ARTIFICIAL NEURAL NETWORK APPLICATIONS IN POWER ELECTRONICS

Dr. Bimal K. Bose, Life Fellow, IEEE
Condra Chair of Excellence in Power Electronics
The University of Tennessee
Knoxville, TN 37996-2100, USA
E-mail: bbose@utk.edu

Abstract: Artificial neural network (ANN) or neural network is an emerging technology that is advancing rapidly and having impact on many scientific and engineering applications. The frontier of power electronics and motor drives has recently been advanced considerably by the advent of this technology, and the future indicates a tremendous promise. Neural network has brought challenge to the expertise of power electronic engineers with traditional background. This paper will discuss principles of neural networks and describe some of their applications in power electronics and motor drives. The specific applications will include static feedback signals estimation for a vector drive, space vector PWM for a two-level voltage-fed inverter and voltage model flux vector estimation. Finally, some problems and challenges for practical applications will be discussed.

I. INTRODUCTION

Neural network belongs to the area of artificial intelligence (AI) like expert system (ES), fuzzy logic (FL) and genetic algorithm (GA), but it is a more generic form of AI compared to behavioral rule-based ES and FL systems. The ANN, FL and GA belong to the area of ‘soft computing’ where approximate computation is required, compared to ‘hard computing’ or precise computation required in many applications. Broadly, the goal of AI is to imitate natural or human intelligence for emulation of human thinking. The natural intelligence is contributed by biological neural network that gives the human beings the capability to memorize, store knowledge, perceive, learn and take intelligent decision. The ANN tends to emulate the biological neural network. When a child is born, its cerebral cortex is said to contain around 100 billion nerve cells or biological neurons which are interconnected to form the biological neural network. As the child grows in age, his network gets trained to give him all the expertise in adult life. The neurologists and behavioral scientists hardly understand the structure and functioning of the brain in spite of painstaking research for a long period of time. Therefore, the ANN model of biological network is far from accurate. However inferior is the model, ANN (also called neurocomputer or connectionist system in literature) tends to solve many important problems in science and engineering. It can be shown that ANN is particularly suitable for solving pattern recognition (or input-output mapping) and image processing type problems that are difficult to solve by traditional digital computer. This mapping capability of ANN is analogous to the associative memory property of human brain. This property helps us to remember or associate the name of a person when we see his face, or recognize an apple by its color and shape. Just like a human brain, an ANN is trained (not programmed like a digital computer) to learn by example input-output associating patterns. This is like teaching alphabet characters to a child by a teacher (defined as supervised learning) where the characters are shown and their names are pronounced repeatedly. Broadly, ANN technology has been applied in process control, identification, forecasting, diagnostics, robot vision, just to name a few. More specifically in power electronics area, the ANN applications include look-up table functions (one or multi-dimensional), PWM techniques, adaptive P-I close loop control, delayless harmonic filtering, model referencing adaptive control (MRAC), optimization control, vector control, drive feedback signal processing, on-line diagnostics, FFT signature analysis, estimation for distorted waves, etc. [1]. Within the length constraint of this paper, only a few applications will be discussed.

II. NEURAL NETWORK STRUCTURE

A. Biological and Artificial Neurons
The structure of an artificial neuron shown in Fig.1(b) in a neural network is inspired by the concept of biological neuron shown in (a) of the same figure. Basically, neuron is a processing element (PE) in brain nervous system that receives and combines signals from other similar neurons through thousands of input paths called dendrites. If the combined signal is strong enough, the neuron "fires", producing an output signal along the axon that connects to dendrites of thousands of other neurons. Each input signal coming along a dendrite passes through a synapse or synaptic junction. The junction is an infinitesimal gap in the dendrite that is filled with neurotransmitter fluid which either accelerates or retards the flow of electrical charges. These electrical signals flow to the nucleus or soma of the neuron. The adjustment of the impedance or conductance of the synaptic gap by the neurotransmitter fluid contributes to "memory" or intelligence of the brain. According to this theory of neuron, we are led to believe that the brain has distributed memory or intelligence characteristics giving it the property of associative memory. An artificial neuron has direct analogy with the biological neuron. Basically, it is analog summer-like network where the input signals (continuous variables or discrete pulses) pass through synaptic weights (or gains) and collect in the summing node before passing through a nonlinear activation or transfer function at the output. A bias signal (not shown) can be added in the summing node.

Fig. 2 shows a few possible activation functions. The simplest of all is the linear activation function (not shown) that can be unipolar or bipolar and saturate at 0 and 1 (unipolar) or -1 and +1 (bipolar), respectively, with large signal amplitude. The threshold and signum functions can be added with a bias signal and the outputs vary abruptly between 0 to +1 and -1 to +1, respectively, as shown. The most commonly used activation functions are sigmoidal (also called log-sigmoid) and hyperbolic tan (also called tan-sigmoid) types which are nonlinear, differentiable and continuously varying types between two asymptotic values 0, 1, and -1, +1, respectively.

Fig. 1(a) Biological neuron
(b) Artificial neuron

Fig. 2. Artificial neuron activation functions

B. Feedforward Neural Network

The interconnection of artificial neurons results artificial neural network (ANN). The structure of biological neural network is not well-understood, but the scientists have come up with large number of ANN models and many more are yet to come. The ANNs can generally be classified as
feedforward and feedback (or recurrent) types. In a feedforward network, the signals from neuron to neuron flow only in the forward direction, whereas in a recurrent network, the signals flow in forward as well as backward or lateral direction. The examples of feedforward network are Perceptron, Adaline and Madaline, Back Propagation network, Radial basis function network (RBFN), General regression network, Modular neural network (MNN), Learning vector quantization (LVQ) network, Probabilistic neural network (PNN) and Fuzzy neural network (FNN). The examples of recurrent neural network (RNN) are Hopfield network, Boltzmann machine, Kohonen’s self-organizing feature map (SOFM), Recirculation network and Brain-state-in-a-box (BSB), Adaptive resonance theory (ART) network and Bi-directional associative memory (BAM).

Fig. 3 shows the structure of most commonly used feedforward multilayer back propagation type network. The name ‘back propagation’ comes from its characteristic training method. Often, it is called multi-layer Perceptron (MLP) type network. The three-layer network shown has three input signals, two output signals and a hidden layer. The circles represent the neurons, and interconnections between the neurons are shown with weights represented as dots in the links. As mentioned before, for unipolar output, sigmoidal activation function and for bipolar output, hyperbolic tan activation functions are normally used. Sigmoidal outputs can be clamped to generate logical outputs whereas hyperbolic tan outputs can be clamped to generate $\pm 1$ output in each channel. The input layer neurons have usually linear activation functions with unity gain (or no activation function at all), but there is a scale factor with each input to convert to per-unit (normalization) signals. Similarly, the output signals are converted from per-unit signals to actual signals by denormalization. A constant bias source (not shown) is often coupled to hidden and output layers through weights. The architecture of the network indicates that it is a fast and massive parallel-input parallel-output computing system where computation is done in a distributed manner. The network can perform input-output static nonlinear mapping, i.e., for a signal pattern input, a corresponding pattern can be retrieved at the output. The pattern matching is possible due to associative memory property contributed by so many distributed weights. A large number of input-output example data patterns are required to train the network. Usually, a computer program (such as, MATLAB Neural Network Toolbox by Mathworks) is used for such training. Once it is trained successfully, it can not only recall the example input-output patterns, but can also interpolate or extrapolate them. The trained ANN can be implemented in parallel by an ASIC chip or serially by a high-speed DSP. An ANN is said to have fault tolerance property because deterioration of a few weights or a few missing links does not affect the input-output pattern matching property. The three-layer network shown in Fig. 3 is often called universal function approximator because it can convert any continuous function at the input to any desired function at the output.

C. Recurrent Neural Network

In many applications, an ANN is required to be dynamic, i.e., it should be able to emulate a nonlinear (or linear) dynamical system with temporal behavior, such as identification of a machine model or estimation of flux by voltage or current model. Dynamic ANNs can be classified as recurrent neural network (RNN) and time delayed neural network (TDNN). Fig. 4 shows a general structure of two-layer RNN where the output signals are fed back to the input layer with a time delay. The network can emulate a second-order system like the response of a R-L-C network. With step input, for example, the response will reverberate in time domain until a steady state condition is reached. The particular RNN shown is defined as real-time recurrent network [4]. Generally, training of RNN is more difficult than feedforward static network. The detailed discussion on ANN, particularly on RNN and TDNN, can be found in references [1] and [4].
III. APPLICATIONS IN POWER ELECTRONIC SYSTEMS

ANN can be applied for various control, identification and estimation applications. Some applications were mentioned at the end on Introduction. In this section, we will briefly discuss some example applications from the literature.

A. Vector Control Static Feedback Signals Estimation

A simple example of feedforward ANN application is the feedback signal computation for a vector-controlled induction motor drive [5]. Fig. 5 shows the block diagram of a direct vector-controlled induction motor drive where the ANN estimates the rotor flux ($\psi_r$), unit vector ($\cos \theta_r$, $\sin \theta_r$) and torque ($T_e$) from the input fluxes ($\psi_{dr}^s$, $\psi_{qr}^s$) and currents ($i_{ds}^s$, $i_{qs}^s$) by solving the follow set of non-dynamical equations:

$$\psi_{dr}^s = \psi_{ds}^s - i_{ds}^s L_{ds}$$  \hspace{1cm} (1)
$$\psi_{qr}^s = \psi_{qs}^s - i_{qs}^s L_{ds}$$  \hspace{1cm} (2)
$$\psi_{dr}^s = \frac{L_{ds}}{L_m} \psi_{ds}^s - L_{ds} i_{ds}^s$$  \hspace{1cm} (3)
$$\psi_{qr}^s = \frac{L_{ds}}{L_m} \psi_{qs}^s - L_{qs} i_{qs}^s$$  \hspace{1cm} (4)
$$\hat{\psi}_r = \sqrt{\psi_{dr}^s + \psi_{qr}^s}$$  \hspace{1cm} (5)
$$\cos \theta_r = \frac{\psi_{dr}^s}{\psi_r}$$  \hspace{1cm} (6)
$$\sin \theta_r = \frac{\psi_{qr}^s}{\psi_r}$$  \hspace{1cm} (7)
$$T_e = \frac{3}{2} \left( \frac{P}{2} \left( \psi_{dr}^s i_{qs}^s - \psi_{qr}^s i_{dr}^s \right) \right)$$  \hspace{1cm} (8)

A DSP-based estimator is also shown in the figure for comparison. Since the feedforward network can not solve any dynamical system, the machine terminal voltages are integrated by a hardware low-pass filter (LPF) to generate the stator flux signals $\psi_{dr}^s$ and $\psi_{qr}^s$ as shown. Voltage integration for flux synthesis by using RNN will be discussed later. The variable frequency variable magnitude sinusoidal signals are then used to calculate the output parameters by a feedforward ANN as shown in Fig. 6. The topology of the network has three layers and

**Fig. 5. Block diagram for direct vector control**

**Fig. 6. ANN based static feedback signal estimation**
the hidden layer contains 20 neurons (4-20-4 network). The hidden and output layers have hyperbolic tan activation function to produce bipolar output signals. The network was trained with the simulated input-output data of an equivalent DSP. The ANN based estimator gives accurate performance and some amount of harmonic immunity. Noise or harmonic filtering is one of the advantages of ANN.

B. Space Vector PWM Controller

![Space Vector PWM Trajectories](image)

**Fig. 7.** Space vector PWM trajectories (a) Undermodulation (b) Overmodulation-Mode 1 (c) Overmodulation-Mode 2

undermodulation, overmodulation-mode 1 and overmodulation-mode 2, respectively. The undermodulation or linear region ends when the circular trajectory describes the inscribed circle in the hexagon. In mode 1, there is a partial loss of output voltage because the command voltage exceeds the hexagon boundary. In order to compensate this loss and attain linearized voltage transfer relation, a circular trajectory of higher radius, as shown, is considered. Mode 1 ends when the trajectory moves along the hexagon sides. In mode 2, the trajectory consists of partial movement on the hexagon sides and partial hold conditions along the active inverter vectors. Finally, in square-wave mode, the voltage vectors jump after π/3 hold angle (α) in each of the six active states. In order to describe three-phase symmetrical pulse patterns at the output with linear voltage transfer relation, all the three modes in SVM can be analyzed mathematically and turn-on time (T_{on}) (see Fig. 10) expressions for the three phases as function of angle θ and voltage V can be derived. Fig. 8 shows the plot of turn-on time for phase a (T_{on}) with angle θ in all the six sectors. The curves for phases b and c are identical except they are mutually phase-shifted by 2π/3 angle. Note that the curve amplitude varies linearly with the V magnitude in the undermodulation region, but saturates with higher voltage in modes 1 and 2. This function shows analogy with the modulating wave of sinusoidal PWM where triplen harmonics are mixed with the signal. Fig. 9 will give actual

![Turn-on Time Function](image)

**Fig. 8.** Turn-on time function for phase a turn-on time in all the modes if the unit amplitude function is multiplied by the voltage function f(V) shown in Fig. 9. This precomputed curve for a dc link voltage (V_d) of 300 V indicates linear relation (f(V) = V) in undermodulation region but highly
nonlinear relation in overmodulation regions. An ANN can now be designed and trained by the information given in Figs. 8 and 9. Fig. 10 shows the ANN topology including the single timer section at the output. There are essentially two ANN subnets: one is the voltage amplitude subnet and the other is the angle subnet. The amplitude subnet receives the voltage magnitude as the input and solves for the function \( f(V') \) shown in Fig. 9. The angle subnet receives the angle \( \theta_a^{\pi} \) at the input and solves the unit amplitude functions \( g_d(\alpha^*) \) and \( g_c(\alpha^*) \) for the three phases at the output. These signals are multiplied by \( f(V') \) and added with the bias signal \( WT/4 \) to generate the corresponding digital word for the three phases given by the general expression

\[
WT_{on} = \frac{WT}{4} + f(V')g(\alpha^*)
\]

Fig. 9. \( f(V') \) – \( V' \) relation

\[\text{Fig. 10. ANN topology for SVM}\]

A single UP/DOWN counter at the output helps to generate the symmetrical pulse widths for all the three phases.

**C. Voltage Model Flux Vector Estimation**

Programmable cascaded low-pass filter (PCLPF) method of integration of machine terminal voltages for estimation of flux vectors has been discussed in literature [7]. Basically, PCLPF permits offset-free integration of machine voltage down to very low frequency (fraction of a Hz.) and permits direct vector control to be used at very low speed. Fig. 11 shows two-stage PCLPF for estimation of stator fluxes \( \psi_{ds} \) and \( \psi_{qs} \). Basically, PCLPF can replace the LPF integrator shown in Fig. 5. Each voltage signal is filtered by a hardware LPF and stator resistance drop is subtracted as shown before passing through two identical stages of first-order programmable low-pass filter (PLPF). The time constant \( \tau \) of the filter is function of frequency \( \omega_0 \) so as to give predetermined phase shift angle – \( 0.5(\pi/2-\phi_0) \) \[ \phi_0 = \text{hardware LPF phase shift angle} \].

If \( \phi_0 = 0 \), the phase shift angle is \(-\pi/4\) irrespective of frequency. For ideal integration, an amplitude compensation as function of frequency is also required. Each flux estimation channel is identical.
and can be represented by a second-order nonlinear dynamical system.

\[
\begin{bmatrix}
  y_1(k+1) \\
  y_2(k+1)
\end{bmatrix} =
\begin{bmatrix}
  W_{11} & 0 \\
  W_{21} & W_{22}
\end{bmatrix}
\begin{bmatrix}
  y_1(k) \\
  y_2(k)
\end{bmatrix}
+ \begin{bmatrix}
  W_{13} \\
  0
\end{bmatrix} u(k)
\]  

... (10)

where \( u = v_{ds} \) is the input and \( y_1 = \psi_{ds} \) is the output for \( \psi_{ds} \) estimation. Fig. 12 (upper part) shows the ANN based flux estimation for both the channels that satisfies the above general equation. The ANN uses linear activation function but the weights are function of frequency as shown in the right part of the figure. The ANN is trained by Extended Kalman Filter (EKF) algorithm for each set of input voltage wave and the corresponding output flux wave at a particular frequency.

**D. ANN Based Stator Flux Oriented Vector Drive**

Fig. 13 shows a simplified block diagram of a stator flux oriented direct vector controlled drive that incorporates the ANN based SVM controller and PCLPF integrator. The signal estimation block

Fig. 12. Hybrid ANN for PCLPF

consists of the following set of static equations (similar to that in Fig. 5):

\[
\dot{\psi}_s = \sqrt{(\psi_{ds})^2 + (\psi_{qs})^2}
\]  

... (11)

\[
\theta_e = \sin^{-1} \left( \frac{\psi_{qs}}{\psi_s} \right)
\]  

... (12)

\[
i_{ds} = i_{qs} \cos \theta_e - i_{ds} \sin \theta_e
\]  

... (13)

\[
i_{qs} = i_{qs} \sin \theta_e + i_{ds} \cos \theta_e
\]  

... (14)

\[
i_{ds} = \frac{\sigma L_i i_{qs}}{\psi_{ds} \sigma L_i i_{ds}}
\]  

... (15)

\[
i_{qs} = \psi_{qs} \cos \theta_e - \psi_{ds} \sin \theta_e
\]  

... (16)

\[
i_{qs} = \psi_{qs} \sin \theta_e + \psi_{ds} \cos \theta_e
\]  

... (17)

\[
T_e = \frac{3P}{4} [\psi_{ds} i_{qs} - \psi_{qs} i_{ds}]
\]  

... (18)

\[
\omega = \left( \psi_{qs} - i_{qs} R_s \right) \psi_{ds} - \left( \psi_{ds} - i_{ds} R_s \right) \psi_{qs}
\]  

... (19)

These equations can be easily implemented by a feedforward ANN. The drive system is intended for EV drive but it can not start with vector control at zero frequency. A finite low frequency (fraction...
of a Hz.) operation of the drive is required for the integrator. The stator flux orientation has the advantage of minimal parameter variation problem, but it introduces some amount of coupling effect which is compensated by a decoupler.

IV. CONCLUSION AND DISCUSSION

The paper has discussed neural network principles and a few selected applications related to vector-controlled induction motor drive. For real time applications, a neural network can be implemented either by a DSP or by an ASIC chip. The ASIC chip implementation is desirable because of fast parallel computation capability. Unfortunately, the nonavailability of large and economical ASIC chips is hindering applications of ANN. Intel introduced electrically trainable analog neural network (ETANN type 80170NX) but withdrew from the market because of drift problem. Large digital ASIC chips with high resolution are not yet available. ANN requires large data sets for training and the training is time-consuming. Currently, a lot of research is going on to solve these problems. Fortunately, the computer speed is also increasing to alleviate this problem. For a parameter varying system such as a drive, the ANN weights are to be adaptive by fast on-line training. On-line training has been attempted by various methods that include random weight change and EKF algorithms. As mentioned before, most of the current ANN applications are restricted to feedforward back propagation type network. There is a tremendous opportunity of expanding the applications by other feedforward and feedback networks. Hybrid AI techniques where ANN can be combined with ES, FL and GA can explore many new areas of applications. In power electronics and motor drives area, the ANN technology offers significant promise for the future.

REFERENCES

[2] B.K.Boke, “Neural network applications in power electronics and drives”, Distinguished Lecture in Texas A&M University, College Station, Texas, November 12, 2001