# Neural Network Controller for Asymmetric Cascaded Multilevel Inverter

Bambang Sujanarko
Dept. of Elect. Eng., Universitas Jember, currently toward Doctor in Institut Teknologi Sepuluh Nopember (ITS) Surabaya, Indonesia

Mochamad Ashari
Dept.of Elect. Eng., Institut Teknologi Sepuluh Nopember (ITS) Surabaya, Indonesia

Mauridhi Hery Purnomo
Dept.of Elect. Eng., Institut Teknologi Sepuluh Nopember (ITS) Surabaya, Indonesia

## ABSTRACT

Among the Artificial Intelligent (AI) technique, Neural Network (NN) is emerging technology that is advancing rapidly and having impact on many scientific and engineering applications. In this paper, NN apply on Asymmetric Cascaded Multilevel Inverter (ACMLI), in order to improve the controller easier to construct, simpler and better performance. To verify these goals, the system simulated based on Matlab Simulink after the NN controller build using m-file program. The simulation results show that the goals can realized.

## Keywords

Neural network, Asymmetric Cascaded Multilevel Inverter, performance

## 1. INTRODUCTION

NN belongs to the area of AI like expert systems (ES), fuzzy logic (FL) and genetic algorithm (GA), but it is a more generic form of emulated human thinking that capability to memorize, store knowledge, perceive, learn and take intelligent decision [4]-[12]. The NN structure consists of artificial neurons, which each order interconnected. Because these capability to emulate human thinking, hence NN will be used to overcome the problems in many scientific and engineering applications.

In this paper, NN use to improve ACMLI controller especially to make the controller easier to design, simpler, and better performance if compared to other control of ACMLI, as electronic circuits controller, micro-controller, Field Programmable Gate Array (FPGA), or computer program within certain language [1],[4-5],[8],[10,[13]. NN will be applied for ACMLI within binary, trinary and sine quantization of DC voltage progression. This ACMLI within these DC voltage progressions selected, because its have better performance than others [2],[7].

This application built using the Matlab toolbox, where the NN structure easy construct using ***gensim(net)*** instruction in the m-file, and the system simulation so easy to construct using the Simulink model. The parameter input for training this NN determine from reference signals such as DC voltages, sine reference, and trigger signals that determined from certain algorithm as explained in the Ref. [3]. To verify the performance, a comparative study used, among the proposed controller to the other controllers.

## 2. SYSTEM DESIGN

## 2.1 NN System

### *2.1.1 Biological and Artificial Neurons*

The structure of biological neuron is shown in Fig. 1. Basically, it is the processing element in the nervous system of the brain that receives and combines signals from other similar neurons through thousands of input paths called dendrites [4],[6]. Each input signal flowing through dendrite passes through a synapse or synaptic junction.

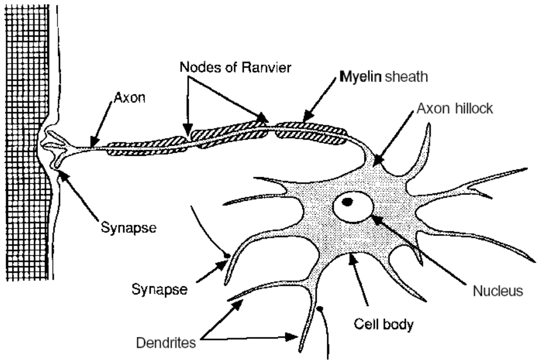

Fig. 1. Structure of Biological neuron

The junction is an infinitesimal gap in the dendrite, which is filled with neurotransmitter fluid that either accelerates or retards the flow of the signal. These signals are accumulated in the nucleus (or soma), nonlinearly modified at the output before flowing to other neurons through the branches of axon. The adjustment of the impedance or conductance of the synaptic gap by the neurotransmitter fluid contributes to the memory or intelligence of the brain.

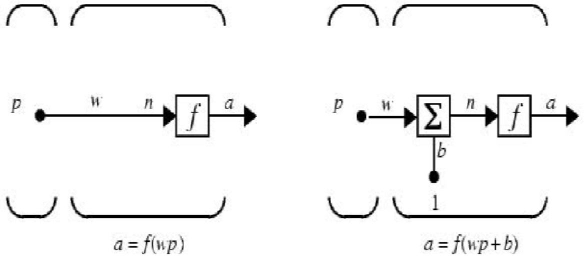

Fig. 2. Structure of neuron (artificial neuron)

The model of a neuron (artificial neuron) is shown in Fig. 2 [4],[6]. It has op-amp summer-like structure. The neuron has some input signal (p), which flows through a gain or weight (w), which can be positive or negative, integer or non-integer. The summing node accumulates all the input-weighted signals, adds to the weighted bias signal (b) (if it's used), and then passes to the output through the nonlinear (or linear) activation or transfer function (TF).

### *2.1.2 Transfer Function*

Transfer Function is Function that maps a neuron's (or layer's) net output **n** to its actual output. There are many types of transfer functions [6], such as *hardlim* (hard limit), *pureline* (linear), *logsig* (log sigmoid), *satlins* (saturation linear), etc. Fig. 3 shows some of transfer function graph. Transfer function used depend on the system to be finished by NN.

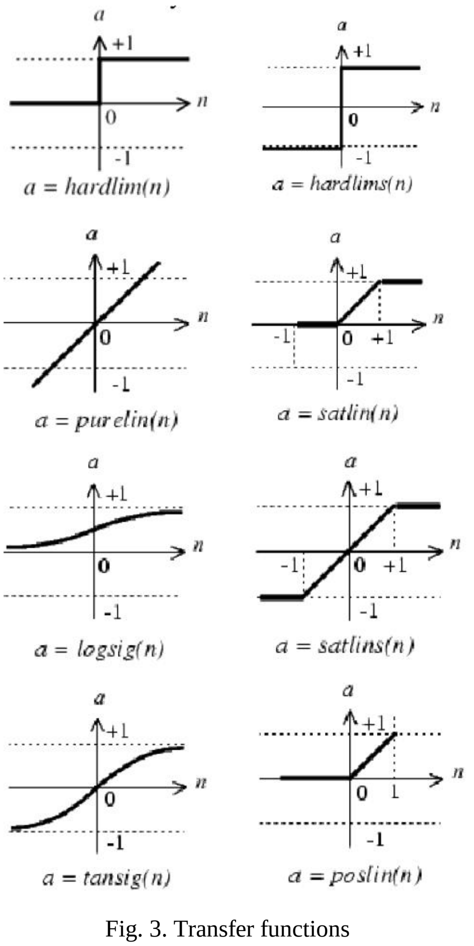

Fig. 3. Transfer functions

### *2.1.3 NN Type*

Many NN models have been proposed, such as Perceptron, Adaline and Madaline, Backpropagation (BP) Network, Radial Basis Function Network (RBFN), Modular Neural Network (MNN), Training Vector Quantization (LVQ) Network, Fuzzy Neural Network (FNN), Kohonen's Self-Organizing Feature Map (SOFM), Adaptive Resonance Theory (ART) Network, Real Time Recurrent Network, Elman Network, Hopfield Network, Boltzmann Machine, Recirculation Network, Brain-State-In-A-Box (BSB), Bi-Directional Associative Memory (BAM) Network. But generally, NN can be classified as feed forward and feedback types. In the feed forward class, the signals flow only in the forward direction, whereas in feedback types, the signals can flow in forward as well as backward or lateral direction. A network can be defined as static or dynamic, depending on whether it emulates static or dynamic system. A NN is characterized by input–output mapping property. Only a few topologies that are most used. Currently, the back propagation network is most popular [4],[6].

There are many algorithms to determine parameters (weight and bias) in neural network system. It is very difficult to know which training algorithm will be the best for a given problem. It depends on many factors, including the complexity of the problem, the number of data points in the training set, the number of weights and biases in the network, the error goal, and others [6],[12].

These algorithms are Levenberg-Marquardt (LM), BFGS Quasi-Newton (BFG), Resilient Backpropagation (RP), Scaled Conjugate Gradient (SCG), Conjugate Gradient with Powell/Beale Restarts (CGB), Fletcher-Powell Conjugate Gradient (CGF), Polak-Ribiére Conjugate Gradient (CGP), One Step Secant (OSS), Variable Training Rate Backpropagation (GDX), etc. Among these algorithms, LM algorithm is the fastest for NN training. It is over four times faster than others [6].

Other classification is learning method, there are two kinds, namely supervised learning, where there is a target output, and unsupervised learning, a method, which no output as target.

### *2.1.4 NN Layer*

To obtain correct expression among inputs and outputs, a network can have several layers. Each layer has a weight matrix W, a bias vector b, and an output vector a. Fig. 4 shows NN in multi layer configuration. The configuration that similar with this NN will be used in this paper.

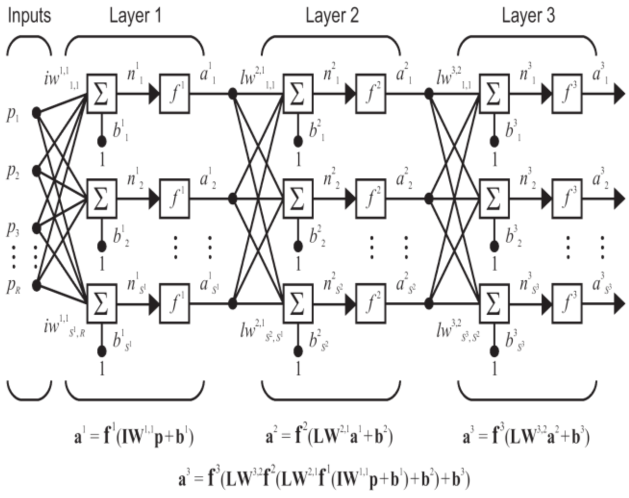
Inputs
Layer 1
Layer 2
Layer 3
$p_1$
$p_2$
$p_3$
$p_R$
$iw^{1,1}_{1,1}$
$iw^{1,1}_{s^1,R}$
$lw^{2,1}_{1,1}$
$lw^{2,1}_{s^2,s^1}$
$lw^{3,2}_{1,1}$
$lw^{3,2}_{s^3,s^2}$
$\mathbf{a}^1 = \mathbf{f}^1(\mathbf{IW}^{1,1}\mathbf{p}+\mathbf{b}^1)$
$\mathbf{a}^2 = \mathbf{f}^2(\mathbf{LW}^{2,1}\mathbf{a}^1+\mathbf{b}^2)$
$\mathbf{a}^3 = \mathbf{f}^3(\mathbf{LW}^{3,2}\mathbf{a}^2+\mathbf{b}^3)$
$\mathbf{a}^3 = \mathbf{f}^3(\mathbf{LW}^{3,2}\mathbf{f}^2(\mathbf{LW}^{2,1}\mathbf{f}^1(\mathbf{IW}^{1,1}\mathbf{p}+\mathbf{b}^1)+\mathbf{b}^2)+\mathbf{b}^3)$

Fig. 4. Multi layer NN

The layers of a multi layer network play different roles [6]. A layer that produces the network output is called an *output layer*. All other layers are called *hidden layers*. The three-layer network shown earlier has one output layer (layer 3) and two hidden layers (layer 1 and layer 2).

## 2.2 ACMLI System

### 2.2.1 ACMLI structure

Recently the multilevel inverter (MLI) is the most popular dc to ac converters for high voltage and high power in the power industry. There are three well-known topologies; diode-clamps, flying capacitor, and cascaded multilevel inverter (CMLI) [2],[5],[7]. Diode clamp and flying capacitor are MLI with common dc sources. These inverters use capacitors in series to divide the dc bus voltage in to a set of voltage levels. While CMLI consists of several H-bridge inverter units and separated DC sources. So CMLI is simply structure inverter because each H-bridge inverter has the same configuration and can eliminates the excessively large number of bulky transformers required by conventional multi pulse inverters, clamping diodes required by multilevel diode clamped inverters, and flying capacitors required by multilevel flying capacitor inverters [2].

Fig. 5 shows a single-phase CMLI schematic that consist of several H-Bridges. If in each H-Bridge the switches are S1j, S2j, S3j, S4j, where j is sequence of H-Bridges, so there are four switching combinations in each H-Bridge, but only three possibilities of output voltage that occurred that is 0, Vdcj and -Vdcj. Fig. 5 also shows that if sequence of H-Bridges (j) until N, the voltage output of CMLI is obtained by summing of the output voltages of H-bridge as equation (1).

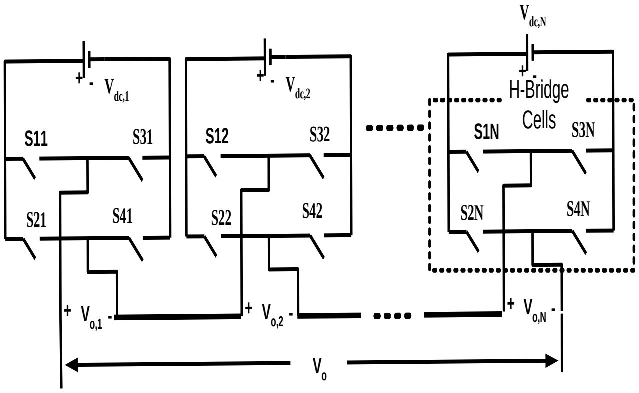

Fig. 5. Single-phase cascaded multilevel inverter

$$V_o(t) = V_{o,1}(t) + V_{o,2}(t) + ..... + V_{o,N}(t) \qquad (1)$$

Based on equation (1) and three output voltages, maximum number of switches of CMLI (s) show as equation (2). But not all of voltage levels are the effective switching. These effective of output voltages usually called voltage levels (n). The effective switches only produced if the output voltages have different amplitudes (3) and it's depend on DC voltage variety.

$$s = 3^N \qquad (2)$$

$$n \leq s \qquad (3)$$

### 2.2.2 DC Voltages Amplitude in ACMLI

CMLI can use equal DC voltages or unequal DC voltages [2],[5],[7],[11], CMLI that used unequal DC voltages also named ACMLI. Among the unequal DC voltages for ACMLI, binary progressions and trinary are the most popular used. In the binary progression the amplitude of DC voltage have ratio 1: 2: 4: 8 . . : $2^N$, where N is amount of H-Bridges in the ACMLI. Similar this progression, trinary progression has amplitude of DC voltages within ratio 1: 3: 9: 27. . : $3^N$.

If the highest voltage of H-Bridge equal to voltage rating of power devices or $V_{dc1}$ =1 pu, then $V_{dc2}$ =1/2 pu,…. $V_{dcmin}$ = $V_{dcN}$ =1/$N^2$ pu. Trinary progression, also called orde-3 [5],[14], have DC voltages $V_{dc1}$ =1 pu, then $V_{dc2}$ =1/3 pu, …. $V_{dcmin}$ = $V_{dcN}$ =1/$N^3$ pu. Using these DC voltages, the number of voltage levels and the maximum output voltage binary and trinary DC voltage progression on ACMLI are shown in (4), (5), (6) and (7).

$$n = 2^{N+1} - 1 \qquad (4)$$

$$V_{Max} = (2^N - 1)V_{dcmin}$$
$$V_{dc,j} = 2^{j-1}V_{dcmin} \quad j = 1 2,....,N \qquad (5)$$

$$n = 3^N \qquad (6)$$

$$V_{Max} = (\frac{3^N - 1}{2})V_{dcmin}$$
$$V_{dc,j} = 3^{j-1}V_{dcmin} \quad j = 1,2 \ ,....,N \quad (7)$$

The other progression is sine quantization. Each DC voltage in this ACMLI can be determined by equation (8), while the maximum output voltage calculate by equation (9). In this equation the voltage of sine wave reference is Vm, the frequency is f, the sequence number of H-Bridge is j and the number of H-Bridge is N. The process is produced by definition of quantization that is conversion of a continuous range of values into a number of discrete levels. Or in the ACMLI context, definition of sine quantization progression is a sequence of DC voltages that amplitude follows sine wave function value on discrete times.

$$\begin{aligned} V_{dc,j} &= V_m \sin(\omega t_j) \\ &= \sqrt{2}V \sin(2\pi f \frac{(1/f)}{4N+2} j) \\ &= \sqrt{2}V \sin(\pi j \frac{1}{2N+1}) \quad j = 1,2,3...,N \quad (8) \end{aligned}$$

$$V_{\max} = \sum_{1}^{N} \sqrt{2}V \sin(\pi j \frac{1}{2N+1}) \quad j = 1,2,3..., N \quad (9)$$

### 2.2.3 Harmonic Minimization

In order to eliminate some harmonic distortion or minimize the THD, ACMLI need controller to fire power electronics devices in the optimum angle. One of these methods is THD minimization [10], [14]. Basic principle of this method shows in fig. 6. To produce output voltage similar to pure sine, the area in the upper and lower of sine reference should be zero. By assuming $V_j$ and $V_k$ are voltage amplitudes adjacent and the area between the sine wave and the stepping wave are equal, the firing angle is equal to equation (10), while the others angle are equal to equation (11).

$$d_i = \sin^{-1}\left(\frac{Vj + Vk}{2}\right) \qquad (10)$$

$$d_j = \sin^{-1}(Vj) \quad and \quad d_k = \sin^{-1}(Vk) \qquad (11)$$

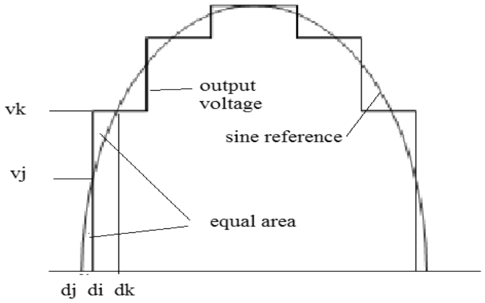

Fig. 5. Basic principle of THD minimization

## 2.3 System Configuration

Base on the sub section 2.1 and 2.2, so NN application for ACMLI controller for single-phase electric systems as discuss in this paper shown at Fig. 7. ACMLI in this system consist of five H-Bridges inverters, within circuit detail as shown in Fig.1. IGBT power electronic devices use as the switching components. Hence in this ACMLI consist twenty IGBT with similar type, and five separated DC sources within amplitude according to the DC voltage progression.

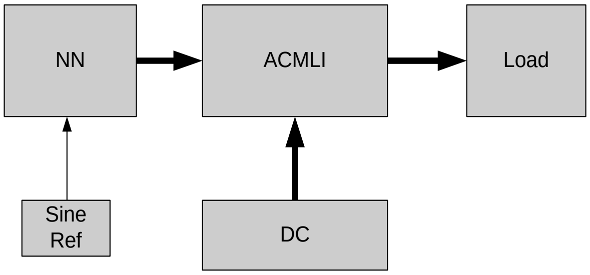

Fig. 7. System configuration

Input signals that entrance to NN have to normalized, and for output signal that exit from NN have to denormalized. Amount layers and neurons t in the each layer determined by optimization, where the optimum condition occurs if the NN system have little amount of neuron but have lowest error rate.

## 3. NN Controller SET UP

## 3.1 Training Signals

Training signals consists of sinus reference and trigger signals as output. Sine signal is easy to determine. But to determinate the trigger signal, some calculation or an algorithm as explained in Ref. [3] must used. After all of Training signal obtained, the Training signals must be save in the worksheet of Matlab programming windows, within file name have according to NN Training programs. An example of Training signal is shown in Fig.8.

Training signals usually need to regulate, hence the NN training due in the optimum data, time, and memories. These regulating include sampling, limiting, resuming and matrix justification.

If in the ACMLI used N amount of H-Bridges, so there are 4N+1 of Training signals, where each H-Bridge need four signal for each power electronics devices and one sine reference signal for all H-Bridges inverter.

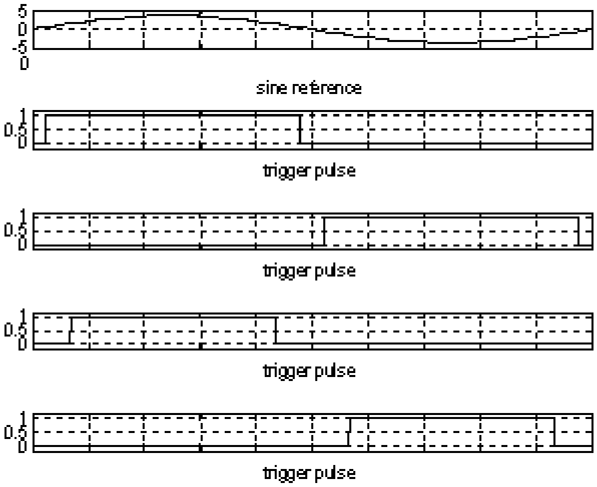

Fig. 8. Training signals example

## 3.2 NN Training

ACMLI within five H-Bridges used in this paper. Each H-Bridges have four power electronics devices that need to trigger. In this paper the triggering done using in the adding scheme only, so in the inverter only happen 2N+1 output voltage levels.

Training of NN is done using m-file program in Matlab, with the urgent instruction of the program as shown in Fig.9. This program have five step, fill the Training signal, setting the NN structures, Training with certain parameter, simulation and build NN in the simulink models.

```
%Training signal
x=(T6');
y=[T1'; T2'; T3'; T4'; T5'] ;
P=[x(:,:)];
T=y;

%NN setting
net = newff(minmax(P),{5 20},{'purelin', 'logsig', 'logsig' ,
'logsig', 'logsig', 'logsig', 'logsig', 'logsig', 'logsig', 'logsig',
'logsig', 'logsig', 'logsig', 'logsig', 'logsig', 'logsig', 'logsig',
'logsig', 'logsig'});
net.trainParam.lr=0.1;
net.trainParam.epochs= 150;

%NN Training
net.trainParam.lr = 0.6;
……….
net = train(net,P,T);
……
%NN simulation
Y = sim(net,P);
……

%Build NN Simulink
gensim(net);
```

Fig. 9. m-file of NN Training

Training signal souce is T1, T2, T3, T4, T5 and T6 matrix, where T6 is an input signal, while others are the output signals. After this fill these signals, NN structure then setting. In this program setting is done in the 1-5-20, that means one input within satlins transfer function, five hidden layer within satlins transfer functions and 20 output signals within purelin transfer functions. The weights and bias of this NN then calculate using Training step. In this Training, some parameters Training can be set, as Training rate, maximum epoch, error rate goals, etc.

Simulation step, optional to used, but build NN simulink step is an urgent step in this program. With using this step, a sophisticated NN can automatically construct base on gensim(net) instruction. Performance of training process in the graph format is shown in Fig. 10. This figure indicate that training have small error.

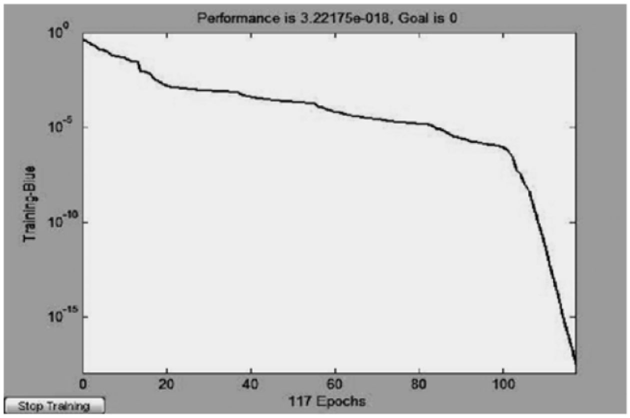

Fig. 10. Performance of training process

## 4. RESULTS AND DISCUSSIONS

Fig. 11 shows ACMLI simulation circuit using NN controller based on Matlab Simulink. This NN circuit obtains from NN training. In the NN circuit weight and bias are :

Hidden layer :

| Weight : | Bias : |
|---|---|
| 9.7278 | 46.8819 |
| -66.4382 | -86.9086 |
| -56.3512 | -193.7695 |
| -26.5248 | 111.0933 |
| -130.5043 | 145.6643 |

Output layer :

| Weight : | Bias : |
|---|---|
| [-3.9873979925178689;- | -1.0378 |
| 2.4516448217640328;- | |
| 9.2898968607227328;- | 2.6857 |
| 6.2620493906409553;- | -0.0098 |
| 19.049422965101748] | ……… |
| ………. | |
| [13.528947259153412;- | -8.7541 |
| 17.45542716175429;- | |
| 63.93938459593258;51.47495749 | 7.4550 |
| 3540209;104.62756360582517] | 5.2099 |

By using this simulation circuit, output voltage and frequency spectrum of ACMLI are shown in Fig.12. If this figure compare to ACMLI within conventional controller as shown in Fig. 13, hence we can say that NN controller can applied to ACMLI controller within comparable performance.

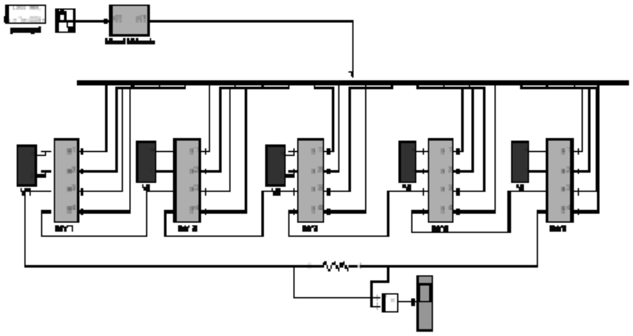

Fig. 11. ACMLI simulation circuit using NN controller

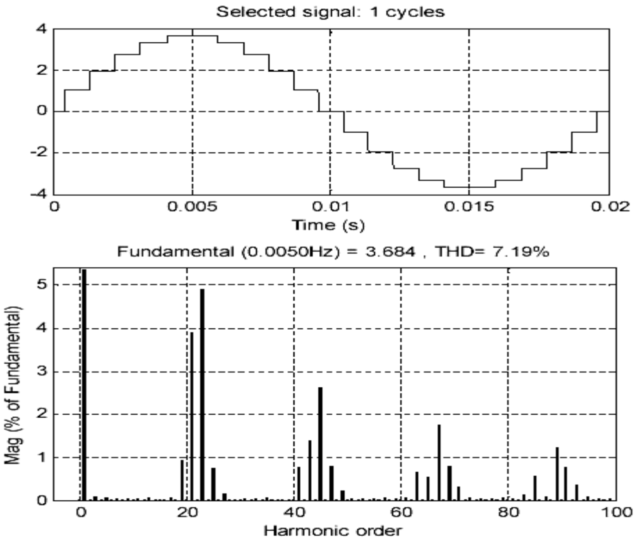

Fig. 12. Output voltage and frequency spectrum of ACMLI usin NN controller

But these result have different condition, ACMLI within conventional controller simulate in the sine frequency 50 Hz, while within NN controller, sine frequency for simulation is 0.005 Hz. These condition done because NN circuit require more time to solve calculation than conventional control. This problem can answer if the NN configuration implemented using Field Programmable Gate Array (FPGA) or others. The advantages of NN controller for ACMLI such as, it can design more easily, more flexible, quicker and better performance (depend on capability of designer) than conventional controller.

## 5. CONCLUTIONS

This paper show that NN controller can applied for ACMLI within more easily, more flexible, quicker and better performance than conventional controller. From simulation this goals can realized, as indicated by low error rate of training, and high performance of simulation, although the NN controller still need implemented on high speed computing component to compensate its speed.

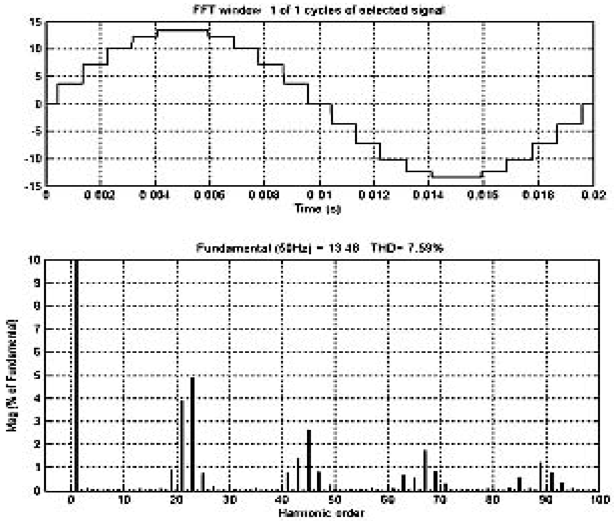

Fig. 13. Output voltage and frequency spectrum of ACMLI usin conventional controller

## 6. REFERENCES

[1] B. Cheng, F. Huang, "Multilevel Inverter Control Scheme", US Patent 6,867,987 B2, March, 2005.

[2] B. S. Suh, G. Sinha, M. D. Manjrekar, T. A. Lipo, "Multilevel Power Conversion – An Overview of Topologies and Modulation Strategies", IEEE-OPTIM Conference Record, pp. 11-24, vol. 2, 1998.

[3] Bambang Sujanarko, Mochamad Ashari and Mauridhi Hery Purnomo, "Universal Algorithm Control for Asymmetric Cascaded Multilevel Inverter", International Journal of Computer Applications (0975 – 8887), Vol.8, No.15, November 2010.

[4] Bimal K. Bose, "Neural Network Applications in Power Electronics and Motor Drives – An Introduction and Prespective", IEEE Trans. On Industrial Electronics., Vol. 54, No. 1, pp. 14-33, Pebruary 2007.

[5] E. Babaei, S.H. Hosseini, G.B. Gharehpetian, M. Tarafdar Haquea, M. Sabahi, "Reduction of dc voltage sources and switches in asymmetrical multilevel converters using a novel topology", Electric Power Systems Research, 77, 2007, pp. 1073–1085

[6] Howard Demuth, Mark Beale,Martin Hagan. 2008, "Neural Network Toolbox™ 6 User's Guide", The MathWorks, Inc. 3 Apple Hill Drive Natick, MA 01760-2098

[7] J.S. Lai and F.Z. Peng, "Multilevel converters—A new breed of power converters", IEEE Transactions on Industry Applications, vol.32, pp. 509–51, May/June, 1996.

[8] Kuhn, H. Ruger, N.E. Mertens, A., "Control Strategy for Multilevel Inverter with Non-ideal DC Sources", Power Electronics Specialists Conference (PESC), Hanover, 2007.

[9] L. M. Tolbert, John N. Chiasson, Zhong Du, and Keith J. McKenzie, "Elimination of Harmonics in a Multilevel Converter With Nonequal DC Sources", IEEE Transactions On Industry Applications, Vol. 41, No. 1, January/February 2005, pp. 75-82.

[10] M. G. Hosseini Aghdam, S. H. Fathi, G. B. Gharehpetian, "A Complete Solution of Harmonics Elimination Problem in a Multi-Level Inverter with Unequal DC Sources", Journal of Electrical Systems, 3-4, 2007, pp.259-271

[11] M. H. Rashid, "Power Electronics Handbook", Academic Press, 2001, Canada

[12] R. Taleb, A. Meroufel, 2009, "Control of Asymmetrical Multilevel Inverter Using Artificial Neural Network", Electronics and Electrical Engineering, No. 8(96), 2009, pp. 93-98.

[13] S. J. Park, F. S. Kang, S. E. Cho, C.J. Moond, H. K. Nam, "A novel switching strategy for improving modularity and manufacturability of cascaded-transformer-based multilevel inverters", Electric Power Systems Research, 74. 2005, pp. 409–416

[14] S. Krishna, "Harmonic Elimination by Selection of Switching Angles and DC Voltages in Cascaded Multilevel Inverters", Fifteenth National Power Systems Conference (NPSC), IIT Bombay, December 2008.

## AUTHOR PROFILE

**Bambang Sujanarko** received the B.Sc. from Universitas Gadjah Mada, Yogyakarta Indonesia and Master from Universitas Jember, Indonesia. He is senior lecture of Departement Electrical Universitas Jember and currently toward his Ph.D in Institut Teknologi Sepeluh Nopember (ITS) Surabaya, Indonesia. His research interests included power electronics and renewable energy systems, hybrid power systems, artificial intelligent, and instrumentation.

**Mochamad Ashari** received the Bachelor degree in electrical engineering from the Institut Teknologi Sepuluh Nopember(ITS) Surabaya, Indonesia, in 1989 and Master and Ph.D. from Curtin University of Technology, Perth, Australia. He has been with ITS since 1990 as a Lecturer in the Department of Electrical Engineering. He is a Professor and head of Electrical Engineering ITS. His research interests include power electronics and inverter applications, power system modeling, simulation, and analysis of hybrid power systems.

**Mauridhi Hery Purnomo** received the B.S. degree from Institut Teknologi Sepuluh Nopember (ITS) Surabaya, Indonesia and Master ad Ph.D from Osaka City University, Osaka, Japan. He is a Professor in the Department of Electrical Engineering, ITS. Since 2007, he was vice director on ITS postgraduate program. He has been engaged in research and teaching in the areas of intelligent system and pattern recognition, power system simulations, and computer programming.