of solutions (roulette-wheel selection, tournament selection, rank selection, elitist strategy);
- genetic operators to produce changes in the new generation thus introducing new solutions to the search algorithm (binary and real mutation, single point, multiple point, and uniform crossover);
- convergence criterion to assess reaching optimality (solution improvement rate versus hard limit of generation number);
- set of specific GA design parameters (number of individuals in the population, probability of application of genetic operators, parameters of the evaluation function, etc.).

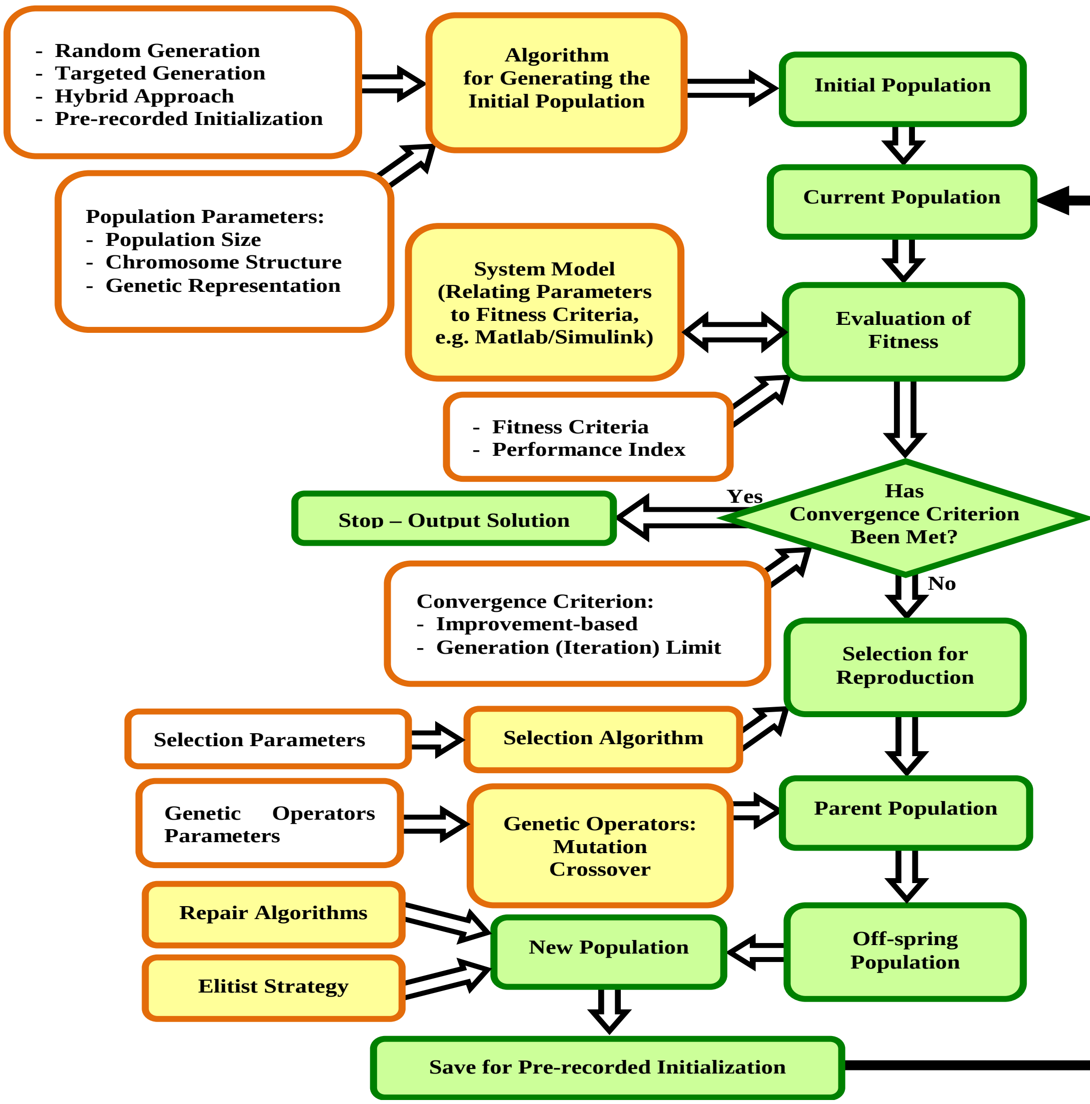

Fig. 1. General flowchart of a genetic algorithm

Three applications from the literature are presented and discussed in class [29-31]. To illustrate how theoretical aspects are approached for GA development and analysis, the Schema Theorem [27] is briefly introduced.

C. Fuzzy Logic (FL)

Classic or binary logic admits for any statement only two alternatives for its truth value, “true” or “false” (typically associated to 1 and 0), as dictated by the law of excluded middle, to only mention one of several equivalent formulations. FL admits instead continuous values in the interval [0, 1] [32]. Students are first introduced to examples that illustrate how fuzzy logic is a necessary complement to classic logic for representing certain aspects of reality. Then, the definition and representation of fuzzy sets and associated operations are presented. The next step consists of discussion on how human knowledge, expertise, and reasoning are formulated through language in terms of IF-THEN rules. FL can then be used to convert such expert databases into algorithms for control, decision making, classification, detection, modeling, and other purposes.

Within the course, the targeted application is system feedback control. However, the students are assured that general concepts and algorithm strategy apply to other areas and they are encouraged to explore this potential in their team project. The main components of a FL-based controller (Fig. 2) are listed next with the specific elements and algorithms covered, mentioned in parentheses:

- input interface or fuzzification module (linguistic variables, linguistic values, membership functions, fuzzy input);
- inference engine or fuzzy command generation based on IF-THEN rules (IF-THEN rule formulation, generalized Modus Ponens, inference rule matrix, individual rule firing, fuzzy command);
- output interface or defuzzification module (crisp command, clipped or scaled fuzzy set approach, defuzzification methods: Center-of-Sum, Height-at-Low-Value, Height-at-Peak-Value, Height-at-Highest-Value, First-of-Maxima, Middle-of-Maxima, Last-of-Maxima).

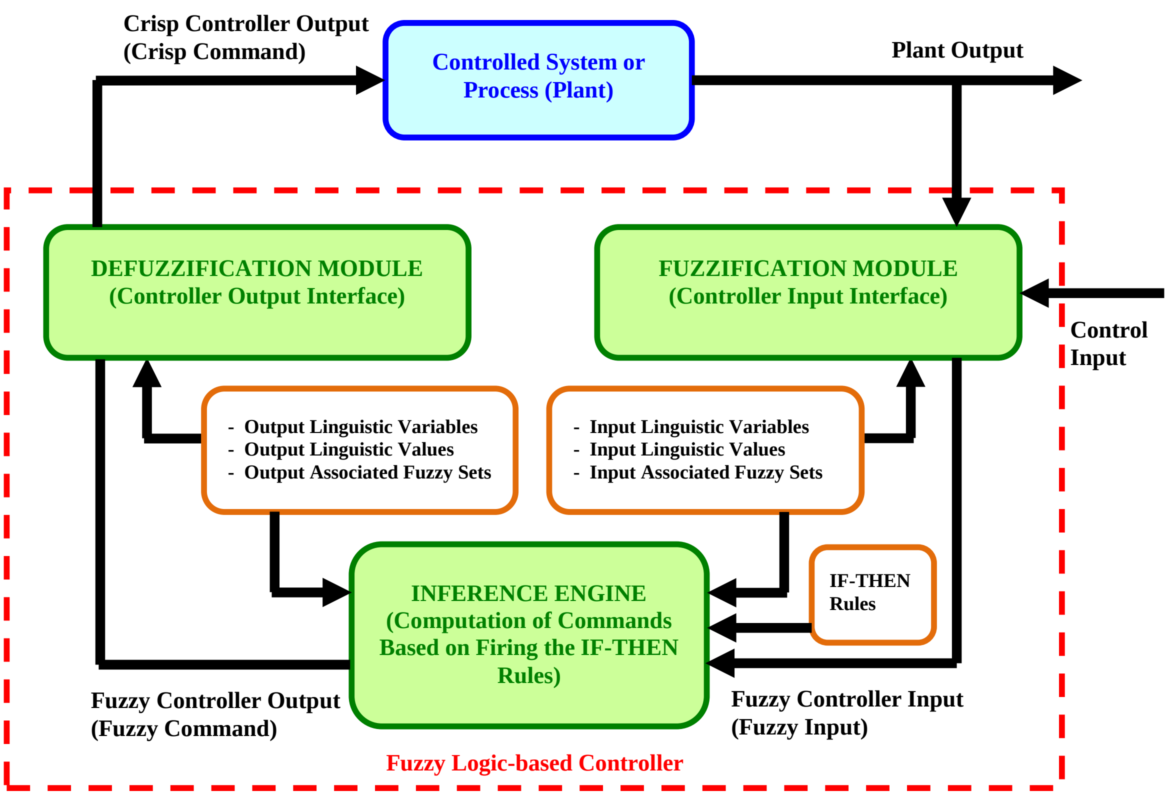

Fig. 2. General block diagram of a fuzzy logic-based controller

Two applications of fuzzy logic for control and detection purposes are also presented and discussed in class [33,34].

D. Artificial Neural Networks (ANN)

The attempts to model the operation of the brain and use these models for technological applications resulted in a new information processing paradigm, widely known as ANN, or *connectionist models*, or *distributed parallel processing*. It should be noted that it is no longer the purpose of ANN to model the brain, or other parts of the nervous system. The biological counter-parts of ANN serve only as inspiration and starting point for the development of generalized function approximators [35,36] with wide areas of potential application.

The students are first introduced to the general structure, components, and functionality of the biological neuron and how this source of inspiration is used to build a simple multi-input-single-output computational unit, the artificial neuron. The general architecture of the neuron model, which is used for the quasi-totality of ANNs, is presented in Fig. 3. Correspondences between the model components and their biological counterparts are put into evidence.

Interconnecting a potentially large number of artificial neurons produces an ANN. The different structural characteristics covered are listed next:

- informational flow direction: feedforward (static) ANN and feedback (dynamic or recurrent) ANN;
- stratification: single or multiple layer ANNs with fixed or variable number of neurons;
- connectivity: partial or full connection, excitatory or inhibitory connection, and fixed or variable weight connection.

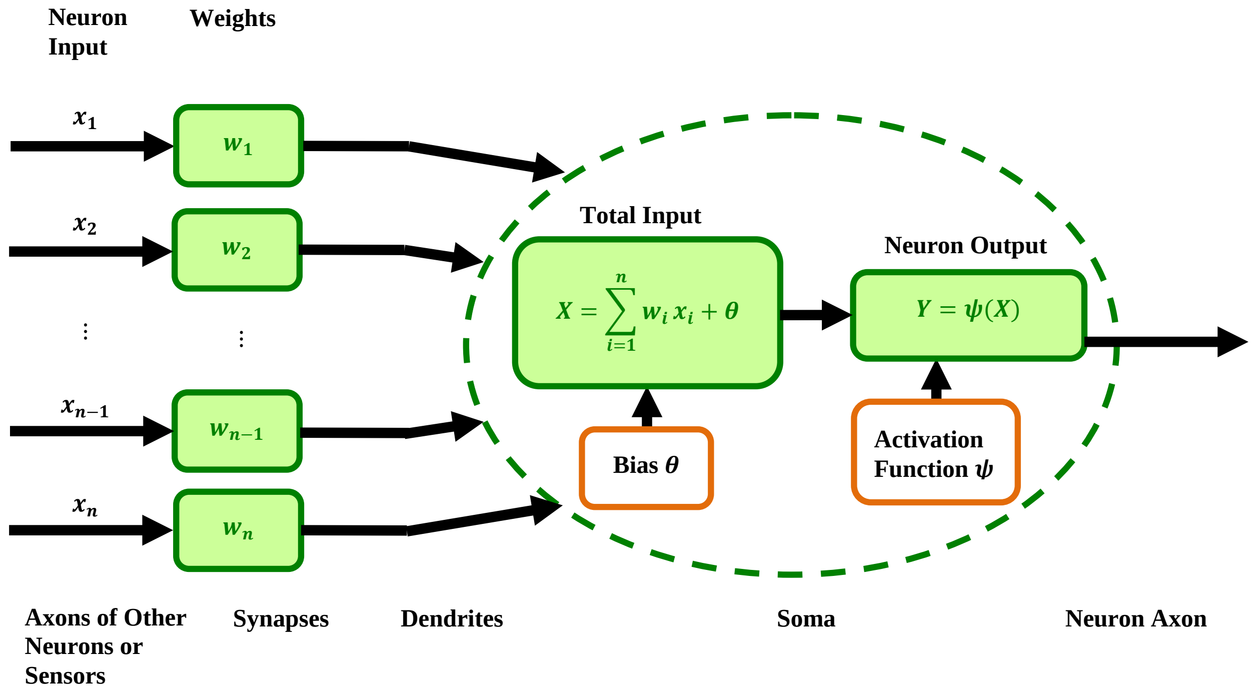

Fig. 3. Artificial Neuron

The different alternatives for selecting the artificial neuron activation function (AF) are discussed next, including: unipolar and bipolar AFs, linear and semi-linear AFs, sigmoid, and radial basis AFs.

The *learning* (or *training*) process of an ANN is a critical component of the ANN design and consists primarily of adjusting the synaptic weights until the "*desired*" output (evaluated with specific criteria) is obtained in response to inputs of the learning data set. *Error-based* or *supervised* learning algorithms that rely on the availability of input/output training data are covered in the course. Students are introduced to the delta learning rule and the back-propagation algorithm.

To illustrate the effects of different ANN design parameters and provide insight into the learning process, an interactive classroom exercise is performed consisting of implementing the logical function "OR" using a perceptron. It should be noted that, while the practicality of such implementation is null, the exercise is a valuable academic tool because it is manageable on the blackboard and provides rapid and clear insight into ANN internal mechanisms and parameter effects. Instructor and students work together through the iterations of the learning process. The following parameters are varied and their effects on the convergence and effectiveness of the learning process are analyzed: selection of training data set, learning rate, initial weights, and order of learning iterations.

In the next section of the course, students are introduced to different ANN architectures and algorithms including: single hidden layer ANN with sigmoid AF, radial basis function (RBF) ANN, ADALINE, Sigma-Pi ANN, and extended minimal resource allocating ANN. A classroom exercise is performed consisting of training a Gaussian single layer RBF ANN for approximating a non-linear function using the back propagation learning algorithm. Instructor and students run a provided Matlab® code and vary the following parameters to illustrate their effects on the ANN design: number of neurons, location of neurons, learning rate, and Gaussian width.

Two applications from the literature are presented and discussed in class [37,38], which illustrate the use of ANN for control and modeling purposes, respectively.

V. Course Assignments

A. *Team Projects*

Students are expected to perform three projects, one in each major area of the course (GA, FL, and ANN). Students work in teams of 3 or 4 people and are expected to be involved equally in all aspects of the project. Student/instructor brain storming sessions are organized for each team to select and formulate the project topic and assignment. The specific project topics are based on proposals from the teams and are eventually established to match the research interests of the students. Weekly status reports, a final project technical report, and a peer assessment of individual team member contribution are required and are part of the project grade. Each team is expected to deliver to the class a 10-15 minute presentation of their projects, as part of the assignment.

The strategy in formulating project assignments is driven by the need of meeting time constraints, while providing the opportunity to experience and address all typical issues raised by the design and implementation of the AI techniques. The complexity and the dimensionality of the problems may be reduced to manageable levels, even if the practicality of the problem is affected. For instance, for a GA optimization project, only 3 or 4 relevant optimization parameters may be considered, even if the real problem would involve a much larger number. On the other hand, over-simplification is not allowed, in order to not miss relevant aspects. For example, at least two non-redundant and contradictory optimization criteria must be considered when formulating a GA problem, such that a composition rule must be used.

It is recommended that the students develop their own code and do not use Matlab® specialized toolboxes, such that they get a better exposure to the functionality of different elements. However, under special circumstance, using some of these tools is allowed and even necessary, in order for the teams to complete the project within the academic constraints.

Examples of GA student projects performed over the past few years include:

- Tuning of a biomechanical model of a walking person;
- Multiple-burn orbital transfer optimization using a GA;
- Optimization of engine parameters for reduced fuel consumption and emissions;
- Optimization of a DC motor operation;
- Optimization of CO2 absorption process.

Some of the FL projects proposed and completed by students within this course are listed next:

- Simulated car driver using FL algorithms;
- FL-based control of quadcopter;
- FL control of aircraft longitudinal dynamics;
- Automated parallel car parking using FL-based controller;
- FL-based control of overflow dam gate.

Finally, the following are examples of topics for the ANN team project:

- Inverse kinematics problem for a 2D robotic manipulator;
- ANN model of virus infection propagation;
- ANN equivalent replacement of FL-based battery current limits on motor torque;
- ANN for hand-written numeral recognition;

B. Quizzes

A total of 6 announced quizzes are recommended with the tentative scheduling presented in Table I. The first quiz covers general aspects of AI techniques characteristics, classification, and applicability. The second quiz addresses basic concepts relevant to GA, and may be characterized as more "theoretical", whereas the third requires that the student apply specific algorithms (e.g. roulette-wheel selection method) and solve numerically more "practical" problems. A similar strategy is used for the fourth and fifth quizzes, this time relative to FL and FL-based applications. Finally, the sixth quiz questions students on the basic concepts of ANN and the use of delta rule for ANN training.

C. Final Exam

Final exam is comprehensive and closed notes. Questions and problems on the final exam have the same structure, content, and level of difficulty as the ones on the quizzes. An average quiz grade of 90 or higher makes the final optional. In this case, the final grade for the course is scaled up by 100/85.

D. Grading

The recommended grading scale is:

- Projects: 25% each
- Final exam: 15%
- Quizzes: 10%

with the typical letter grade distribution:

- A= 90% - 100%
- B= 80% - 89%
- C= 70% - 79%
- D= 60% - 69%
- F= <60%

VI. Impact and Student Feedback

The introductory course to computational AI techniques has been offered regularly at WVU over the past few years with an average enrollment of 16 students. Mechanical and aerospace engineering graduate students combined formed the majority, followed by chemical, civil, industrial, and electrical engineering. On the average, 1-2 undergraduate students per year attended the course with excellent performance.

The WVU student evaluation of instruction (SEI) survey includes a total of 17 questions with ratings between 1 (poor) and 5 (excellent). Three questions on the overall course rating are primarily used as evaluation metrics. They address the quality of the course, the teaching effectiveness of the instructor, and the overall student perceived learning. The average SEI score for the course for all questions was 4.71 and for the primary 3 questions, 4.64, with a distribution in a fairly narrow range (Table II).

Approximately 70% of the students used at least one AI technique in their graduate program research and/or in other projects.

TABLE II
Average Course SEI

Metric	Score
Course quality	4.66
Teaching effectiveness	4.66
Learning	4.61
Primary course rating (3 questions)	4.64
Overall course rating (17 questions)	4.71

The comments provided by the students at the end of the course, as part of the SEI survey, primarily reflect the following aspects. The active and experiential learning environment created through the design of the team projects was found to be very beneficial for the learning process. The exposure to computational AI techniques triggered interest and was convincing relative to their capabilities for real-world applications. Students gained confidence in the potential and value of the techniques they learned and in their own ability to implement the techniques successfully. Some of the student comments are presented next:

- *"The course and the instructor were excellent because we could apply the knowledge in practical projects."*
- *"I will use what I learned in the class in my research and career."*
- *"Great course! It successfully introduced me to the ground level of the three different subjects presented. I now have an interest/foundation to take more courses in these areas."*
- *"The setup of this course was very effective. The projects really drove the concepts home. I would recommend this class to others."*
- *"I think this subject was very interesting and now I gain a basic understanding of AI that I will continue exploring in the future."*
- *"Course is well structured and the clarity of the handouts was excellent."*
- *"Great course structure, with many opportunities for learning and showing proficiency in different ways (e.g. quizzes + projects). Enthusiasm."*
- *"I had… FUN!"*

VII. Conclusions

An introductory course on computational AI techniques for graduate students outside CS curriculum has been successfully designed and taught at WVU.

The course demonstrated that engineering students can develop, implement, and effectively use GA, FL, and ANN with the basic instruction provided by a 3 credit hour course.

Active and experiential learning methodologies have been successfully implemented, primarily through the team projects. They allowed to directly connect course material to real-world problems and offered a certain level of student autonomy that facilitated initiative, creativity, strong team interaction, and decision making.

The technical area covered and the academic design of the course were well received by students, increasing their motivation, participation, and interest. The computational AI techniques were perceived as potentially valuable additions to student professional background.

References

[1] A. P. Engelbrecht, *Computational Intelligence, An Introduction*, Second Edition, Wiley & Sons, ISBN 978-0-470-03561-0, 2007.

[2] H. Ishibuchi, (editor), *Computational Intelligence*, Eolss Publishers Co. Ltd., United Kingdom, 2015.

[3] D. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, 1999.

[4] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, 3rd Edition, Springer-Verlag, Berlin, 1992.

[5] W. Pedrycz, *Fuzzy Control and Fuzzy Systems*, John Wiley & Sons Inc., 1993.

[6] D. Driankov, H. **Hellendoorn, and M. Reinfrank,** *An Introduction to Fuzzy Control*, Springer Verlag, Berlin, Heidelberg, 1996.

[7] B. J. A. Kroese and P. P. van der Smagt, *An Introduction to Neural Networks*, University of Amsterdam, 1993.

[8] K. Gurney, *An Introduction to Neural Networks*, UCL Press, UK, 1997.

[9] D. Dasgupta and Z. Michalewicz, *Evolutionary Algorithms in Engineering Applications – Mechanical and Industrial Engineering*, Springer Verlag, 1997.

[10] G. Winter, J. Periaux, and M. Galan, (editors), *Genetic Algorithms in Engineering and Computer Science*, John Wiley & Sons, Ltd., 1995.

[11] A. Kusiak, *Computational Intelligence in Design and Manufacturing*, Wiley-Interscience, 2000.

[12] C. C. Bennett and K. Hauser, "Artificial Intelligence Framework for Simulating Clinical Decision-Making: A Markov Decision Process Approach", *Artificial Intelligence in Medicine*. 57(1), pp: 9-19, 2013.

[13] R. Belazzi, A. Abu-Hanna, and J. Hunter, (editors), "Artificial Intelligence in Medicine", *Proceedings of the 11th Conference on Artificial Intelligence in Medicine*, AIME 2007, Amsterdam, The Netherlands, July 7-11, 2007.

[14] S. H. Chen, P. Wang, and P. P. Wang, *Computational Intelligence in Economics and Finance*, Advanced Information Processing Series, Springer 2006.

[15] J. Liebowitz, *Strategic intelligence: Business intelligence, competitive intelligence, and knowledge management*, Taylor & Francis, Inc., 2006.

[16] S. H. Chen, A. J. Jakeman, and J. P. Norton, "Artificial Intelligence techniques: An introduction to their use for modelling environmental systems", *Mathematics and Computers in Simulation*, Vol. 78, Iss. 2–3, pp: 379-400, July 2008.

[17] S. E. Haupt, A. Pasini, and C. Marzban, *Artificial Intelligence Methods in the Environmental Sciences*, Springer, 2009.

[18] CS2008 Review Taskforce, *Computer Science Curriculum 2008: An Interim revision of CS 2001*, ACM, IEEE Computer Society, 2008, available at http://www.acm.org/ education/curricula/ ComputerScience2008.pdf

[19] N. Pillay, "Artificial Intelligence in Computer Science Teaching and Research." *Conference'04*, Vol. 4, pp: 1-5, South Africa, 2004.

[20] A. Minale, P. Sanati-Mehrtzy, A. Sanati-Mehrtzy, and R. Sanati-Mehrtzy, "Computational Intelligence Course in Undergraduate Computer Science and Engineering Curricula", *20th ASEE Annual Conference and Exposition*, Atlanta, GA, 2013.

[21] D. Kumar, and L. Meeden, "A robot laboratory for teaching artificial intelligence", *Proc. of* ***29th*** *Technical Symposium on Computer Science Education, SIGCSE'98*, pp: 342-344, Atlanta GA, Mar. 1998.

[22] J. DeNero and D. Klein, "Teaching Introductory Artificial Intelligence with Pac-Man", *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI-10)*, pp:1885-1889, Atlanta, GA, 2010.

[23] . Borm and L. Rothkrantz, "Teaching Artificial Intelligence Techniques Using Multi-Agent Soccer", *3rd E-Learning Conference*, Coimbra, Portugal, Sept. 2006.

[24] M. Black, "A One-Credit Artificial Course for a General Audience", *Proceedings of Consortium for Computing Sciences in Colleges Conference*, 2010.

[25] M. Silberman, *Active learning: 101 strategies to teach any subject*, Allyn & Bacon, Needham Heights, MA 1996.

[26] D. Kolb D., *Experiential Learning: Experience as the Source of Learning and Development*, Prentice-Hall, Englewood Cliffs, NJ, 1984.

[27] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*, University of Michigan Press, Ann Arbor, MI, 1975 (revised and republished in 1992).

[28] L. Davis, "*Handbook of Genetic Algorithms*", Van Nostrand Reinhold, New York, 1991.

[29] M. G. Perhinschi, "Controller Design for Autonomous Helicopter Model Using a Genetic Algorithm", *Proc. of the American Helicopter Society 53rd Annual Forum*, Virginia Beach VA, USA, pp. 788-794, 1997.

[30] M. G. Perhinschi, "Parameter Optimization Via Genetic Algorithm of Fuzzy Controller for Autonomous Airvehicle", *Proc. of the AIAA Guidance, Navigation, and Control Conf.*, Portland OR, USA, pp. 790-797, Aug. 1999.

[31] M. G. Perhinschi, C. Marlowe, S. Tamayo, J. Tu, and W. S. Wayne, "Evolutionary Algorithm for Vehicle Driving Cycle Generation", *Journal of the Air & Waste Management Association*, Vol. 61, pp 923-931, Sept. 2011.

[32] L. A. Zadeh, "Fuzzy Sets", *Information and Control* no. 8, 1965.

[33] M. G. Perhinschi, B. Smith, and P. Betoney, "Fuzzy Logic-Based Detection Scheme for Pilot Fatigue", *Aircraft Engineering and Aerospace Technology: An International Journal*, Vol. 82, Iss. 1, pp. 39-47, Jan.-Feb. 2010.

[34] M. G. Perhinschi., "Parameter Optimization Via Genetic Algorithm of Fuzzy Controller for Autonomous Airvehicle", *Proceedings of the*

AIAA Guidance, Navigation, and Control Conference, Portland OR, USA, pp: 790-797, August 1999.

[35] B. Kosko, *"Neural Networks and Fuzzy Systems"*, Englewood Cliffs, Prentice Hall, NJ, 1992.

[36] L. Fausett, *"Fundamentals of Neural Networks"*, Prentice-Hall, 1994.

[37] M. G. Perhinschi, M. R. Napolitano, G. Campa, B. Seanor, and S. Gururajan, "Design of Intelligent Flight Control Laws for the WVU F-22 Model Aircraft", *Proceedings of the AIAA Intelligent Systems Technical Conference*, Chicago IL, AIAA2004-6282, 2004.

[38] M. G. Perhinschi, W. S. Wayne, N. N. Clark, and D. W. Lyons, "Neural Network Modeling of Emissions from Medium-Duty Vehicles Operating on Fisher-Tropsch Synthetic Fuel", *Proceedings of the SAE World Congress*, Detroit, MI, April 2007.

Mario G. Perhinschi is a Professor with the Department of Mechanical and Aerospace Engineering at West Virginia University. He is teaching courses in Feedback Control, Flight Simulation, Mechatronics, Aircraft Health Management, and Artificial Intelligence Techniques. His current research interests include primarily design of intelligent fault tolerant control laws, trajectory planning and tracking for unmanned aerial vehicles, and development of advanced simulation environments for aerospace and mechanical engineering teaching and research.