TELKOMNIKA Indonesian Journal of Electrical Engineering
Vol.12, No.7, July 2014, pp. 5121 ~ 5128
DOI: 10.11591/telkomnika.v12i7.5927

Solution Trajectories for a Single-Phase Programmed PWM Inverter

Ayong Hiendro*, Syaifurrahman, Dedi Triyanto, Junaidi
Department of Electrical Engineering, UniversitasTanjungpura
Jalan Jend. A. Yani, Pontianak 78124, Indonesia
*Corresponding author, e-mail: ayongh2000@yahoo.com

Abstract

This paper presents solution trajectories for programmed PWM technique to eliminate specific order harmonics in a single phase inverter. Evolutionary algorithm is applied to determine optimum switching angles in order to elmininate low order harmonics for modulation index: -1 ≤ M ≤ +1. An implementation using a DE2-115 Cyclone IVE FPGA device is also reported in this paper. The experimental results show that the technique effectively eliminates the specific harmonics, and offers low harmonic distortions on the inverter output after filtering.

Keywords*: FPGA, harmonic elimination, evolutionary algorithm, programmed PWM, trajectory*

1. Introduction

Pulse width modulation (PWM) inverters have been widely studied and applied in many applications, such as motor drive, uninterruptable power supply, and renewable power generation. In order to get good sinusoidal output from the PWM inverter, it has to operate on higher switchingfrequency. Consequently, the high switching frequency generates highswitching stress on semiconductor devices and hence, increasespower losses in them [1].

Programmed PWM is an alternative to traditional PWM techniques. Programmed PWM works with low switching frequencyso thatswitching losses is minimal. This technique can be use to eliminate specific undesired lower-order harmonic components from output of singlephase inverters and to control its fundamental. The remaining higher-order harmonics are then suppressed by using a small L-C passive filter.

The main problem of the programmed PWM is in solving a set of nonlinear transcendental equations. Many computational techniques have been developed in [2-9] to determine optimum switching angles in order to eliminate the specific lower-order harmonics, but the most of them need very hard derivation to find the numerical expression of the nonlinear equations before obtaining the optimum switching angles. In this paper, an evolutionary algorithm is applied to optimize the switching pattern for the PWM inverter using the transcendental equations without any numerical transformations.

An FPGA device is implemented to generate the optimum PWM waveforms. The FGPA as a hard-speed hard-wired logic has a high computation speed capability [10, 11]. Such computation capability is required to convert switching angles obtained from computational processes into PWM waveforms.

The objective of this paper is to report solutions for switching angles problems in a programmed PWM inverter for single-phase applications. The solution patterns for both positive and negative values of modulation index (*M*) in the range of the programmed PWM capability are investigated. Finally, experimental results are presented to validate the theoretical arguments.

2. Schemes for Programmed PWM in a Two-Level Single-Phase Inverter

The programmed PWM pattern for a single-phase inverter (as seen in Figure 1), is shown in Figure 2. As the pattern is odd quarter-wave symmetry, only odd-order harmonic components exist. The Fourier series expansion of this output voltage waveform is written as:

Received February 19, 2014; Revised March 15, 2014; Accepted March 28, 2014

$$v(\omega t) = \sum_{n=1,3,5,7,...}^{\infty} V_n \sin(n\omega t) \quad (1)$$

The magnitude of harmonic components (including the fundamental) V_n is then defined by:

$$V_n = \frac{4V_{dc}}{n\pi}\left[1 + 2\sum_{k=1}^{N}(-1)^k \cos(n\alpha_k)\right] \quad (2)$$

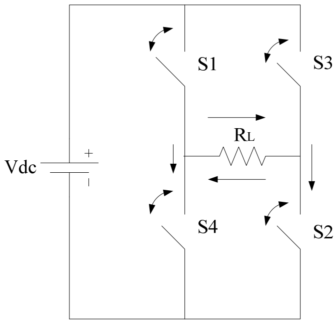

Figure 1. Single-phase Inverter Circuit

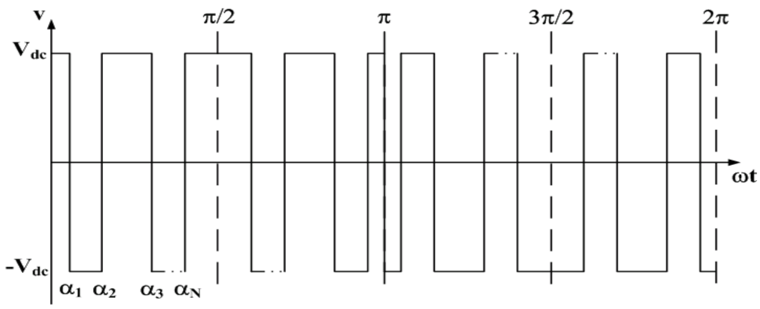

Figure 1. Programmed PWM Waveform for a Single-phase Inverter

The programmed PWM technique concerns to eliminate (*N*-1) harmonics completely from the waveform through calculating of *N* switching angles in the first quarter period. The remaining one equation is reserved for controlling the fundamental magnitude V_1 at a particular modulation index value. Hence, the equations for solving the *N* switching angles α_1, α_2,…,α_N are arranged into:

$$f_1(\alpha) = \frac{4}{\pi M}\left[1 + 2\sum_{k=1}^{N}(-1)^k \cos(\alpha_k)\right] - 1$$

$$f_3(\alpha) = \frac{4}{3\pi M}\left[1 + 2\sum_{k=1}^{N}(-1)^k \cos(3\alpha_k)\right]$$

$$f_5(\alpha) = \frac{4}{5\pi M}\left[1 + 2\sum_{k=1}^{N}(-1)^k \cos(5\alpha_k)\right]$$

$$,\ldots$$

$$f_{(2N-1)}(\alpha) = \frac{4}{(2N-1)\pi M}\left[1 + 2\sum_{k=1}^{N}(-1)^k \cos((2N-1)\alpha_k)\right] \quad (3)$$

Where f_3, f_5,…,$f_{(2.N-1)}$ are the normalized magnitude (with respect to the fundamental) of harmonics to be eliminated and $M=V_1/V_{dc}$ is the modulation index.

The objective function created for optimization is expressed by:

$$f(\alpha)=|f_1|+|f_3|+|f_5|+|f_7|+|f_9|+...+|f_{(2N-1)}|,$$
$$|f_1|,|f_3|,|f_5|,|f_7|,|f_9|,...,|f_{(2N-1)}|, f<\varepsilon \quad (4)$$

And the solutions must satisfy the conditions:

$$\alpha_1<\alpha_2<\ldots<\alpha_N<90^o \quad (5)$$

With the desired level of accuracy ε.

Once the switching angles α_1, α_2,…,α_N are found, the rest of angles are calculated by:

$$\alpha_k=180° - \alpha_{(2N+1)-k}, \text{ for } k=N+1 \text{ to } 2N,$$
$$\alpha_{2N+1}=180°,$$
$$\alpha_k=360° - \alpha_{(4N+2)-k}, \text{ for } k=2N+2 \text{ to } 4N+1,$$
$$\alpha_{4N+2}=360° \quad (6)$$

Each pulse delay (PD) and pulse width (PW) between two consecutive switching angles for the cycle duration T (as shown in Fig. 2),are specified by:

$$PD_k=\alpha_{(2k-1)}T/360,$$
$$PW_k=(\alpha_{2k} - \alpha_{2k-1})T/360°, \text{for } k=1 \text{ to } 2N+1 \quad (7)$$

3. Optimization of Swithing Angles

The main procedures in an evolutionary algorithmfor the optimization process are initialization, mutation, crossover or recombination and selection. The initial population is randomly selected and should cover the entire parameter space. The mutant individuals are created by adding the weighted difference between two parent individuals. Then, the parameter of the mutant individual and the parent individual are mixed to yield the trial individual. If the trial individual is better than the parent individual, then the trial individual replaces the parent individual in the next generation.

The parameters to be set for the algorithm to work consist of objective parameter (N), population size (NP), mutation factor (F), crossover rate (CR) and boundaries. In this application, the objective parameter specifies the number of optimized switching angles. The boundaries must satisfy (5) and both F and CR are in the range of [0, 1].

The first step in the optimization process is to create an initial population of switching angles as the candidate solutions by assigning random values to each boundary. Such switching angle must lie inside the feasible bounds (lower and upper bounds). The initialization is assigned by:

$$\boldsymbol{\alpha}_{i,j}^{(0)} = \boldsymbol{\alpha}_{minj} + rand_j(\boldsymbol{\alpha}_{maxj} - \boldsymbol{\alpha}_{minj}), \quad (8)$$
$$i=1,2,\ldots,NP, \; j=1,2,\ldots,N$$

Where $\boldsymbol{\alpha}_{i,j}^{(0)}$ is the initial population (generation G=0) of candidate solutions, $\boldsymbol{\alpha}_{minj}$ and $\boldsymbol{\alpha}_{maxj}$ are the lower and upper bounds of j^{th} decision switching angles, and $rand_j$ is a random value within [0,1] generated according to a uniform probability distribution.

Evaluating the fitness value of each switching angle of the population is carried out by using (4).The best switching angle $\alpha_{bestj}^{(0)}$ and the best value $f_{best}^{(0)}$ are then selected by using:

$$f_{best}^{(0)}=f(\alpha_{bestj}^{(0)}), \quad \alpha_{bestj}^{(0)} \in \boldsymbol{\alpha}_{i,j}^{(0)} \quad (9)$$

Mutation or differential operator creates a mutant switching angle by perturbing a randomly selected switching angle with the difference of the two other randomly selected switching angles. All of these switching angles must be different from each other. To control the perturbation and improve convergence, the difference between two switching angles is amplified by a mutation factor *F*, a constant in the range of [0, 1]. For each parent (target) switching angle $\alpha_{i,j}^{(G),}$ a mutant switching angle $v_{i,j}^{(G)}$ is produced using one of the following mutation variant:

$$v_{i,j}^{(G+1)}=\alpha_{ra,j}^{(G)}+F(\alpha_{rb,j}^{(G)}-\alpha_{rc,j}^{(G)})+F(\alpha_{rd,j}^{(G)}-\alpha_{re,j}^{(G)}) \quad (10)$$

Where the indices *ra*, *rb*, *rc*, *rd*, $re \in$ {1,2,…,*NP*} are randomly chosen mutually exclusive integers and must be different from each other and all are different from the base index *i*. The angles $\alpha_{ra,j}^{(G)}$, $\alpha_{rb,j}^{(G)}$ and $\alpha_{rc,j}^{(G)}$ are the shuffled individuals from the population $\boldsymbol{\alpha}_{i,j}^{(G)}$, while *G*=0,1,2,…, G_{max} denotes the subsequent generation created for each iteration (generation). The angle $\alpha_{bestj}(G)$ is the best switching angle with the best fitness in the population at generation *G.*

Following the mutation operation, crossover is applied to the population. Crossover operator is used to increase the diversity of the mutant switching angles. It generates a trial switching angle $t_{i,j}^{(G+1)}$. The trial switching angle is a combination of $v_{i,j}^{(G+1)}$ and $\alpha_{i,j}^{(G)}$ based on binomial scheme. In the binomial scheme, if the random number is less or equal than *CR*, the parameter will come from $v_{i,j}^{(G+1)}$, otherwise the parameter comes from $\alpha_{i,j}^{(G)}$. If *CR*=1, it means that $t_{i,j}^{(G+1)}$ will be composed entirely of $v_{i,j}^{(G+1)}$. The binomial crossover can be expressed as:

$$t_{i,j}^{(G+1)} = \begin{cases} v_{i,j}^{(G+1)} & \text{if } (rand_j \le CR) \text{ or } (j = j_{rand}) \\ \alpha_{i,j}^{(G)} & \text{otherwise} \end{cases} \quad (11)$$

Where *i*=1,2,…,*NP*, *j*=1,2,…,*N*, and j_{rand} is a randomly chosen index $\in$ {1,2,…,*N*} that guarantees $t_{i,j}(G+1)$ to get at least one parameter from $v_{i,j}^{(G+1)}$.

Finally, $t_{i,j}^{(G+1)}$ yielded from the crossover operation is accepted for the next generation if and only if it has an equal or lower value of the objective function than that of its parent $\alpha_{i,j}^{(G)}$. It can be expressed as follows:

$$\alpha_{i,j}^{(G+1)} = \begin{cases} t_{i,j}^{(G+1)} & \text{if } f(\alpha_{i,j}^{(G+1)}) \le f(\alpha_{i,j}^{(G)}) \\ \alpha_{i,j}^{(G)} & \text{otherwise} \end{cases} \quad (12)$$

The best switching angle and value of the current generation are also selected in here as:

$$f_{best}^{(G+1)}=f(\alpha_{bestj}^{(G+1)}), \quad \alpha_{bestj}^{(G+1)} \in \boldsymbol{\alpha}_{i,j}^{(G+1)} \quad (13)$$

The mutation, crossover and selection processes are repeated to create a new next generation until $f_{best}^{(G+1)}$ meets its criterion ε and results in $\alpha_{bestj}^{(G+1)}$ as the satisfied switching angle.

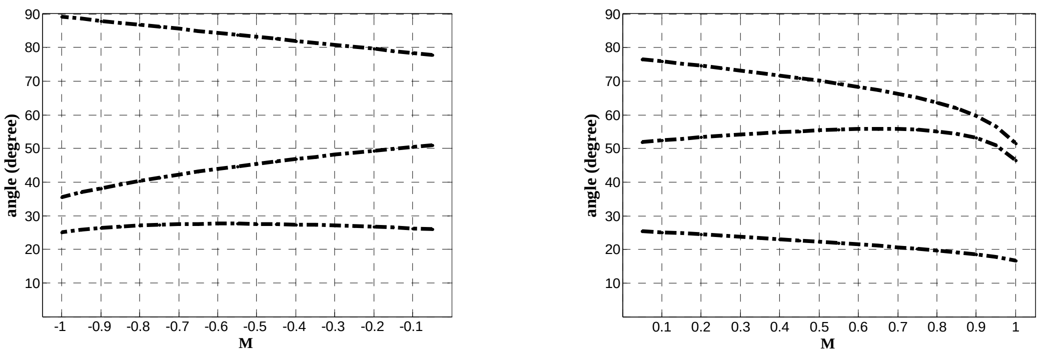

Figure 3. Swithing Angles Solution Trajectories for N=3

Figure 3 illustrates switching patterns of the single-phase inverter for index modulation: -1 ≤*M*≤ +1 as results of the aforementioned optimization.

4. Field Programmable Gate Array (FPGA) Implementation

The implementation system of programmed PWM is using the DE2-115 Cyclone IVE FPGA device. The pulse generator development diagram is shown as in Figure 4. The DE2-115 board includes an oscillator that produces 50MHzclock signal. The timer/counter is used to count against clock pulse frequency of 50Hz or 20ms. The ROM lookup table (LUT) stores the PWM signal pattern. The PWM signal is then compared with the counter output.Thecomparator is an XNOR gate that generates HIGH pulse output if both inputs are the same. The toggle flip-flop (T-FF) constructs a flip-flop and it toggles from one state to the next (HIGH to LOW or LOW to HIGH) at every clock cycle in order to generate PWM.

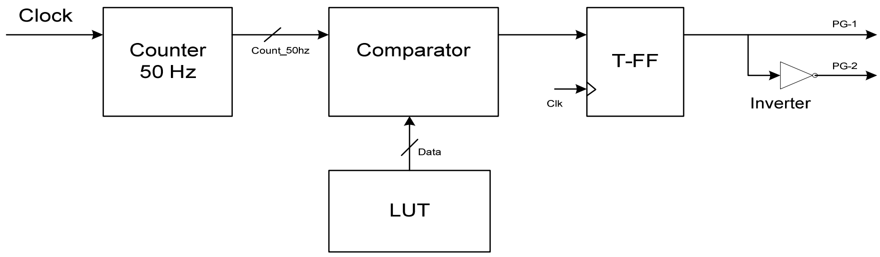

Figure 4. Block Diagram of Pulse Generator Development

In this paper, the optimal switching angles at*M*=-1 and *N*=3 are used to be the FPGA implementation. The switching angles for *M*=-1 and *N*=3 (as shown in Figure 3) is presented as in Table 1. Pulse delay (PD) and pulse width (PW) of the PWM can be obtained by using (7) andthe results are presented in Table 2. The results are converted into microsecond (ms) unit.

Table 1. Optimal Switching Angles

	M=-1
α_1	24.9940°
α_2	35.5260°
α_2	89.1520°

Table 2. PD and PW for PWM

M=-1			
S1 & S2		S3 & S4	
PD (ms)	PW(ms)	PD(ms)	PW(ms)
1.3886	0.5851	0.0000	1.3886
4.9529	0.0942	1.9737	2.9792
8.0263	0.5851	5.0471	2.9792
10.0000	1.3886	8.6114	1.3886
11.9737	2.9792	11.3886	0.5851
15.0471	2.9792	14.9529	0.0942
18.6114	1.3886	18.0263	0.5851

Pulse generator for S1 and S2 (as seen in Figure 1) can be calculated into bit count as in Table 3. Pulse generator for S3 and S4 is the inverse of it is for S1 and S2. Simulation results of the pulse generators are shown in Figure 5. The simulation results match experimental results as seen in Figure 6.

Table 3. Pulse Generator Construction for S1 and S2

t (ms)	t (ms)	count	bit count
0.0000	0.0000	0	00000000000000000000
1.3886	1.3886	69430	00010000111100110110
0.5851	1.9737	98685	00011000000101111101
2.9792	4.9529	247645	00111100011101011101
0.0942	5.0471	252355	00111101100111000011
2.9792	8.0263	401315	01100001111110100011
0.5851	8.6114	430570	01101001000111101010
1.3886	10.0000	500000	01111010000100100000
1.3886	11.3886	569430	10001011000001010110
0.5851	11.9737	598685	10010010001010011101
2.9792	14.9529	747645	10110110100001111101
0.0942	15.0471	752355	10110111101011100011
2.9792	18.0263	901315	11011100000011000011
0.5851	18.6114	930570	11100011001100001010
1.3886	20.0000	1000000	11110100001001000000

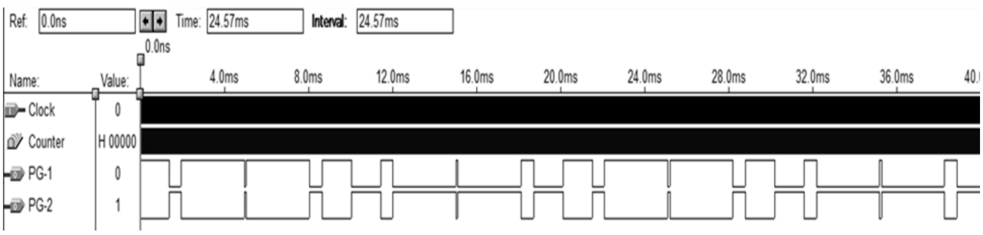

Figure 5. Simulation Results of Pulse Generators for S1, S2 (PG-1) and S3, S4 (PG-2)

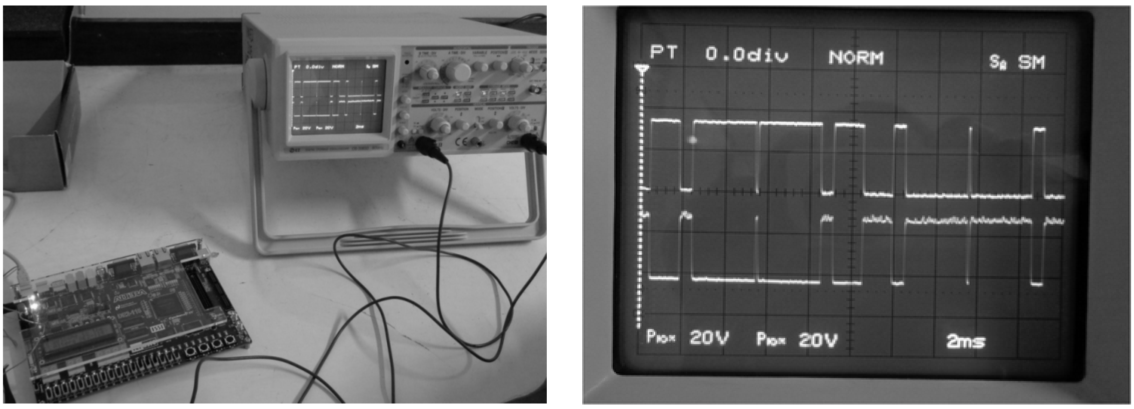

Figure 6. Experimental Results of Pulse Generators

5. Results and Analysis

The experiments are carried out with the following parameters: V_{dc}=10V, L=50mH, C=30μF, R=100Ω, and f_1=50Hz. The afformentioned evolutionary algorithm has been successfullyimplemented in the DE2-115 Cyclone IVE FPGA device. The inverter is operated in a single-phasemode and is modulated to generate a normalized fundamentalvoltage V_1= 0.73 with a fundamental frequency f_1= 50 Hz. The waveformhas threeswitching angles (N=3)in each quarter period. As the result, two harmonics are eliminated.This is evident from the measured spectrum as shown in Figure 7, wherethe third and fifth harmonics are notpresent. Even-order harmonics naturally do not exist. As shown in Figure 7, AC output voltage of the inverter without filter givesthe THD of 68.2%, but the 3rd and 5th harmonics have been eliminated completely. The voltage THD is decreased to 15.0% when a filter of L=50mH, C=30μF is inserted into the inverter. The higher order harmonics are reduced as seen in Figure 8.

.

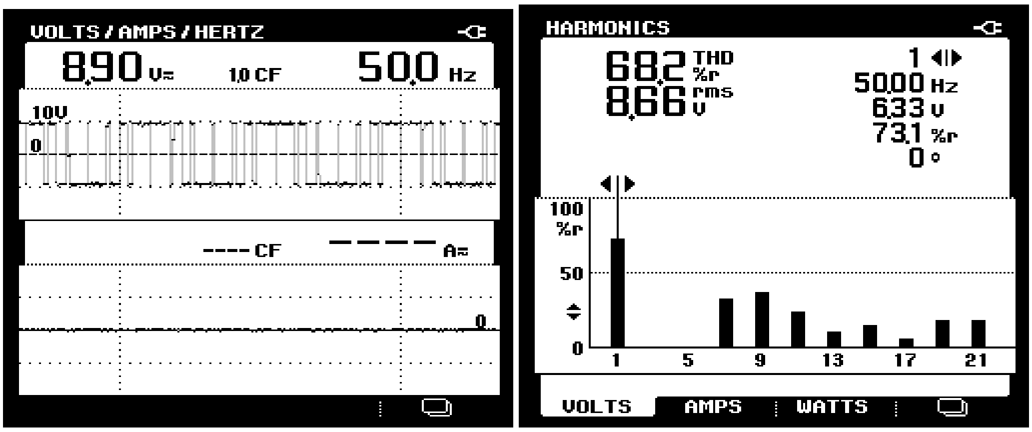

Figure 7. AC Output Voltage of the Programmed PWM Inverter without Filter

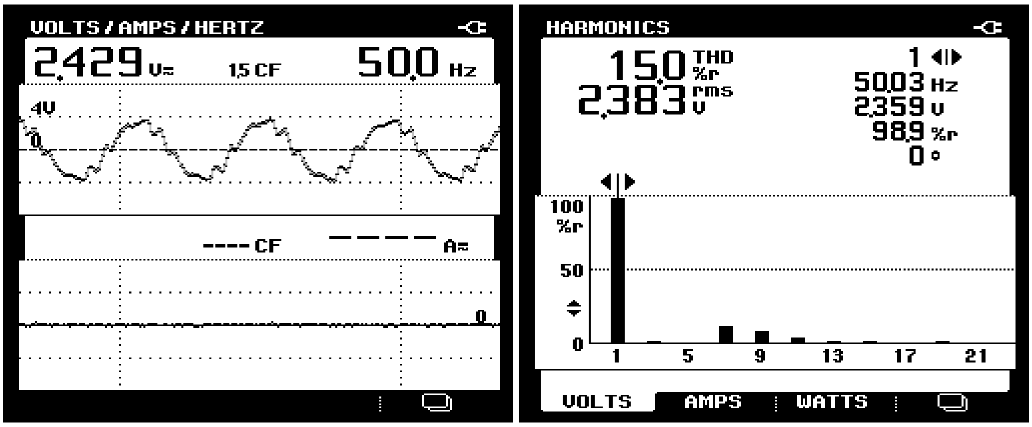

Figure 8. AC Output Voltage of the Programmed PWM Inverter without Filter L=50mH and C=30μF

6. Conclusion

Optimum switching patterns in a single-phase programmed PWM inverter is investigatedusing an evolutionary algorithm. Field programmable gate array is used to implement the switching pattern. Experimental results show that the optimum switching angles eliminate all low order harmonics of the AC output voltage of inverter. Applying an L-C filter reduces the higher order harmonics, and hence minimizes the voltage THD.

References

[1] Jaffery ZA, Chaudhry SK. *Quality Assessment of a Single-phase Resonant Inverter*. *International Journal of Power Electronics and Drive System (IJPEDS)*. 2012; 2(4): 364-370.

[2] Taufiq JA, Mellitt B, Goodman CJ. *Novel Algorithm for Generating Near Optimal PWM Waveforms for AC Traction Drives*. IEE Proceedings B on Electric Power Applications. 1986; 133(2): 85-94.

[3] EnjetiP, Lindsay JF. Solving Nonlinear Equation of Harmonic Elimination PWM in Power Control. *Electronics Letter*. 1987; 23(12): 656-657.

[4] Swift F, Kamberis A. A New Walsh Domain Technique of Harmonic Elimination and Voltage Control in Pulse Width Modulated Inverters. *IEEE Transactions on Power Electronics*. 1993; 8(2): 170-185.

[5] Liang TJ, O'Connell RM, Hoft RG.Inverter Harmonic Reduction Using Walsh Function Harmonic Elimination Method. *IEEE Transaction on Power Electronics*.1997; 12(6): 971-982.

[6] Czarkowski D, Chudnovsky DV, Chudnovsky GV, Selesnick IW. Solving the Optimal PWM for Single-Phase Inverters. *IEEE Transactions on Circuits and Systems-I: Fundamental Theory and Applications*. 2002; 49(4): 465-475.

[7] Chiasson JN, Tolbert LM, McKenzie KJ, Du Z. Control of a Multilevel Converter Using Resultant Theory. *IEEE Transactionson Control Systems Technology*. 2003; 11: 345-354.

[8] Chiasson JN, Tolbert LM, McKenzie KJ, Du Z. A Complete Solution to the Harmonic Elimination Problem. *IEEE Transactions on Power Electronics*. 2004; 19: 491-499.

[9] Vicente J, Pindado R, Martinez I. *Design Guidelines Using Selective Harmonic Elimination Advanced Method for DC-AC PWM with the Walsh Transform*. Proceedings of 7th International Conference-Workshop on Compatibility and Power Electronics (CPE). Tallinn. 2011; 220-225

[10] Sutikno T, Idris NRN, Widodo NS, Jidin A. FPGA Based a PWM Technique for Permanent Magnet AC Motor Drives. *International Journal of Reconfigurable and Embedded Systems (IJRES)*. 2012; 1(2): 43-48.

[11] Stephen V, Suresh LP. Investigation of FPGA Based PWM Control Technique for AC Motors. *International Journal of Power Electronics and Drive System (IJPEDS)*. 2013; 3(2): 193-199.