Artificial Intelligence Techniques: How Can it Solve Problems in Power Electronics?

An advancing frontier

by Bimal K. Bose

Artificial intelligence (AI) techniques have recently emerged as the powerful and advancing frontier in power electronics and their impact in the next generation of power electronics appears to be tremendous. The advent of AI has brought a challenge to the engineers specialized in power electronics which is already a complex, interdisciplinary and fast-moving technology. The AI provides very powerful tools for design, simulation, control, estimation, fault diagnostics, and fault-tolerant control of power electronic systems, which are particularly important in modern smart grid and renewable energy systems (RESs) in addition to its applications in traditional power electronics and motor drives area. A smart or intelligent grid is basically an advanced electric power grid of tomorrow using state-of-the-art technologies in power electronics, power systems, computers, communications, information, AI and cyber that will improve the system availability, reliability, power quality, energy efficiency, and security with optimum resource utilization and economical electricity to the consumers.

Digital Object Identifier 10.1109/MPEL.2020.3033607
Date of current version: 16 December 2020

What is AI? AI basically deals with computer emulation of human thinking. A human brain with a biological nervous system has natural intelligence and ability to think, that is, to learn, reason and comprehend. The goal of AI is planting human intelligence in a computer so that the computer can think intelligently and solve problems like a human being. Can a computer really think and take intelligent decision? Computational intelligence has been a debatable subject for a long time. It is true that the intelligence of a computer is far inferior to the natural intelligence of a human being. But it is also true that computer can help solving complex problems which are difficult to solve by traditional methods. The AI technology has gone through evolution in the past several decades, and has been extensively applied in information management, space exploration, military systems, finance, industrial and power systems, etc. which also include power electronics and motor drives. AI techniques for applications in power electronics can be generally classified into four different areas. These are: Expert systems (ES), fuzzy logic (FL), artificial neural network (ANN or NNW), and genetic algorithm (GA) or evolutionary computation (EC). However, the concept of machine learning (ML) that involves the study of computer algorithms that improve automatically through experience has evolved over a long time. Lotfy Zadeh, the inventor of fuzzy logic, defines ES as "hard" or precise computing, whereas FL, NNW, and GA are defined as "soft" or approximate computing. The technology has gone through fast evolution in the past several decades. In the recent years, AI has been almost synonymous with the NNW applications. The potentiality of this area appears to be tremendous. In this article, the principles of ES, FL and NNW will be briefly reviewed along with discussion of an example application in each area.

Expert System

The ES, also known as classical AI, is the forerunner of all the AI techniques. It is basically an intelligent computer program that is designed to embed the expertise of a human being in a certain domain, such as power electronics. A power electronics engineer acquires the expertise by education and training over a prolonged period of time. The objective of an ES is to replace a power electronics expert by the advanced computer-embedded program that is designed to solve the problem [1]. Figure 1 shows the basic elements of an ES with interconnection as follows:

- Knowledge Base
- Inference Engine
- User Interface
- Explanation Subsystem for user education.

The core of the ES is the knowledge base which consists of expert knowledge that is supported by database as shown. The expert knowledge basically consists of a matrix of IF .... THEN rules using Boolean variables which are similar to logical thinking of a human being to solve a problem. An example rule for fault-tolerant control of a converter-fed ac drive system may be

IF converter dc link voltage < 200 V AND
ac line voltage $= 0$ AND
motor speed > 1500 rpm
THEN reduce motor speed by 20% by regenerative braking.

The expert knowledge is acquired from the domain expert (i.e. a power electronics engineer) by a software expert defined as knowledge engineer. The inference engine is basically the controller or executive software of the ES that tests the rules of the knowledge base in sequence and tries to draw inferences or conclusions that are supplied to the user. The inference engine communicates with the user in a very user-friendly dialog in a natural language, such as English. Through the inference engine, the ES requests the parameter values of the rules from the user to draw conclusions, or solution of the problem. One very important function of the ES is the user education through the explanation subsystem.

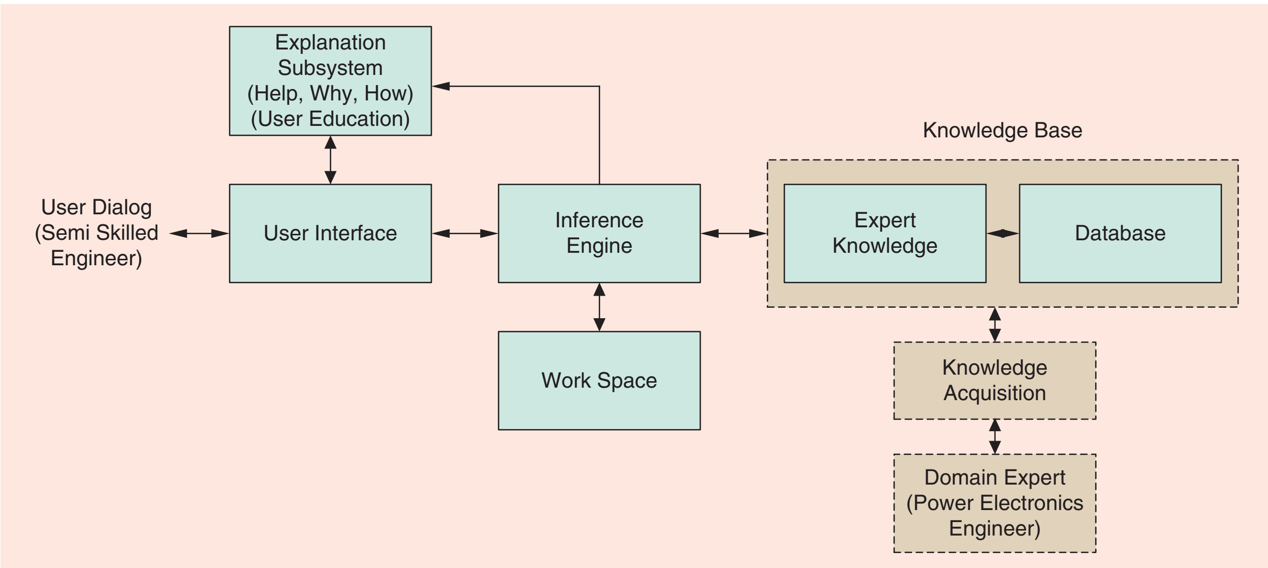

FIG 1 ES block diagram showing the different elements and their interconnection.

This subsystem responds during the problem solving by the HELP, WHY and HOW commands of the user. The ES user is usually a semi-skilled person and does not understand in depth the complexity of problem solving. The ES explains to the user the technical features of the problem with explanatory texts and figures when HELP command is used. The WHY command explains to the user why the ES is asking the relevant information from the user, and HOW command explains how the problem has been solved. Often, user education is the sole purpose for designing the ES.

The structure of a knowledge base with the matrix of 16 rules, for example, can be read as

$$\text{Rule 1:}\quad \text{IF } X = X_1 \text{ AND } Y = Y_1\text{, THEN } Z = A \text{ .... (1)}$$
$$\text{Rule 2:}\quad \text{IF } X = X_2 \text{ AND } Y = Y_2\text{, THEN } Z = B \text{ .... (2)}$$
$$\vdots$$
$$\text{Rule 16:}\quad \text{IF } X = X_4 \text{ AND } Y = Y_4\text{, THEN } Z = P \text{ ...... (16)}$$

where X, Y and Z are defined as the rule parameters, and X_1, Y_1, X_2, Y_2, etc. are the respective parameter values. The data base that supports the knowledge base embeds the parameter values which can be in the form of data (logical, numeric, facts or statements). A rule has a conditional (premise or antecedent) part in the IF statement and the action (consequent or conclusion) part in the THEN statement. The logical connectives can be AND, OR and NOT operations for drawing conclusion. A rule can be fired or executed if the conditional part is true (called forward chaining), and then the action part guided by the THEN statement is executed. In backward chaining, the system starts with the desired conclusion and then finds the rules that could have caused the conclusion. In a practical system, there can be large number of rules and the parameters with the values may also be large and complex. The knowledge base can be simple or adaptive in nature, depending on the system changes or technology advancement. Knowledge can be defined as shallow or deep. Shallow knowledge base can be directly obtained from the present knowledge of the system. A deep knowledge can be derived by machine learning from the system model and simulation responses based on designer's or researcher's knowledge.

The ES knowledge is normally structured or represented in the form of a tree with the help of a number of frames. A frame consists of a cluster of characteristic rules and their associated parameters. The advantage of frame-based structure of knowledge base is the logical organization of a large amount of knowledge in the modular form. The root frame is the core of the knowledge base. It may have child frames and grandchild subframes. Each subframe can be considered as a subdomain of expert knowledge. Assume, for example, a customer wants to select a certain commercial drive product [2] from a vendor for a certain application with the consultation of an ES program. In this case, the root frame embeds the expertise of a general sales engineer. Consider that there are two child subframes which embed the application engineer's expertise of cage type induction motors (IM) and PM synchronous motors (PMSM), respectively. The user interfaces the root frame in the beginning, and based on the user dialog and then consultation with IM and PMSM drive child subframes, the IM drive appears to be the choice. Once the type of drive is selected, the details of the converter and the machine will be calculated by the ES based on the specifications supplied by the user. The iterations of computation can be done by the ES until the user is fully satisfied with the ratings and performances of the drive. A grandchild subframe will then provide the auxiliary features of IM drive, such as price, delivery and installation considerations.

An ES SHELL is a software environment platform [3] for efficient and user-friendly development of ES program. The SHELL can interface with external programs, such as data base files, graphical files, simulation files, interface routines, and mathematical files. A limited amount of data, logical and arithmetic capabilities can be directly embedded in the ES program, but for large data, such as product catalog file, database files should be constructed. For complex calculations, such as solving differential equations, the ES program can access external programs. Similarly, the SHELL can control and transfer data to and from simulation programs. A powerful feature of shell is that it can integrate pictures with the knowledge base which are compressed as files. Thus, a fully designed power electronic system can be the output of the ES in the form of a circuit schematic showing all the numerical values of the parameters. A number of ES shells are available to design an ES program. One example is PC Plus [3] developed by Texas Instruments. It uses PC SCHEME language which is a dialect of LISP language. The designer should have familiarity with PC SCHEME although English-like Abbreviated Rule Language (ARL) is used for fast development of the program. For time-critical real-time applications, C language should be used. If the program is resident in the SHELL, the developer can easily alter or update it, but program modification is not possible in client computer environment.

Expert System Based Control of Smart Grid

In the past, ES has been applied extensively in many power electronic systems [1]. The applications include automated P-I tuning of drive, power electronic system fault diagnostics, selection of commercial drive product, drive system configuration selection, design and simulation that generates real-time controller object code for a DSP, and control strategy development of smart grid. As an example of ES application, we will discuss briefly the preliminary control of smart grid based on ES [4]. Suffice to say that control and protection of a modern smart grid are extremely complex, particularly if the grid is large. In the present status of the technology, the control and protection strategy of future large smart grid is not yet well-defined.

As mentioned before, a smart grid is basically an advanced electric power grid using state-of-the-art technologies that will improve system availability, reliability, power quality, energy efficiency, and security with optimum resource utilization and economical electricity to the consumers. Figure 2 shows a simplified control block diagram of smart grid based on ES. The grid incorporates a large segment of RES along with bulk power generation by fossil and nuclear power plants. There are bulk energy storage plants (mainly to support RES), HVDC systems, flexible ac transmission system (FACTS), VAR compensating static compensators (STATCOMs), and modular multilevel converters (MMCs), which are distributed strategically throughout the grid. The whole grid is segmented into a number of regions which are under the control of a centralized master controller, as indicated in the Figure 2. The master controller at the center of the Figure has an ES oriented knowledge base, where the rules are formulated on the basis of extensive *a priori* offline analysis, design and simulation of the whole system. This is a massive task that requires a large number of iterations. The ES rules usually have a large number of parameters with different values. The knowledge base can be updated continuously as more knowledge is gained about the system. A simple typical rule may have the structure as

IF photovoltaic station A output < 20 kW AND
wind station D output > 1000 kW,
THEN trip circuit breaker M and assign
battery storage station D output $= 120$ kW.

The knowledge base can also incorporate fuzzy inference system (FIS) based rules which will be described later. The master controller has an operator interface on the left and a real-time simulator (RTS) interface on the right, and controls all the regional controllers are indicated in the Figure. All the regional controllers have local override in case of a problem with the master controller. The RTS is based on supercomputers which embeds the dynamic model of the whole grid including the power electronic systems that require very fast response. The virtual or simulated grid runs essentially independently from the actual grid. The RTS helps the master controller to generate the command and control signals for the regional controllers based on real-time solution of the virtual grid. The RTS receives the whole system signals, whereas the regional controllers receive only the regional signals. A large number of grid signals can be obtained through system-wide PMUs (phasor measurement units). The signals may be sensor-based or sensorless. Redundant signals may be needed for unreliable signals.

The operator interface in the central control station is assisted with large display boards (not shown) that monitor all the key signals (such as system frequency, bus voltages, active and reactive power flow in lines, including the status of the system operation for information and assistance of the operators). Note that FL and NNW based controls can also be used within the elements of the smart grid system. The reliable operation of hardware and software of the controllers and the RTS are extremely important for reliable grid operation. For this reason, some redundancy in hardware and software should be built in. The RTS can also be used for initial planning, analysis and design of the grid and to formulate optimum default control strategy with the predicted loading, generation and energy storage capacity curves, including the planned outages of the equipment, but this discussion is beyond the scope of this article. Note that RTS can also be used to study future expansion of the grid.

Fuzzy Logic

FL, unlike Boolean logic, is multivalued, i.e., its value varies between 0 and 1. FL is used to solve problems that have fuzziness or uncertainty. The general methodology of reasoning in FL is also based on IF …. THEN rules like ES. A general rule in FL may read as

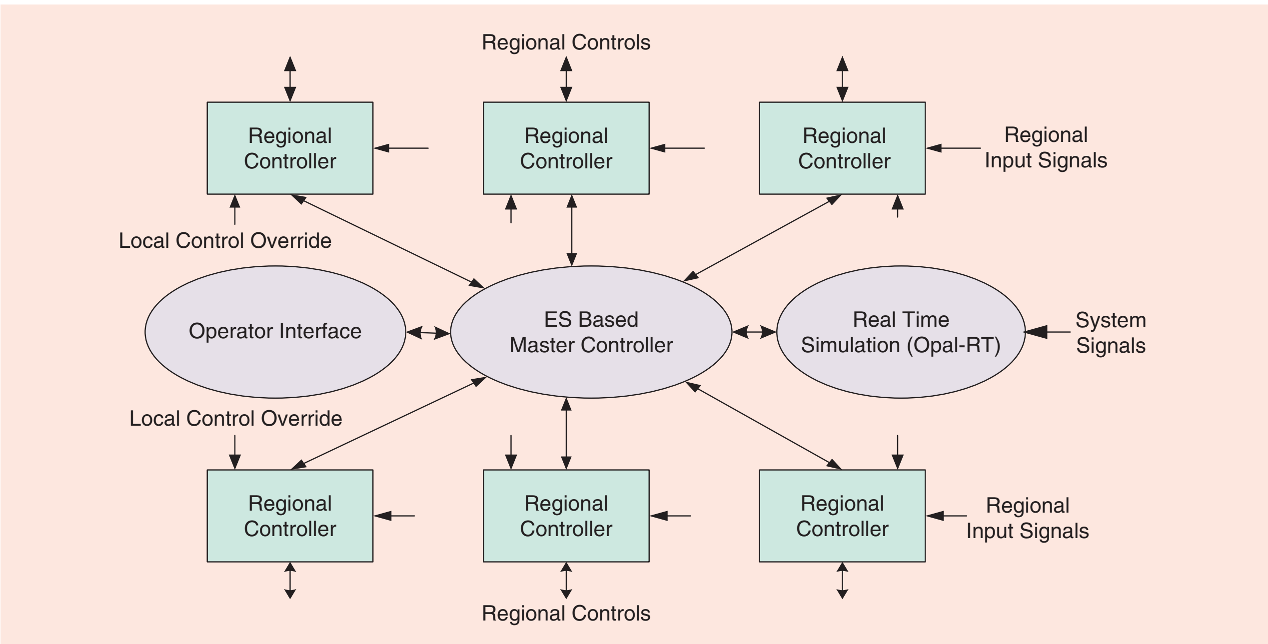

FIG 2 Simplified control block diagram of smart grid based on ES.

IF motor speed is high and its stator temperature is medium
THEN set the machine stator current low.

The fuzzy variables (underlined) in the rule are normally represented by membership functions (MFs). Figure 3 shows the general principle of Mamdani type fuzzy control system (defined as fuzzy inference system or FIS) using triangular MFs. Mamdani method is most commonly used in literature. Here, X, Y and Z are the fuzzy variables, where X and Y are the inputs and Z is the output. Each fuzzy variable is represented by three MFs, as indicated in the figure. The fuzzy variables can be unipolar or bipolar, and range of values of each is defined as a "universe of discourse". In the FIS, there are three control rules as indicated in the figure which can be defined as

Rule 1: IF X is negative small (NS) AND Y is zero (ZE) THEN Z is positive small (PS)
Rule 2: IF X is zero (ZE) AND Y is zero (ZE) THEN Z is zero (ZE)
Rule 3: IF X is zero (ZE) AND Y is positive small (PS) THEN Z is negative small (NS).

For convenience, fuzzy variables are normally defined on a per unit (pu) basis. All the rules in FIS are normally summarized as a rule matrix in the form of a table.

Because of the analogy, a FIS is often defined as fuzzy ES. The computations in FIS consist of the following five steps to determine the inference or conclusion

1) Fuzzification of the input crisp variables (X, Y, etc.);
2) Application of fuzzy operator (AND, OR, NOT) in the IF (or antecedent) part of the rule;
3) Implication from the antecedent to the consequent (THEN) part of the rule;
4) Aggregation of the consequents of the rules;
5) Defuzzification to convert the fuzzy output to crisp value.

All the three rules in the figure are shown in the consecutive horizontal rows with the input values of $X = -3$ and $Y = +1.5$. Since all the rules use the AND operator, the degree of fulfilment (DOFs) from the antecedent parts of the rules are calculated as $DOF_1 = 0.6$, $DOF_2 = 0.4$ and $DOF_3 = 0.4$, respectively. The corresponding fuzzy output MFs (shaded) are PS', ZE' and NS', respectively from the consequent part of the rules. The total fuzzy output, as shown in the figure, is the aggregation of the consequents of the rules. The defuzzification calculates the center of the overlapped area (COA), where Z_0 is the defuzzified or crisp output. Note that a practical FIS can have a large number of input and output variables and each variable may have a large number of MFs. A FIS can be designed with the help of Math Works Fuzzy Logic Toolbox [5].

Fuzzy Logic Control of Modern Wind Generation System

FL has been applied extensively in power electronic systems [1]. The applications include robust control of drives with parameter variation, P-I tuning of control system, system

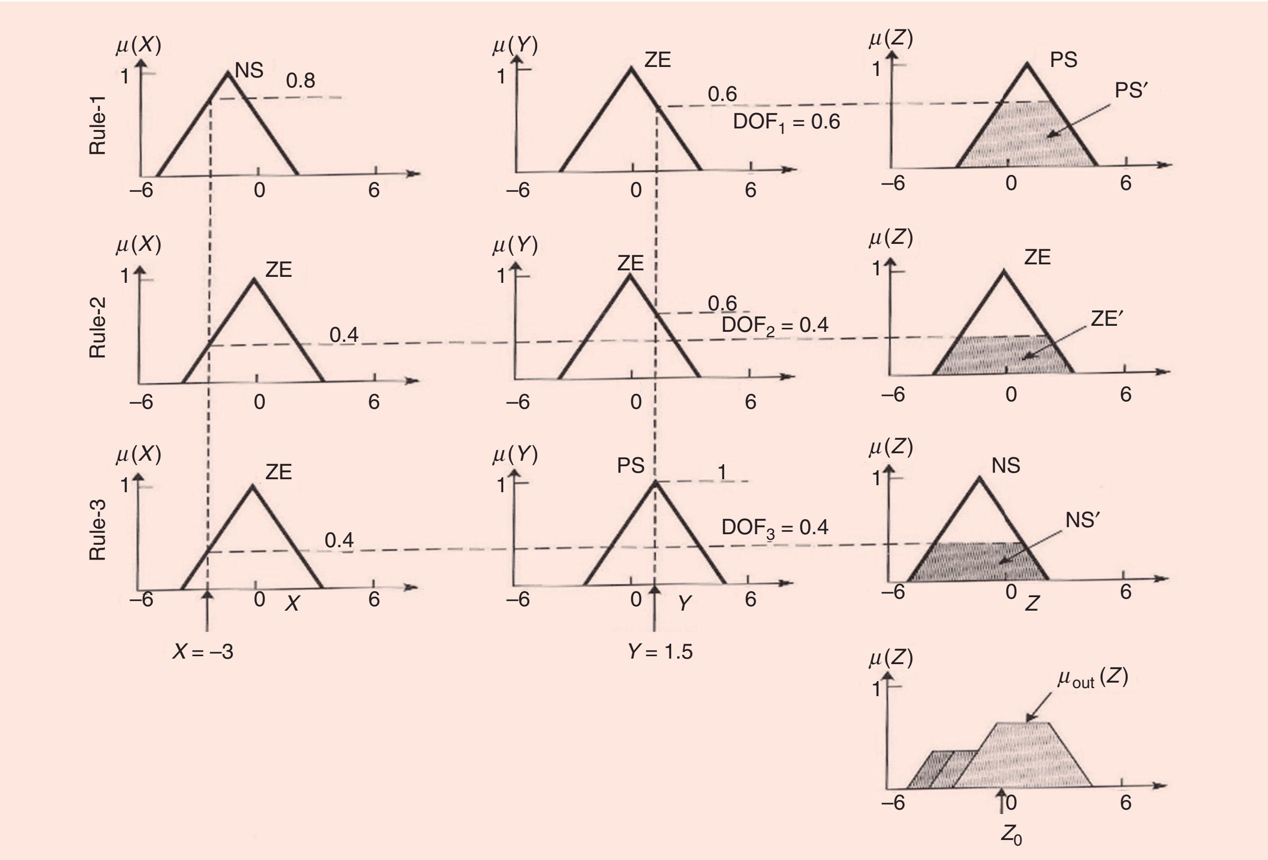

FIG 3 Three-rule Mamdani type FIS showing the output defuzzification.

nonlinearity compensation, control optimization based on online search, estimation of parameters, slip-gain control of induction motor drive, online diagnostics and fault-tolerant control, etc. In this section, fuzzy logic based modern wind generation system control will be discussed as an example.

Figure 4 shows a modern wind generation system [4] that uses a horizontal axis variable speed wind turbine, IPM synchronous generator and two-sided PWM converter system. The machine is gearless to improve weight, efficiency, reliability, cost, and noise, although the machine has some cost and weight penalty for such a direct drive system. As a result, the generator cut-in speed is somewhat lower. The IPM machine (with NdFeB magnet) has high efficiency and armature reaction effect due to salient poles that permits stator flux programming for light-load efficiency optimization control. Both the converters are direct vector controlled with synchronous current control and space vector PWM that permit fast response and harmonic-free machine and line currents further improving the drive efficiency. The generator uses stator flux (Ψ_s)- oriented vector control, as indicated in the figure. The active power (P_0) (for dc-link voltage v_d control) and reactive power (Q_0) sharing of the line-side converter are also indicated.

The wind generation system in Figure 4 uses three fuzzy controllers (FLC-1, FLC-2 and FLC-3)[1], as shown, with Mamdani-type FIS in all of them. The operation of the fuzzy controllers FLC-1 and FLC-2 is explained in the power output curves of wind generation system in Figure 5 which gives output power (P_0) as function of generator speed (ω_r) at different wind velocity (V_w). The FLC-1 is an online search based maximum power point tracking (MPPT) (MPPT-1) control, where optimum turbine speed (ω_r) is searched for maximum line power (P_0) at constant (V_w) so that the turbine operates at maximum aerodynamic efficiency. In a fuzzy MPPT-1 control, the inputs are power increment ΔP_0 and the last increment/decrement of $L\Delta\omega_r^*$ and the output is the command speed increment $\Delta\omega_r^*$ (the details are not shown). The controller FLC-1 brings the operating point from A at speed ω_{r1} to the maximum power point B at speed ω_{r2} when working at constant wind velocity V_{w4}. Then the controller FLC-2 works to bring the operating point to C at the same speed. The FLC-2 operates on the same online search principle (MPPT-2) to set the optimum light-load flux command Ψ_S^* as a function of torque command T_e^* for best light load generator efficiency. If V_w now increases from V_{w4} to V_{w2}, the operating point jumps to D. Further operation of the fuzzy controllers will bring the operating point to F. The effect on decrease of V_w is also indicated in the figure. Note that no information on wind velocity is needed in the fuzzy controllers. Evidently, the fuzzy controls are abandoned when the wind velocity changes. The online search-based algorithms are insensitive to plant parameter variation. The fuzzy P-I speed controller

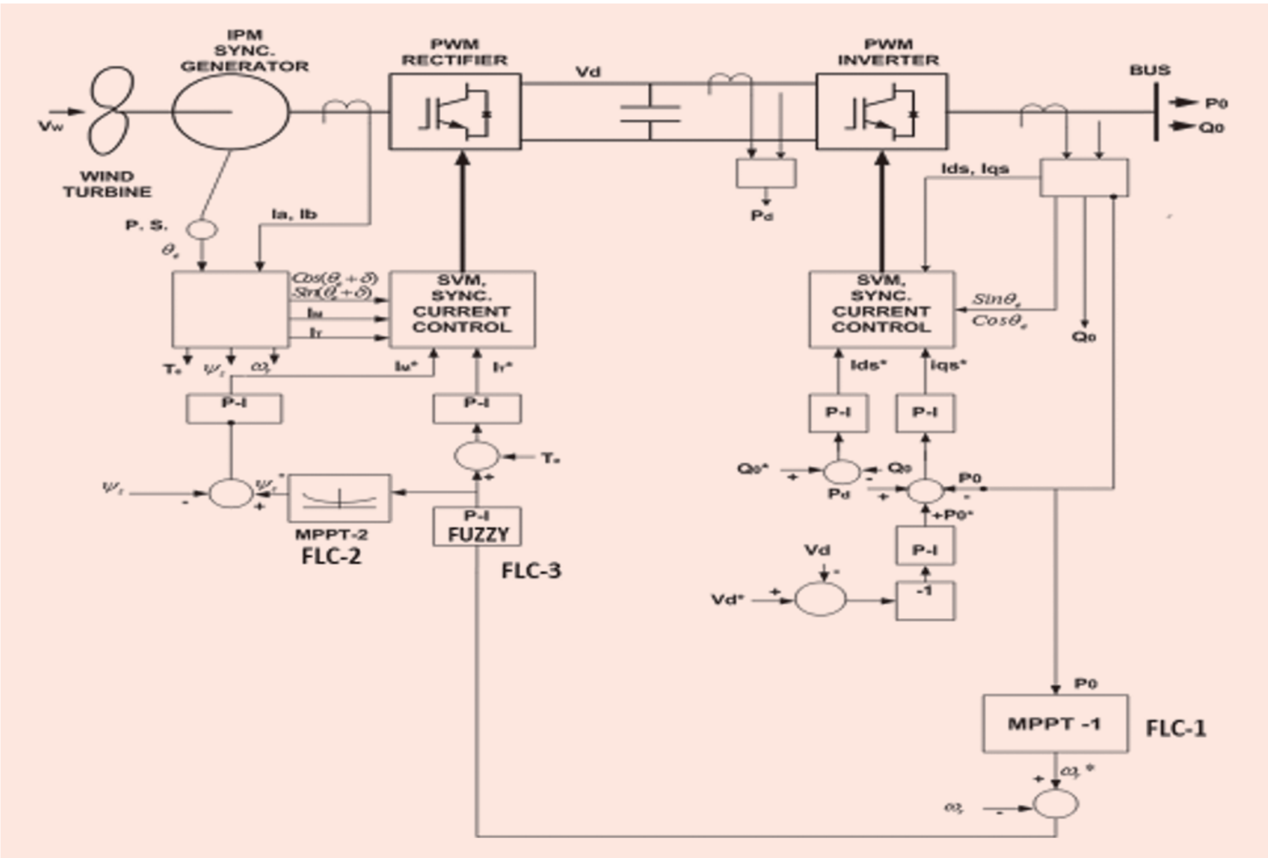

FIG 4 Fuzzy control of modern wind generation system with IPM synchronous generator.

(FLC-3) gives robust and deadbeat speed control for wind vortex and pulsating turbine torque. The fuzzy control has the advantages that it is adaptive in nature giving fast convergence and accurate signal information is not needed.

Neural Networks

Among all the AI techniques, NNWs are most important, and in fact, modern AI technology is synonymous with NNW techniques and their applications. It is a generic form of AI, and therefore more powerful. The invention of NNW is often considered as significant as the invention of transistor. A NNW is basically interconnection of artificial neurons [1], [4], [7], [8] that emulate the characteristics of biological neuron in our brain nervous system as indicated in Figure 6(a). The model of artificial neuron that closely matches with biological neuron is shown in Figure 6(b). Basically, it has op-amp summer-like structure. The input signals X_1, X_2, etc. which may be continuous variables or discrete pulses, flow through a gain or weight (called synaptic weight or connection strength) that can be positive or negative, integer or noninteger.

The summing node accumulates all the input-weighted signals, adds to the weighted bias signal b and passes to the output through the nonlinear (or linear) activation or transfer function (TF), as shown in the figure. The activation function may be linear bipolar, threshold, signum, Gaussian, sigmoidal (or log-sigmoid), or hyperbolic-tan (or tan-sigmoid) [1]. The magnitude of these functions varies between 0 and 1, or −1 to +1. The nonlinearity of TF gives nonlinear input-output mapping property of NNW. The NNW can have many feedforward and feedback (called recurrent) topologies, but the most commonly used feedforward topology for applications in power electronics is shown in Figure 7.

Each circle of the NNW represents a neuron with its TF and the dot represents the weight for interconnection between the neurons. The external bias source generally connects to all the neurons, but is shown only for the middle layer for simplicity. The network has three layers of neurons: input layer, hidden layer and output layer, with three, five and two neurons, respectively. Often, it is called 3-5-2 NNW. The input signals (logical, continuous or discrete, unidirectional or bidirectional) are processed through the NNW and appear at the output. Usually, the signals are processed in normalized form with input scaling and output descaling, as indicated. The network has associated memory or pattern recognition property due to the distributed intelligence of the neurons contributed by the synaptic weights of the neurons. This means that when one signal pattern is impressed at the input a desired pattern appears at the output. This is basically modeling of human intelligence with the help of NNW. This modeling is possible if the NNW is trained adequately with example input-output data patterns (called supervised training) through a training software. Figure 7 indicates the commonly used training algorithm, called backpropagation training. Basically, for a given input data pattern, the neuron weights are adjusted in the backward

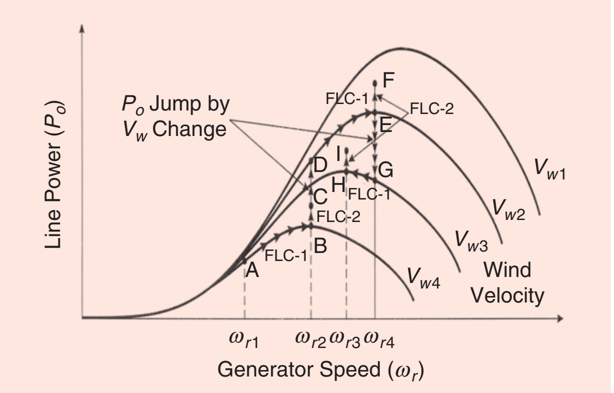

FIG 5 Output power maximization of wind generation system with fuzzy MPPT-1 (or FLC-1) and MPPT-2 (or FLC-2) controls.

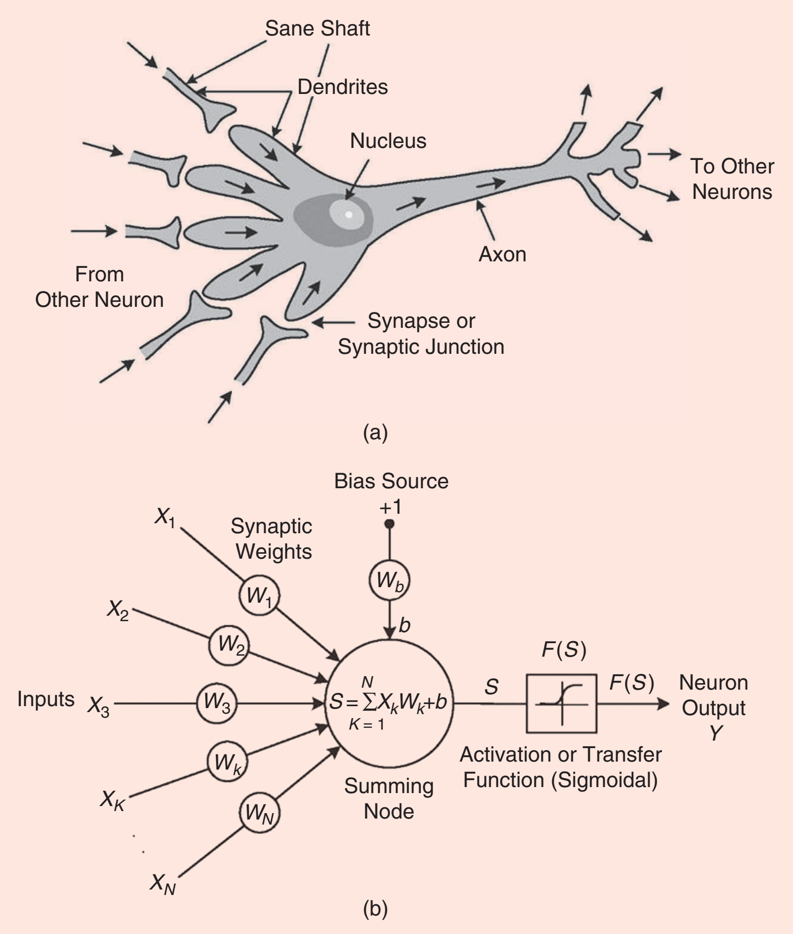

FIG 6 (a) Biological neuron, (b) artificial neuron.

direction so that the output data pattern matches with the desired output pattern (pattern matching or pattern recognition) [9].

Neural Network based Fault Pattern Identification of Smart Grid Subsystem

The applications of NNW in power electronics include PWM (SHE and SVM) implementation, multi-dimensional look-up table generation, waveform processing, delayless filtering, estimation for distorted waves, FFT analysis, adaptive control of drives, vector control, feedback signal estimation of drives, on-line system diagnostics, fault-tolerant control, neuro-fuzzy based vector and DTC control, system health monitoring, etc. We will discuss here briefly the technique of fault identification of smart grid subsystem by neural mapping technique [4].

This application is concerned with identification of fault and power quality problems in a smart grid subsystem (or microgrid) as indicted in Figure 8. Three-phase voltage and current wave signals of a power system give the footprint of different types of faults in a system. The voltage and current wave signals are sensed and their space vectors are computed. These are then analyzed to identify different types of faults in the system. For example, in a healthy harmonic-free balanced system with the rated sinusoidal voltage and current waves, the space vectors are circles of fixed diameters. For overvoltage and overcurrent conditions, the respective circle diameters will have proportional increase. Similarly, for decrease in voltage and current, the respective circle diameter will decrease proportionally. Therefore, the quantitative deviations from the rated values can be determined by measurement of the circle diameter. For system imbalance or harmonics in the waves, the space vectors will be distorted characteristically. For every abnormal condition of the system, the corresponding space vectors will have some distortion, the signature of which can be analyzed to identify the fault or any abnormal condition quantitively. The space vector patterns are displayed on optical grids with $m \times n$ matrix, and the characteristic logic signals 1's (shaded) and 0's (unshaded) can be generated similar to optical character recognition (OCR) system used by NNW [10].These matrices of the characteristic logic signals are impressed at the input of a feedforward neural network, as shown in Figure 8. The NNW is initially trained with example data patterns by backpropagation technique to identify the corresponding fault or any abnormal condition in the system. Some of the faults are indicated in the figure. Based on these signals, the network can generate warning signals, be protected by selective tripping or fault-tolerant system control can be designed.

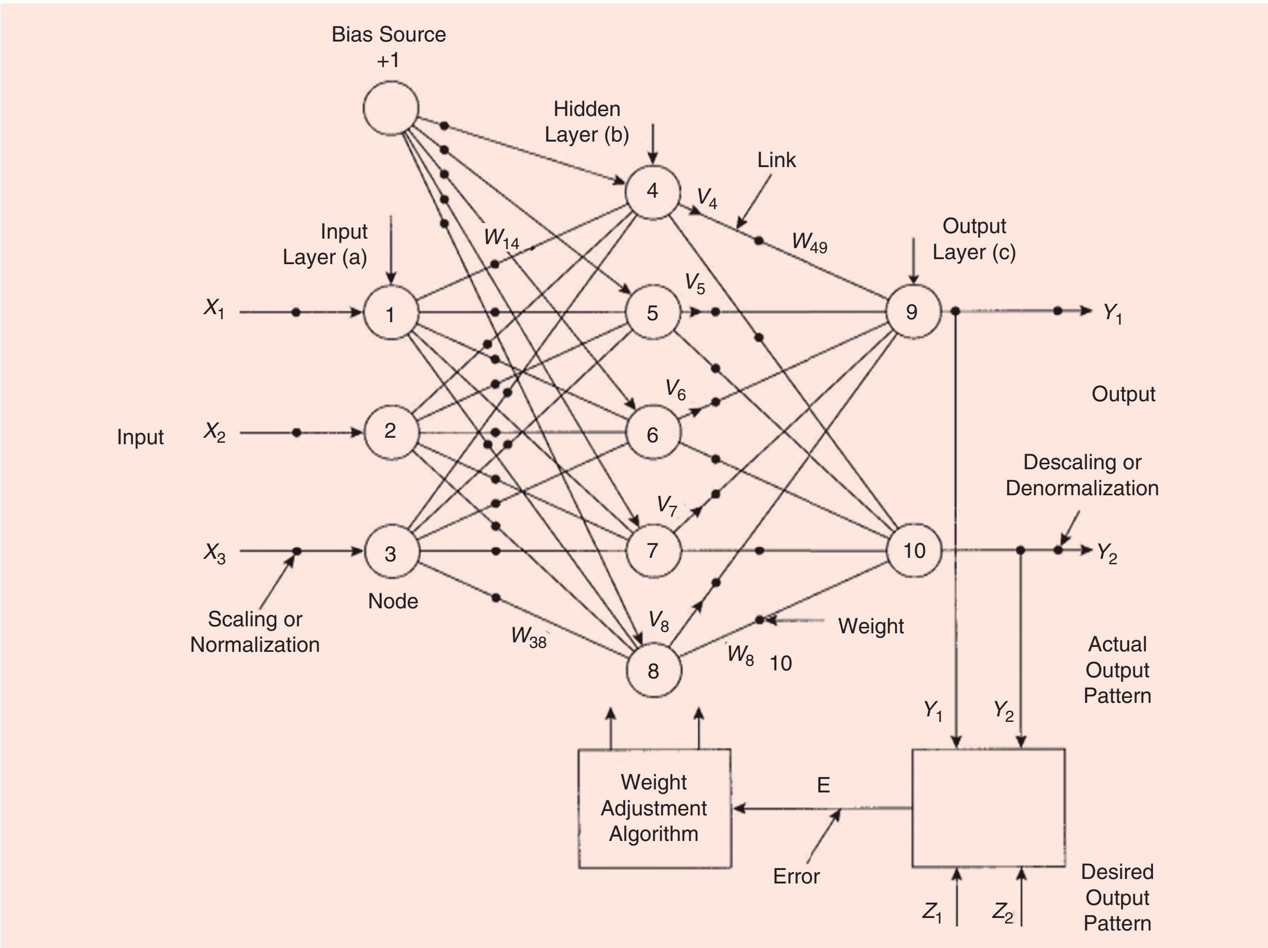

FIG 7 Feedforward NNW structure (3-5-2) showing backpropagation training.

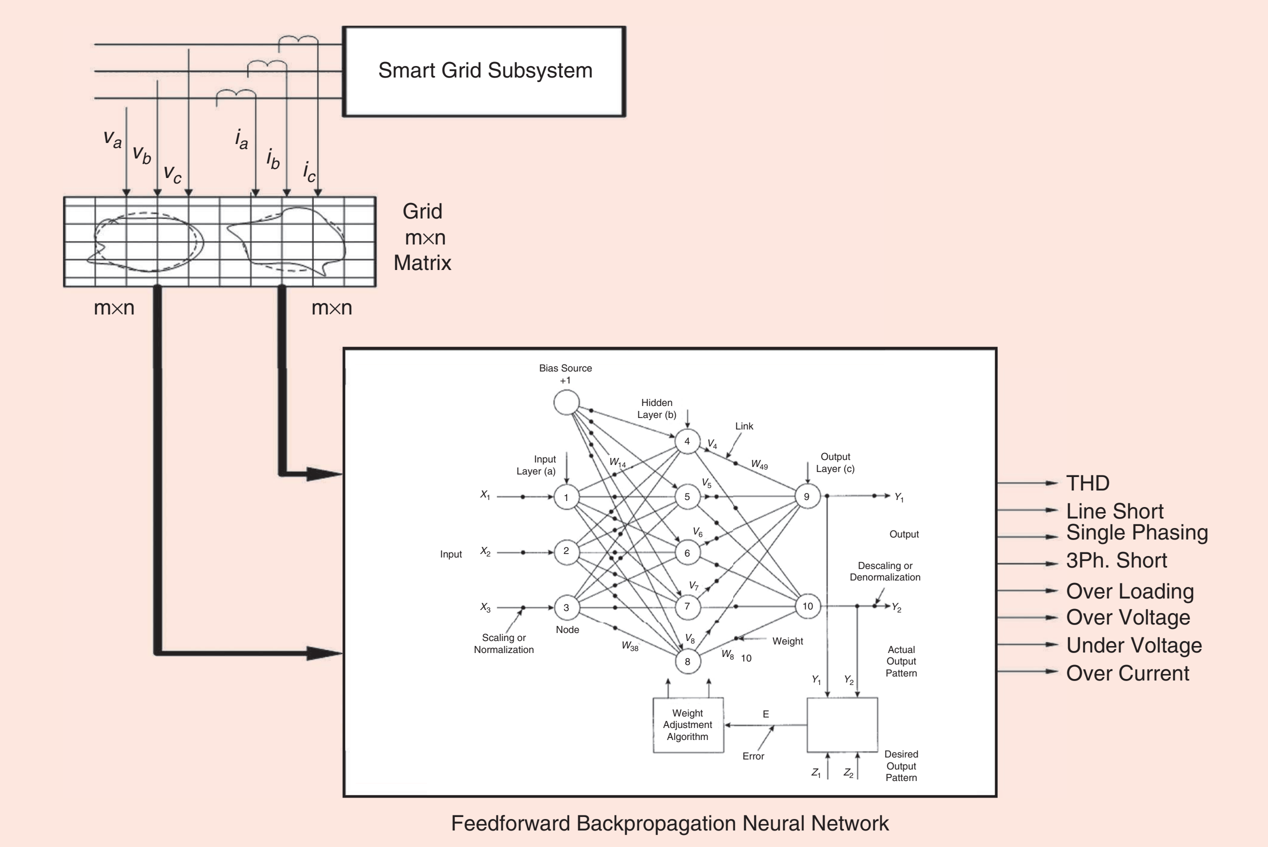

FIG 8 Fault and abnormal condition identification of micro grid using NNW-based mapping.

Summary

The advent of powerful AI techniques has opened a new frontier for advanced control and protection of power electronics/power systems. In this article, a brief but comprehensive review of different elements of AI are given. An example application in each area of AI is given briefly. Selecting a particular AI technique depends on the problem complexity. The ES can be simple software if the problem is simple. However, it can be complex for a complex problem, such as smart grid control. Similarly, FL applications can be simple or complex and can be solved either by fuzzy software or NNW technique. NNW is the most generic AI technique and is used for solving complex nonlinear mapping problems.

About the Author

Bimal K. Bose received the M.S. degree from the University of Wisconsin, Madison, USA and Ph. D. degree from Indian Institute of Engineering Science and Technology, India. Currently, he is Emeritus Professor of Electrical Engineering in the University of Tennessee, Knoxville, United States. From 1987 to 2003, he was the Endowed Chair Professor in the same university. From 1976 to 1987, he was a research engineer in GE-CRD, Schenectady, NY, and from 1971 to 1976, he was a faculty member in Rensselaer Polytechnic University, Troy, NY, United States. He is a seven times winner of IEEE awards. He is also a Member of US National Academy of Engineering.

References

[1] B. K. Bose, *Modern Power Electronics and AC Drives*. Upper Saddle River, NJ: Prentice Hall, 2001.

[2] C. Daoshen and B. K. Bose, "Expert system based automated selection of industrial AC drives," IEEE IAS Annu. Conf. Rec., 1992, pp. 387–392.

[3] R. G. Vedder, "PC based expert system shells: Some desirable and less desirable characteristics," *Expert Syst.*, vol. 6, pp. 28–42, Feb.1989. doi: 10.1111/j.1468-0394.1989.tb00074.x.

[4] B. K. Bose. "Artificial intelligence techniques in smart grid and renewable energy systems: Some example applications," *Proc. IEEE (Special Issue on Power Electronics in Smart Grid and Renewable Energy Systems, Guest Edited by B. K. Bose)*, vol. 105, no. 11, pp. 2262–2273, Nov. 2017. doi: 10.1109/JPROC.2017.2756596.

[5] *Fuzzy Logic Toolbox User's Guide*, Math Works Inc., April 2003.

[6] B. K. Bose, "A high performance inverter-fed drive system of an interior permanent magnet synchronous machine," *IEEE Trans. Ind. Appl.*, vol. 24, pp. 989-997, Nov./Dec. 1988.

[7] B. K. Bose, "Artificial intelligence techniques in power electronics and motor drives," ECCE 2011 Tutorial, IEEE Power Electronics and Industry Applications Societies, Phoenix, Arizona, Sept. 17–22, 2011, pp. 367–400.

[8] B. K. Bose, "Neural network applications in power electronics and motor drives: An introduction and perspective (Invited Paper)," *IEEE Trans. Ind. Ind. Electron.*, vol. 54, no.1, pp. 14–33, Feb. 2007. doi: 10.1109/TIE.2006.888683.

[9] *Neural Networks Toolbox User's Guide*, Math Works Inc., Oct. 2004.

[10] L. H. Tsoukalas and R. E. Uhrig, *Fuzzy and Neural Approaches in Engineering*, New York: Wiley, 1997.