A New Artificial Neural Network Controller for Direct Control Method for Matrix Converters

Hong Hee Lee  
NARC, Ulsan University, Korea  
hhlee@mail.ulsan.ac.kr

Phan Quoc Dzung  
Faculty of Electrical & Electronic Engineering  
HCMC University of Technology  
Ho Chi Minh City, Vietnam  
pqdung@hcmut.edu.vn

Le Minh Phuong  
Faculty of Electrical & Electronic Engineering  
HCMC University of Technology  
Ho Chi Minh City, Vietnam  
lmphuong@hcmut.edu.vn

Le Dinh Khoa  
Faculty of Electrical & Electronic Engineering  
HCMC University of Technology  
Ho Chi Minh City, Vietnam  
khoaledinh@hcmut.edu.vn

Abstract—This work presents a new artificial neural network (ANN) Controller for implementing the Direct control method (DCM) for Matrix converters (MC) to decrease the time of calculation of the conventional DSP control system. To avoid the difficult calculation of ANN-DCM, the design uses the individual training strategy with the fixed weight and the supervised models. A computer simulation program is developed using Matlab/Simulink together with the Neural Network Toolbox. The simulated results demonstrate the good quality and the robustness of the proposed ANN-DCM-Controller for MC.

I. INTRODUCTION

Three- phase matrix converters (fig.1) have received considerable attention in recent years because they may become a good alternative to voltage-source inverter pulse-width-modulation (VSI-PWM) converters. In reality, the matrix converter provides important benefits such as bidirectional power flow, sinusoidal input current with adjustable displacement angle (i.e. controllable input power factor), and a great potential for size reduction due to the lack of dc-link capacitors for energy storage [1-3].

The direct control method DCM [4] for matrix converter has good behaviors such as simple method by using mainly the look-up tables, no requirements for coordinate transformation and PWM pulse generation. The use of DCM for matrix converter has been worked out with a good performance [4].

In order to avoid time-consuming searches in the tables according to this method, a lookup-table is used, which consists of all possible combinations of 12 line-side voltage sections with 6 load-side sectors and 12 load-side current sections with 6 line-side sectors. This look-up table is very large, which consists of 5184 elements and aims to increase the execution time [4]. Therefore, it is difficult to implement DCM using common DSP hardware with serial calculations.

However, the distortion of the line-side currents could be reduced, if a shorter cycle time of the controller could be realized. Therefore, Artificial Neural Network (ANN) Controller, having faster parallel calculation and simpler circuit structure, is a good alternative for implementation of DCM.

The applications of ANN technique have been developed strongly in power electronics for recent years. Several researches of ANN implementation of Space vector modulation have been worked out for conventional VSI [5-8].

Fig. 1. Schematic representation of a matrix converter

Fig. 2. The block diagram of the conventional control

This paper presents an new ANN controller for Direct control method for Matrix converter with 7 types of subnets and about two hundreds neurons to compare with the DSP serial calculations of the DCM for MC, the
control precision and execution time of DCM can be significantly improved using the ANN algorithm.

The proposed back-propagation type feed-forward ANN-DCM in this paper has been successfully trained by using individual training strategy with 10 subnets to overcome the complexity of DCM.

II. PRINCIPLE OF A DIRECT CONTROL METHOD FOR MATRIX CONVERTERS

The block diagram of the control and the flowchart of the mode selection in the direct control method are shown in Fig. 2, Table 1,2. In principle, the control technique of the matrix converter selects, at each sampling period, the proper switching configuration, which allows the compensation of instantaneous errors in output current and (or) input current [4].

Fig. 3. The flowchart of the mode selection

Table 1. Sector of output voltage $V_O$ as function of switching configuration mode $M$ and section of input voltage $V_I$

<table>
<thead>
<tr>
<th>Mode</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>

III. NEURAL-NETWORK BASED DCM CONTROLLER

Based on DCM for MC, the neural network controller (Fig.4) is divided in to 7 sub-nets, which are individually trained: 1) Mode selection for load side control error sub-net (supervised) ANN-1. 2) Mode selection for line side control error sub-net (supervised) ANN-2. 3) Optimal mode selection (supervised) ANN-3. 4) Hysteresis comparator sub-net (fixed-weight) with recurrent neurons ANNx,y,xy. 5) Code generation for mode selection sub-net (supervised) ANN M. 6) Code generation for output selection sub-net (supervised) ANN SV,SI. 7) Zero voltage vectors generation sub-net (supervised) ANN-0.

A. Mode selection for load side control error sub-net

This sub-net is implemented for the purpose to determine which modes can be selected for reduction of...
the control error $\Delta k_0$ (Table 3). The Table 1 can be transformed into the Table 3 for this purpose.

A two-layer network is employed to implement this subnet. Sector of input voltage $\theta V_1$ and sector of output voltage $\theta V_o$ are inputs of this network. Modes of switching configurations ($M_o$) are outputs (Fig.7). The group-3 modes in the table 3 are sorted in such a way, which has low, medium and large amplitude.

The sub-net is obtained by training (supervised) with trainlm function – Levenberg –Marquardt algorithm, the acceptable for training squared error is $10^{-10}$. The optimal number of neurons of 1st layer is 60 logsig neurons, the 2nd layer has 5 purelin neurons. So, the total number of neurons is 65 neurons (convergence obtained for 393 epochs) (Fig.5, 6).

B. Mode selection for line side control error sub-net

Similar to the previous subnet, this sub-net is implemented for the purpose to determine which modes can be selected for reduction of the control error $\Delta_k$ (Table 4). The Table 2 can be transformed into the Table 4 for this target.

![Fig. 5](image)

Listing m-file for training ANN-1 with Matlab/Simulink

**Fig. 5:** Listing m-file for training ANN-1 with Matlab/Simulink

**Fig. 6:** Training ANN-1 with Matlab/Simulink

**Fig. 7:** Mode selection for load side control error sub-net (ANN-1)

A two-layer network is employed to implement this subnet. Sector of input current $\theta i_i$ and sector of output current $\theta i_o$ are inputs of this network. Modes of switching configurations ($M_o$) are outputs (Fig.8).

The groupe-3 modes in the Table 4 are sorted in such a way, which has low, medium and large amplitude.

**TABLE 3. MODE AS FUNCTION OF SECTOR OF OUTPUT VOLTAGE $V_o(1-6, 0)$ AND SECTOR OF INPUT VOLTAGE $V_i(1-12)$**

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4.7, 18.3</td>
<td>7.8, 18.2</td>
<td>8.1, 18.2</td>
<td>8.2</td>
<td>3.5, 16.1</td>
<td>3.5, 16.1</td>
<td>8.9, 16.2</td>
<td>8.9, 16.2</td>
<td>7.9, 17.2</td>
<td>7.9, 17.2</td>
<td>1.4, 17.2</td>
</tr>
<tr>
<td>4</td>
<td>9.6, 17.2</td>
<td>9.6, 17.2</td>
<td>9.6, 17.2</td>
<td>9.6, 17.2</td>
<td>9.6, 17.2</td>
<td>9.6, 17.2</td>
<td>9.6, 17.2</td>
<td>9.6, 17.2</td>
<td>9.6, 17.2</td>
<td>9.6, 17.2</td>
<td>9.6, 17.2</td>
</tr>
<tr>
<td>6</td>
<td>5.1, 22.3</td>
<td>5.1, 22.3</td>
<td>5.1, 22.3</td>
<td>5.1, 22.3</td>
<td>5.1, 22.3</td>
<td>5.1, 22.3</td>
<td>5.1, 22.3</td>
<td>5.1, 22.3</td>
<td>5.1, 22.3</td>
<td>5.1, 22.3</td>
<td>5.1, 22.3</td>
</tr>
</tbody>
</table>

![Fig. 7](image)

**Fig. 7:** Mode selection for load side control error sub-net (ANN-1)

![Fig. 8](image)

**Fig. 8:** Mode selection for line side control error sub-net (ANN-2)
TABLE 4. MODE AS FUNCTION OF SECTOR OF INPUT CURRENT $I_1$ (1-12) AND SECTOR OF OUTPUT CURRENT $I_0$ (1-6, 0)

<table>
<thead>
<tr>
<th>$I_1$</th>
<th>$I_0$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1,16,18.2</td>
<td>1,16,18.2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1,16,18.2</td>
<td>1,16,18.2</td>
</tr>
<tr>
<td>2</td>
<td>6.8,11,23.2</td>
<td>6.8,11,23.2</td>
<td>6.8,11,23.2</td>
<td>6.8,11,23.2</td>
<td>6.8,11,23.2</td>
<td>6.8,11,23.2</td>
<td>6.8,11,23.2</td>
<td>6.8,11,23.2</td>
<td>6.8,11,23.2</td>
<td>6.8,11,23.2</td>
<td>6.8,11,23.2</td>
<td>6.8,11,23.2</td>
<td>6.8,11,23.2</td>
</tr>
<tr>
<td>3</td>
<td>2.16,16.1</td>
<td>2.16,16.1</td>
<td>2.16,16.1</td>
<td>2.16,16.1</td>
<td>2.16,16.1</td>
<td>2.16,16.1</td>
<td>2.16,16.1</td>
<td>2.16,16.1</td>
<td>2.16,16.1</td>
<td>2.16,16.1</td>
<td>2.16,16.1</td>
<td>2.16,16.1</td>
<td>2.16,16.1</td>
</tr>
<tr>
<td>4</td>
<td>6.9,13,24.5</td>
<td>6.9,13,24.5</td>
<td>6.9,13,24.5</td>
<td>6.9,13,24.5</td>
<td>6.9,13,24.5</td>
<td>6.9,13,24.5</td>
<td>6.9,13,24.5</td>
<td>6.9,13,24.5</td>
<td>6.9,13,24.5</td>
<td>6.9,13,24.5</td>
<td>6.9,13,24.5</td>
<td>6.9,13,24.5</td>
<td>6.9,13,24.5</td>
</tr>
<tr>
<td>5</td>
<td>2.14,17,2</td>
<td>2.14,17,2</td>
<td>2.14,17,2</td>
<td>2.14,17,2</td>
<td>2.14,17,2</td>
<td>2.14,17,2</td>
<td>2.14,17,2</td>
<td>2.14,17,2</td>
<td>2.14,17,2</td>
<td>2.14,17,2</td>
<td>2.14,17,2</td>
<td>2.14,17,2</td>
<td>2.14,17,2</td>
</tr>
<tr>
<td>6</td>
<td>4.7,10,22.2</td>
<td>4.7,10,22.2</td>
<td>4.7,10,22.2</td>
<td>4.7,10,22.2</td>
<td>4.7,10,22.2</td>
<td>4.7,10,22.2</td>
<td>4.7,10,22.2</td>
<td>4.7,10,22.2</td>
<td>4.7,10,22.2</td>
<td>4.7,10,22.2</td>
<td>4.7,10,22.2</td>
<td>4.7,10,22.2</td>
<td>4.7,10,22.2</td>
</tr>
</tbody>
</table>

The sub-net is obtained by training (supervised) with trainlm function – Levenberg – Marquardt algorithm, the acceptable for training squared error is $10^{-10}$. The optimal number of neurons of 1st layer is 52 logsig neurons, the 2nd layer has 5 purelin neurons. So, the total number of neurons is 57 neurons (convergence obtained for 1703 epochs).

C. Optimal mode selection sub-net

The purpose of this sub-net is to find out a mode, which satisfies both controllers (load side control and line side control) simultaneously into Table 3 and Table 4. This subnet has the advantage in comparison with traditional approach, while the search for optimal mode takes a large time-consuming.

A two-layer network is employed to implement this subnet. Two outputs of ANN-1 (5 inputs) and ANN-2 (5 inputs) are inputs of this network. Optimal mode (one mode) of switching configurations ($M_s$) and generated code $C_4$ ($C_4=0$ if $M_s \neq 0$; $C_4=1$ if $M_s=0$) are outputs. The sub-net is obtained by training (supervised) with trainlm function – Levenberg – Marquardt algorithm, the acceptable for training squared error is $10^{-10}$. The optimal number of neurons of 1st layer is 60 logsig neurons, the 2nd layer has 2 purelin neurons. So, the total number of neurons is 62 neurons (convergence obtained for 8556 epochs) (Fig.9).

Table: Optimal mode selection sub-net (ANN-3)

Table 5. Codes as function of codes from the load side and line side control error

<table>
<thead>
<tr>
<th>$C_x$</th>
<th>$C_y$</th>
<th>$C_{x,y}$</th>
<th>$C_{x,y}$</th>
<th>$C_{x,y}$</th>
<th>$C_{x,y}$</th>
<th>$C_{x,y}$</th>
<th>$C_{x,y}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>ANN</td>
<td>ANN</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>ANN</td>
<td>ANN</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>ANN-ANN</td>
<td>ANN-ANN</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>ANN-ANN</td>
<td>ANN-ANN</td>
</tr>
</tbody>
</table>

Fig. 9. Optimal mode selection sub-net (ANN-3)

Table 6. Codes as function of the values of the load side and line side control error

<table>
<thead>
<tr>
<th>$X$</th>
<th>$Y$</th>
<th>$C_{x,y}$</th>
<th>$C_{x,y}$</th>
<th>$C_{x,y}$</th>
<th>$C_{x,y}$</th>
<th>$C_{x,y}$</th>
<th>$C_{x,y}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

437
D. Hysteresis comparator Sub-Net [2]

The hysteresis comparator (Fig.10), which is implemented by a recurrent network with hardlim and purelin neurons (fixed – weight), is to generate the codes of the load side \( C_x \), the line side \( C_y \) control error and \( C_{xy} \) such as follows:

- If the load side control error is out of a tolerable margin \( \varepsilon \) then \( C_x = 1 \), otherwise \( C_x = 0 \).
- If the line side control error is out of a tolerable margin \( \varepsilon \) then \( C_y = 1 \), otherwise \( C_y = 0 \).
- If the load side control error is greater than the line side then \( C_{xy} = 1 \), otherwise \( C_{xy} = 0 \).

E. Code generation for mode selection sub-net

Similar to the previous subnet, this sub-net is implemented for the purpose to generate codes, which help to realize the algorithm of DCM as shown in Table 5.

- If the load side and the line side control error are not out of a tolerable margin \( \varepsilon \) then \( C_1 = 0, C_2 = C_3 = 1 \) ⇒ take a Zero Vector : Subnet ANN0;
- If there is a mode which satisfies both controllers simultaneously subnet load side and line side ⇒ use this mode : Subnet ANN3, \( C_4 = 0 \);
- If \( M_s = 0 \) and the weighted line side control error larger than the load side error ⇒ use subnet ANN2, \( C_1 = 1, C_2 = 0, C_3 = C_4 = 1 \);
- If \( M_s = 0 \) and the weighted line side control error smaller than the load side error ⇒ use subnet ANN1, \( C_1 = 1, C_2 = 1, C_3 = 0, C_4 = 1 \).

A two-layer network is used to implement this subnet. The codes \( x, y \) are inputs of network. The codes \( C_1,2,3 \) are outputs. The sub-net is obtained by training (supervised) with trainlm function – Levenberg – Marquardt algorithm, the acceptable for training squared error is \( 10^{-10} \).

The optimal number of neurons of 1st layer is 2 logsig neurons, the 2nd layer has 4 purelin neurons. So, the total number of neurons is 6 neurons (convergence obtained for 4 epochs) (Fig.11).

<table>
<thead>
<tr>
<th>( C_1 )</th>
<th>( C_2 )</th>
<th>( C_3 )</th>
<th>( M_s )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0, 24</td>
</tr>
</tbody>
</table>

F. Code generation for output selection sub-net

Similar to the previous subnet, this sub-net is implemented for the purpose to generate codes, which are used to select within group 3 a mode, which produces a vector in the desired direction and a low, medium, or large amplitude, respectively (Table 6).

- \( C_{v1}, C_{i1} \) : the multiplied coefficients for the groupe -1 modes (= 0 : do not use this groupe)
- \( C_{v2}, C_{i2} \) : the multiplied coefficients for the groupe -3 modes (=1 : select the small load side, line side amplitude)
- \( C_{v3}, C_{i3} \) : the multiplied coefficients for the groupe -3 modes (=1 : select the medium load side, line side amplitude)
- \( C_{v4}, C_{i4} \) : the multiplied coefficients for the groupe -3 modes (=1 : select the large load side, line side amplitude)

A two-layer network is used to implement this subnet. The codes \( x, y \) are inputs of network. The codes \( C_{v1...4}, C_{i1...4} \) are outputs. The sub-net is obtained by training (supervised) with trainlm function – Levenberg – Marquardt algorithm, the acceptable for training squared error is \( 10^{-10} \).

The optimal number of neurons of 1st layer is 2 logsig neurons, the 2nd layer has 4 purelin neurons. So, the total number of neurons is 6 neurons (convergence obtained for 4 epochs) (Fig.12, 13).

G. Zero voltage vectors generation sub-net

This sub-net is implemented for the purpose to generate the group 4 zero voltage vectors depending on codes \( C_{1,3} \) (Table 7).

A two-layer network is used to implement this subnet. The codes \( C_{1,2,3} \) are inputs of network. The codes \( M_0 \) are outputs. The sub-net is obtained by training (supervised) with trainlm function – Levenberg – Marquardt algorithm, the acceptable for training squared error is \( 10^{-10} \).

The optimal number of neurons of 1st layer is 2 logsig neurons, the 2nd layer has 3 purelin neurons. So, the total number of neurons is 5 neurons (convergence obtained for 5 epochs) (Fig.14).
IV. SIMULATION OF THE PROPOSED ANN–DCM CONTROLLER

A Simulink/Matlab program with the toolbox of neural–network is used to train and simulate the complete ANN-DCM controller with the above-mentioned sub-nets for different mode of operation (Fig.15). The ANN-DCM controller consists of 6 inputs (x, y, \( \theta_{vi} \), \( \theta_{vo} \), \( \theta_{ii} \), \( \theta_{io} \)) and 1 output (M).

![Fig. 15. Complete ANN-DCM Controller for MC](image)

![Fig. 16. Testing ANN-1 Subnet of the controller](image)

TABLE 8. TABLE OF SIMULATION RESULTS FOR ANN-DCM CONTROLLER

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>( \theta_{vi} )</th>
<th>( \theta_{vo} )</th>
<th>( \theta_{ii} )</th>
<th>( \theta_{io} )</th>
<th>M</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>7</td>
<td>6</td>
<td>5</td>
<td>2</td>
<td>14</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>8</td>
<td>5</td>
<td>6</td>
<td>4</td>
<td>25</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>10</td>
<td>5</td>
<td>8</td>
<td>3</td>
<td>10</td>
</tr>
</tbody>
</table>

Simulation results (Fig.16, Table 8) demonstrate the validity of the proposed ANN-DCM Controller for MC, while the values of mode (M) are exactly the same in comparison with the conventional algorithm.

Furthermore, experimental results would be validated by DSPACE 1104.

V. CONCLUSION

This paper presents a new complete artificial-neural-network based direct-control method (ANN-DCM) scheme for the Matrix converter. Based on the understanding of DCM inconvenient (very large look-up-tables), the supervised methods with the training individually strategy are implemented for the controller design.

Compared with the DSP based DCM, the proposed ANN-DCM scheme for Matrix converter incurs much shorter execution times and, hence, the errors caused by control time delays are minimized and the distortion of the line-side currents could be reduced.

VI. REFERENCES


