CORE
Metadata, citation and similar papers at core.ac.uk
Provided by Missouri University of Science and Technology (Missouri S&T): Scholars' Mine

Missouri University of Science and Technology
Scholars' Mine

Electrical and Computer Engineering Faculty Research & Creative Works
Electrical and Computer Engineering

01 Sep 2008

# Optimal SVM Switching for a Multilevel Multi-Phase Machine using Modified Discrete PSO

Chris M. Hutson

Ganesh K. Venayagamoorthy
*Missouri University of Science and Technology*

Keith Corzine
*Missouri University of Science and Technology*

Follow this and additional works at: https://scholarsmine.mst.edu/ele_comeng_facwork

Part of the Electrical and Computer Engineering Commons

## Recommended Citation

C. M. Hutson et al., "Optimal SVM Switching for a Multilevel Multi-Phase Machine using Modified Discrete PSO," *Proceedings of the IEEE Swarm Intelligence Symposium, 2008*, Institute of Electrical and Electronics Engineers (IEEE), Sep 2008.
The definitive version is available at https://doi.org/10.1109/SIS.2008.4668326

This Article - Conference proceedings is brought to you for free and open access by Scholars' Mine. It has been accepted for inclusion in Electrical and Computer Engineering Faculty Research & Creative Works by an authorized administrator of Scholars' Mine. This work is protected by U. S. Copyright Law. Unauthorized use including reproduction for redistribution requires the permission of the copyright holder. For more information, please contact scholarsmine@mst.edu.

2008 IEEE Swarm Intelligence Symposium
St. Louis MO USA, September 21-23, 2008

# Optimal SVM Switching for a Multilevel Multi-Phase Machine Using Modified Discrete PSO

Chris M. Hutson, Ganesh K. Venayagamoorthy, and Keith A. Corzine

*Abstract*—**This paper searches for the best possible switching sequence in a multilevel multi-phase inverter that gives the lowest amount of voltage harmonics. A modified discrete particle swarm (MDPSO) algorithm is used in an attempt to find the optimal space vector modulation switching sequence that results in the lowest voltage THD. As with typical PSO cognitive and social parameters are used to guide the search, but an additional mutation term is added to broaden the amount of area searched. The search space is the feasible solutions for the predetermined vectors at a given modulation index. Comparison of the MDPSO algorithm to an integer particle swarm optimization (IPSO) is presented for all three modulation indices tested. The resulting switching sequences found show that the MDPSO algorithm is capable of finding a minimal THD solution for all modulations indices tested. The MDPSO algorithm performed better overall than the IPSO in terms of converging to the best solution with significantly lower iterations.**

## I. Introduction

RECENTLY there has been growing interest in multi-phase motors and drives due to their increased power density and reliability. With the lower cost of semiconductor devices, electrical systems with more than three phases have become practical. Since the new semiconductor devices have only become available in the last two or three decades, research into the controls of multi-phase machines was limited. The desire to reduce harmonics in voltage, current, and torque is common for most machines. Voltage harmonics will often translate into current harmonics which causes excess losses in the machine. Torque ripple caused by current harmonics can cause vibrations from torque ripple will cause audible noise which may be undesirable. Multi-phase machines inherently reduce torque ripple [1-2], and proper motor design can also lead to lower torque ripple [3]. Therefore an optimal control system reduces all of these potentially negative problems.

Manuscript received October 9, 2001. This work was supported the National Science Foundation (NSF) CAREER Grant ECCS # 0348221 and the US Department of Education GAANN Fellowship Program.

C. M. Hutson is with the Real-Time Power and Intelligent Systems Laboratory and ECE Department, Missouri University of Science and Technology, Rolla, MO 65409 USA (phone: 314-914-1430; e-mail: cmh42c@mst.edu).

G. K. Venayagamoorthy is with the Real-Time Power and Intelligent Systems Laboratory and ECE Department, Missouri University of Science and Technology, Rolla, MO 65409 USA (e-mail: gkumar@ieee.org).

K. A. Corzine is with the Real-Time Power and Intelligent Systems Laboratory and ECE Department, Missouri University of Science and Technology, Rolla, MO 65409 USA (e-mail: keith@corzine.net).

There have been many attempts at finding an optimal switching sequence for either voltage, current, or torque harmonics in multi-phase machines. The most common methods are either space vector modulation (SVM) or an application pulse-width modulation (PWM). While attempts at SVM and PWM have had success, it is hard to claim that any have found the absolute best switching sequence solution [4-5].

Particle Swarm Optimization (PSO) is an algorithm based on the movement patterns of flocks of birds or schools of fish [6-8]. The algorithm is able to search a multi-dimensional solution space by collectively searching with different particles and communicating the best solutions found to the other particles. This communication allows for an intelligent decision on where each particle should move in an attempt at ultimately finding the best possible solution. A certain amount of randomness and weighting factors are also used in the algorithm to prevent early convergence in a local best solution instead of the global best solution.

In order to deal with discrete problems an adaptation of PSO called integer particle swarm optimization (IPSO) was created [9]. It is very similar to classical PSO except that the values used are all integers instead of real numbers. All solutions are initially set to random integers in a given range and any updates made are by adding or subtracting integers. By rounding any updates made to positions, it is possible to keep all potential solutions as integers.

An improvement on ISPO is Modified Discrete Particle Swarm Optimization (MDPSO). It is a different version of IPSO that includes mutation operators which greatly increase the amount of space searched by randomly scattering particles throughout the search process. New solutions are found by comparing the PSO's recommended movement to random numbers which then determine the new position or solution that will be used. The new solution will either be one of three previously found solutions or a new random solution.

The approach of this paper is to apply an MDPSO algorithm in order to find the optimal switching sequence for SVM control. All feasible solutions are found before the algorithm is applied. Each solution is numbered starting at one and incrementing by one with each new feasible solution. These numbered feasible solutions are the search space that the MDPSO algorithm will use in its search. The algorithm is tested on a three-level five-phase electrical system, and the results are compared to IPSO. Due to

978-1-4244-2705-5/08/$25.00 ©2008 IEEE

Authorized licensed use limited to: University of Missouri. Downloaded on February 11, 2009 at 18:05 from IEEE Xplore. Restrictions apply.

symmetry only five positions will need to be found or evaluated for all three methods.

## II. Space Vector Modulation

Space vector modulation is a method of controlling the switching sequence of an inverter by picking switching states in the quadrature-direct ($q$-$d$) plane. The $q$-$d$ plane is created by transforming the $a$-$b$-$c$ line variables into $q$-$d$-0 variables [10]. Equation (1) shows how to transfer the variables of a 5-phase electrical system into $q$-$d$-0. In this case where there are more phases than usual, a second set of $q$-$d$ variables are created making $q_1,d_1,q_2,d_2,0$ variables. The $q$-$d$ transformation also allows for different reference frames to be used for the $q$-$d$-0 variables. Common choices are the stationary, rotor, and stator reference frames. For the purpose of state selection, the stationary reference frame is necessary (whereby $\theta = 0$).

The stationary reference frame can be used to give a simple representation of all possible switching states. Figure 1 shows all the possible switching states for a three-level five-phase inverter. As the number of levels and number of phases increase so does the number of switching states. Each switching state is represented by a circle on the graph, with many states overlapping towards the center of the plot. By giving each voltage level a value it is possible to go through every possible switching state and transfer it to the $q$-$d$ voltage vector plot. In Figure 1 ground was given value 0, $V_{dc}/2$ was given 1, and $V_{dc}$ was given value 2.

A relatively simple method of implementing an SVM pattern is to create a set of vectors that approximate a unit circle in the $q1$-$d1$ plane. These vectors will approximate a circle which amounts to sinusoidal $a$-$b$-$c$ voltages. Since these vectors will not fall directly on SVM switching states, they can be approximated by a number of valid switching states. Three states are enough to approximate any of the vectors, however more can be used. This paper will use only three states to approximate a vector, but even with three states there is a large set of possible choices. There are many different combinations that exist, and there can be different ordering for each set of SVM states approximating a vector. All of these factors can affect the output voltage, current, and torque waveforms which make it very difficult to decide on a proper switching sequence.

$$\begin{bmatrix} f_{qs1} \\ f_{ds1} \\ f_{qs2} \\ f_{ds2} \\ f_{0s} \end{bmatrix} = \frac{2}{5} \begin{bmatrix} \cos(\theta) & \cos\left(\theta-\frac{2\pi}{5}\right) & \cos\left(\theta-\frac{4\pi}{5}\right) & \cos\left(\theta+\frac{4\pi}{5}\right) & \cos\left(\theta+\frac{2\pi}{5}\right) \\ \sin(\theta) & \sin\left(\theta-\frac{2\pi}{5}\right) & \sin\left(\theta-\frac{4\pi}{5}\right) & \sin\left(\theta+\frac{4\pi}{5}\right) & \sin\left(\theta+\frac{2\pi}{5}\right) \\ \cos(\theta) & \cos\left(\theta+\frac{4\pi}{5}\right) & \cos\left(\theta-\frac{2\pi}{5}\right) & \cos\left(\theta+\frac{2\pi}{5}\right) & \cos\left(\theta-\frac{4\pi}{5}\right) \\ \sin(\theta) & \sin\left(\theta+\frac{4\pi}{5}\right) & \sin\left(\theta-\frac{2\pi}{5}\right) & \sin\left(\theta+\frac{2\pi}{5}\right) & \sin\left(\theta-\frac{4\pi}{5}\right) \\ 0.5 & 0.5 & 0.5 & 0.5 & 0.5 \end{bmatrix} \begin{bmatrix} f_{as} \\ f_{bs} \\ f_{cs} \\ f_{ds} \\ f_{es} \end{bmatrix} \quad (1)$$

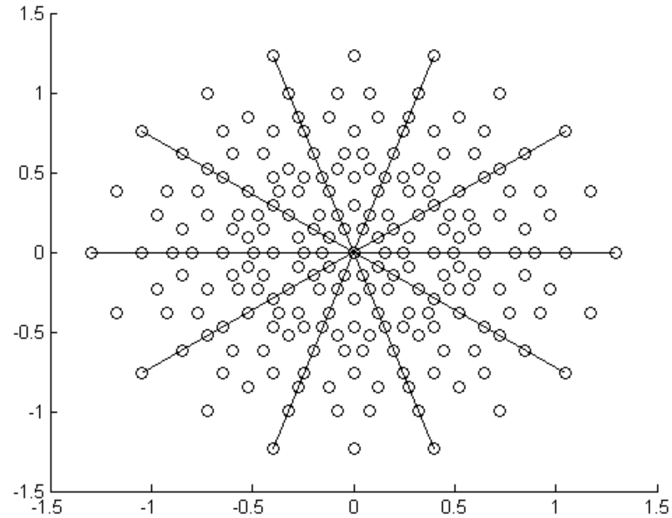


Fig. 1. Vector plot of all switching states.

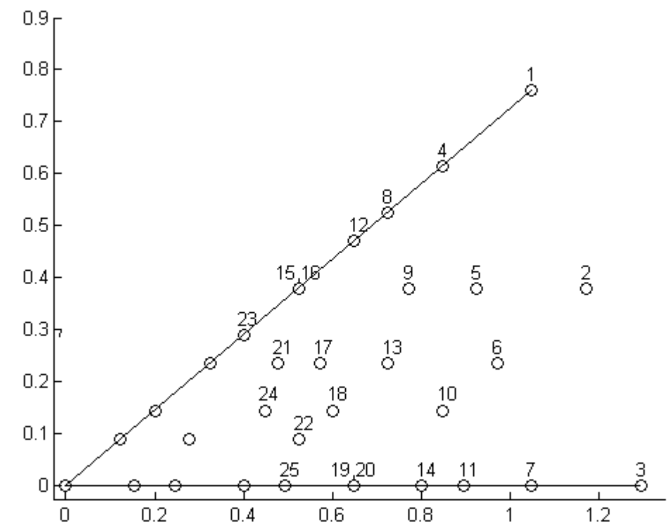


Fig. 2. Portion of $q$-$d$ plane being searched.

## III. Modified Particle Swarm Optimization

The MDPSO algorithm used is an adaptation of the PSO algorithm. The algorithm is based upon the PSO algorithm presented by Coello [11] as well as the paper by Moore [12]. A ring topology was chosen as in the other papers that make use of a local best solution for each particle instead of a global best solution.

The ring topology limits the communication between particles to only the two particles directly before and after each given particle. With this limited communication, a more localized search is achieved with the best solutions slowly propagating around the ring rather than instantly. This slower propagation allows for better resistance to local minimal solutions. Figure 3 shows how each particle connects with each other particle. The first and last particles close the ring and become neighbors.

Authorized licensed use limited to: University of Missouri. Downloaded on February 11, 2009 at 18:05 from IEEE Xplore. Restrictions apply.

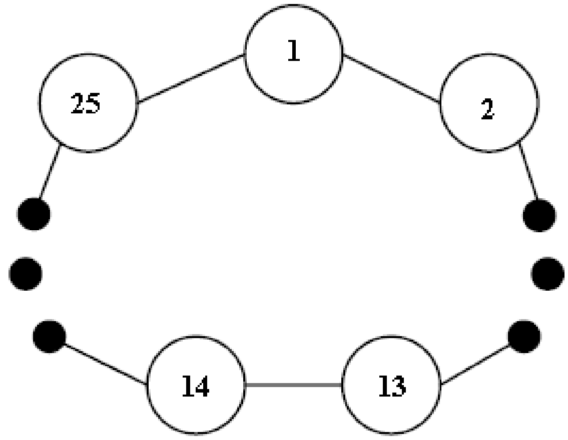


Fig. 3. Ring topology adapted from [1].

In MDPSO the velocity is first found in the usual manner and then the velocity is rounded. The resulting equation is shown in (2) where $w$ is the inertial constant, $c_1$ is the cognitive constant, and $c_2$ is the social constant. This velocity is then normalized in the interval [0-1] to be used in (3). With the normalized velocity determined, (3) compares the normalized velocity against flip functions which are randomly determined and are from the set [0:0.1:1] which begin at 0 and increments by 0.1 until 1. Two individual flip functions are calculated to be compared to both the normalized velocity and one minus the normalized velocity. If the first flip function satisfies the condition for using the *Lbest* solution then the second flip function is ignored. Otherwise the second flip function is considered for the *Pbest* solution. If both flip function conditions are not fulfilled then the old position ($pos_{old}$) is used. If the $pos_{old}$ solution is to be used then a random value in the interval [0-1] is compared to the mutation rate (4). The mutation rate used in this paper is 0.5. If the value is less than the mutation rate, then a new random integer is created in the interval of feasible solutions. For example if there were 500 combinations of states that could make up one vector then the interval is [1-500]. This mutation prevents early convergence on a local minimum by periodically throwing the particle out away from its current search space and allowing it to come back if a better solution is not found. If the random value is greater than the mutation rate, then the previous solution is used during THD evaluation. The MDPSO algorithm can be observed in the flowchart presented in Figure 4.

$$v_{j,i} = round\begin{pmatrix} w * v_{j,i} + c_1 * rand_1 * (Pbest_{j,i} - x_{j,i}) \\ + c_2 * rand_2 * (Lbest_j - x_{j,i}) \end{pmatrix} \quad (2)$$

$$pos_{new} = \begin{Bmatrix} L_{best}, & vel_n < FF_1 \\ P_{best}, & (1 - vel_n) < FF_2 \\ pos_{old}, & vel_n > FF_1, or, vel_n > FF_2 \end{Bmatrix} \quad (3)$$

$$pos_{old} = \begin{pmatrix} x_{j,i}, & rand > 0.5 \\ ceil(rand), & else \end{pmatrix} \quad (4)$$

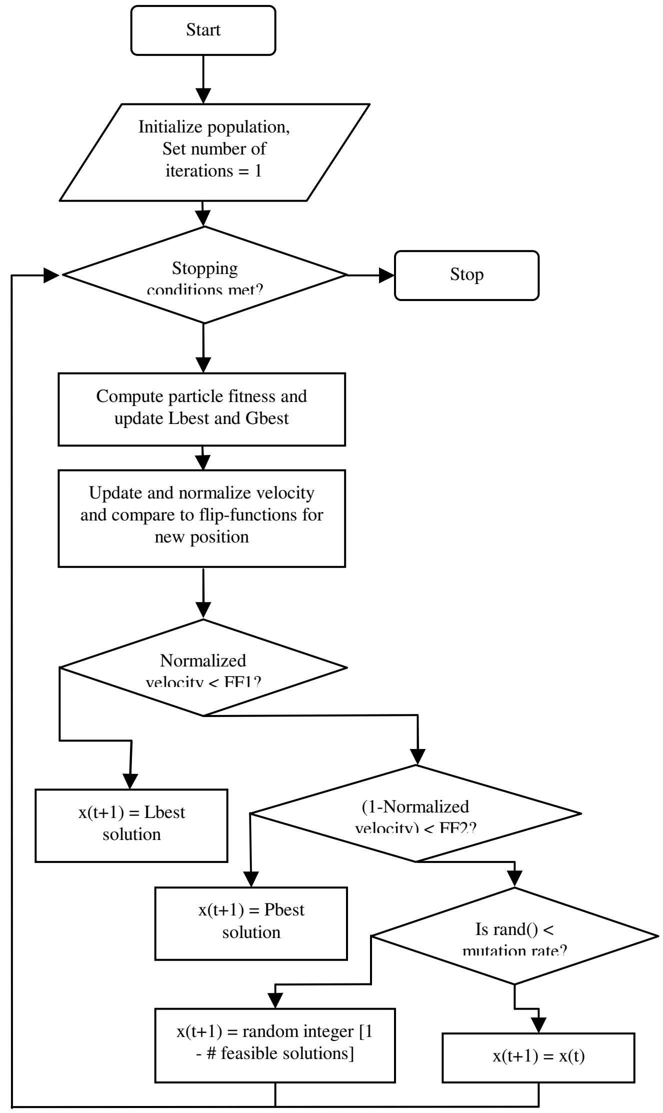


Fig. 4. MDPSO algorithm flowchart.

## IV. SVM Switching using MDPSO

Use Figure 2 shows a portion of the *q-d* vector plot which represents the area each algorithm will be searching. The switching states are numbered to give them a discrete value that the IPSO and MDPSO algorithm can work upon. Table I shows the switching values for each numbered state. The switching values are as follows: 2 for full dc voltage, 1 for half dc voltage, and 0 for ground. The *a*-phase switching state is first and the *e*-phase last. Two distinct sets of switching states are given for each number because if the values are determined for the states in the first tenth, Figure 2, as well as the second tenth of the *q-d* plane, one spot counter-clockwise from the first tenth, then all the other switching states for a full cycle can be determined. The states for the other eight tenths can be determined by performing a left circular shift to the determined states. For

Authorized licensed use limited to: University of Missouri. Downloaded on February 11, 2009 at 18:05 from IEEE Xplore. Restrictions apply.

example, if the first state in the first tenth was SVM state number 1 then the voltage levels at the inverter is 2 2 0 0 0. A circular shift left of these values results in 2 0 0 0 2. This rotated value is then the first state in the third tenth of the q-d plane. Subsequent circular shifts produce the switching values in the fifth, seventh, and ninth. The same operation is performed with the set ABCDE-2 for all of the even tenths of the q-d plane.

TABLE I
SVM STATE NUMBERS AND ABCDE SWITCHING STATES

| SVM State Number | ABCDE-1 | ABCDE-2 |
|---|---|---|
| 1 | 2 2 0 0 0 | 2 2 2 0 0 |
| 2 | 2 2 0 0 1 | 2 2 1 0 0 |
| 3 | 2 2 0 0 2 | 2 2 0 0 0 |
| 4 | 2 2 1 0 1 | 1 2 1 0 0 |
| 5 | 2 1 0 0 0 | 2 2 2 0 1 |
| 6 | 2 2 1 0 2 | 1 2 0 0 0 |
| 7 | 2 1 0 0 1 | 2 2 1 0 1 |
| 8 | 2 2 0 1 0 | 2 1 2 0 0 |
| 9 | 1 2 0 0 1 | 2 2 1 1 0 |
| 10 | 2 2 0 1 1 | 2 1 1 0 0 |
| 11 | 1 2 0 0 2 | 2 2 0 1 0 |
| 12 | 2 2 2 0 2 | 0 2 0 0 0 |
| 13 | 2 1 1 0 1 | 1 2 1 0 1 |
| 14 | 2 0 0 0 0 | 2 2 2 0 2 |
| 15 | 1 1 0 0 0 | 1 1 1 0 0 |
| 16 | 2 2 1 1 1 | 2 2 2 1 1 |
| 17 | 1 2 1 0 2 | 1 2 0 1 0 |
| 18 | 2 1 0 1 0 | 2 1 2 0 1 |
| 19 | 1 1 0 0 1 | 1 1 0 0 0 |
| 20 | 2 2 1 1 2 | 2 2 1 1 1 |
| 21 | 2 0 1 0 0 | 1 2 2 0 2 |
| 22 | 2 1 2 0 2 | 0 2 0 0 1 |
| 23 | 2 2 0 2 0 | 2 0 2 0 0 |
| 24 | 1 2 0 1 1 | 2 1 1 1 0 |
| 25 | 0 2 0 0 2 | 2 2 0 2 0 |

A range of switching states are selected from the *q-d* plane in order to limit the number of possible states. By limiting the possible states to one-tenth of the entire *q-d* plane, it is then reasonable to calculate all possible feasible representations of a vector within that tenth of the plane. By limiting to these 25 states there is also only two repeated states whereas further towards the center, there are more repeated states. A solution is feasible if the vector can be approximated by the three chosen switching states. If the vector cannot be approximated by the three switching states then that solution is removed from the search space. With all feasible solutions predetermined, the search becomes reasonable for the either IPSO or MDPSO algorithms.

Five vectors were chosen in the first tenth of the *q-d* plane in order to approximate a sine wave. The vectors were chosen such that they were centered in the first tenth with 7.2 degrees of separation from each other. This orientation left 3.6 degrees of separation of the first and fifth vector from the edges of the first tenth of the *q-d* plot. The vector positions were determined by the modulation index and the maximum possible sine wave magnitude. The maximum magnitude was decided to be the magnitude of SVM switching state 2 in Figure 2. This is the greatest magnitude an arbitrary vector could take at any phase angle and still be approximated by three of the SVM switching states. Since the goal of this project was to reduce THD at given modulation indices there was no need to account for vectors that had magnitudes greater than SVM vector 2.

In order to ensure that feasible solutions are given by the algorithm, all possible combinations of states were tested for feasibility and stored for later use by the MDPSO algorithm. Each modulation index needed its own set of feasible solutions since the target vectors were different in each case. The lowest number of feasible solutions for a single vector is 459 and the highest number of solutions for a single vector is 2912. Each particle consists of five of these feasible solutions, one for each vector approximation. These five solutions are then extrapolated to the other nine tenths of the q-d plane.

For the $w$, $c_1$, and $c_2$ values, 0.8, 2, and 2 are chosen. These are values are commonly used as a starting point and adapted to each case. Time limitations meant that other search constants could not be tested, but the best result of changing these values could only decrease the necessary number of iterations and not affect the overall accuracy. Twenty-five particles were used to find the values of the five vectors.

Since the desired outcome of this search is the lowest voltage THD, the fitness will simply be the percent ripple expressed as THD of the line-to-neutral voltage. A lower THD indicates lower voltage ripple equaling better fitness. Equation (5) transforms the line-to-ground voltages found through the MDPSO algorithm to line-to-neutral voltages. The equation for THD used is (6) and the voltages at every frequency were taken from the FFT function in Matlab. The magnitude of all frequencies were squared then summed. The squared fundamental value was then subtracted to make up the numerator of (6). The squared fundamental value was then divided out to give the THD.

$$\begin{bmatrix} v_{as} \\ v_{bs} \\ v_{cs} \\ v_{ds} \\ v_{es} \end{bmatrix} = \frac{1}{5} \begin{bmatrix} 4 & -1 & -1 & -1 & -1 \\ -1 & 4 & -1 & -1 & -1 \\ -1 & -1 & 4 & -1 & -1 \\ -1 & -1 & -1 & 4 & -1 \\ -1 & -1 & -1 & -1 & 4 \end{bmatrix} \begin{bmatrix} v_{ag} \\ v_{bg} \\ v_{cg} \\ v_{dg} \\ v_{eg} \end{bmatrix} \tag{5}$$

$$Fitness = THD = \frac{V_2^2 + V_3^2 + V_4^2 + V_5^2 + \ldots.. + V_n^2}{V_1^2} \tag{6}$$

With the large number of feasible solutions available, the MDPSO algorithm went through many iterations to get to an acceptable accuracy. This need led to a relatively high

Authorized licensed use limited to: University of Missouri. Downloaded on February 11, 2009 at 18:05 from IEEE Xplore. Restrictions apply.

computational time since the number of iterations was much higher than in Moore's paper [11]. However, the number of iterations could possibly have been reduced if more appropriate MDPSO search parameters had been determined and used.

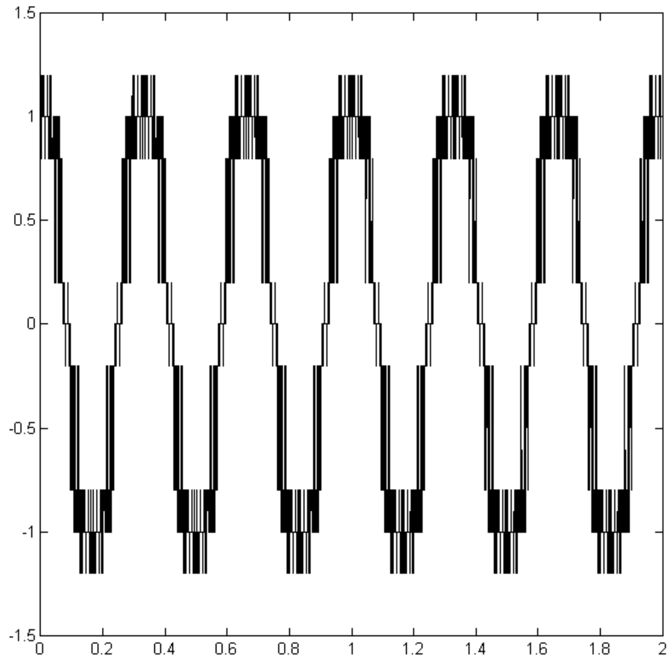


Fig. 5. Line-to-neutral voltage found by MDPSO (M = 0.9)

## V. Results

The results in Table II show the THD increasing as the modulation index is decreased. As the modulation index decreased, the magnitude of the sine wave being produced decreased. In order to approximate a sine wave of smaller magnitude, more switching from the higher states to the lower states is necessary. This increase in voltage level changes resulted in the observed increase in THD for the lower modulation indices.

The MDPSO algorithm performed very well. All three modulation indices result in low THD, and over 100 iterations the lowest THD value was found over 2/3 of the time for all attempts. The average THD over all 100 iterations is very close to the minimum THD showing that even when the algorithm fails to find the absolute minimum THD case, it will still find a solution very close to the best. Figure 5 shows one of the output voltage waveforms found for a 0.9 modulation index. It is apparent by the shape of the waveform that a small amount of harmonics are present since the waveform looks very sinusoidal with minimal variations compared to other methods such as sine triangle modulation.

The IPSO algorithm found switching sequences with low THD, but the algorithm was limited by the 7500 iteration constraint. Had the algorithm been allowed to search longer it may have been able to find the same minimum THD results. In addition to its failure in finding the same minimum THD solution in all three cases, the algorithm also had a significantly larger average THD over the 100 iteration test run. This large average shows that the algorithm failed to repeatedly find the minimum THD solution that it found.

## VI. Conclusion

This paper investigated the application of an MDPSO algorithm for selection of a modulation sequence for a three-level five-phase motor drive. The results show that the MDPSO method can consistently find better switching sequences than the IPSO algorithm. Three different modulation indices were used to verify that the algorithm. All three indices resulted in low voltage THD and fast solution convergence.

TABLE I
Results of MDPSO and IPSO for 100 Iterations

| Modulation Index | Comparisons | MDPSO | IPSO |
|---|---|---|---|
| M = 0.9 | Minimum THD | 7.1184% | 7.2015% |
| | Average THD | 7.1184% | 7.968% |
| M = 0.75 | Minimum THD | 11.7850% | 11.8372% |
| | Average THD | 11.7856% | 13.5937% |
| M = 0.6 | Minimum THD | 20.3436% | 20.9943% |
| | Average THD | 20.3583% | 22.9362% |

Authorized licensed use limited to: University of Missouri. Downloaded on February 11, 2009 at 18:05 from IEEE Xplore. Restrictions apply.

## REFERENCES


[1] L. Parsa and H.A. Toliyat, “Five-Phase Permanent Magnet Motor Drives”, *IEEE Transactions on Industry Applications*, volume 41, pages 30-37, January/February 2005.

[2] H.A. Toliyat, L. Xu and T.A. Lipo, “A Five Phase Reluctance Machine with High Specific Torque”, *IEEE Transactions on Industry Applications*, volume 28, pages 659-667, May/June 1992.

[3] L. Parsa, H.A. Toliyat, and A. Goodarzi, “Five-Phase Interior Permanent-Magnet Motors With Low Torque Pulsation,” *IEEE Transactions on Industry Applications, ,* volume 43, number 1, pages 40-46, January/February 2007.

[4] J. Huang, K.A. Corzine, C.M. Hutson, and S. Lu, "Extending Voltage Range and Reducing Torque Ripple of Five-Phase Motor drives with Added Voltage Harmonics," *IEEE Applied Power Electronics Conference and Exposition*, pages 866-872, February 2008.

[5] O. Ojo and D. Gan, "Generalized discontinuous carrier-based PWM modulation scheme for multi-phase converter-machine systems," *IEEE Industry Applications Society Conference,* volume 2, pages 1374-1381, 2005.

[6] R. C. Eberhart and J. Kennedy, "A new optimizer using particles swarm theory," *Sixth International Symposium on Micro Machine and Human Science*, pages 39-43, Nagoya, Japan, 1995.

[7] X. Hu, R.C. Eberhart, and Y. Shi, "Engineering optimization with particle swarm," *IEEE Swarm Intelligence Symposium*, pages 53-57, Indianapolis IN, 2003.

[8] J. Kennedy and R.C. Eberhart, "A discrete binary version of the particle swarm algorithm," *IEEE Conference on Systems, Man and Cybernetics*, pages 4104-4108, Orlando FL, 1997.

[9] Y. del Valle., G.K. Venayagamoorthy, S. Mohagheghi, J.C. Hernandez, R.G. Harley, "Particle Swarm Optimization: Basic Concepts, Variants and Applications in Power Systems," in *IEEE Transactions on Evolutionary Coputation,* volume 12, pages 171-195, April 2008.

[10] P.C. Krause, O. Wasynczuk, and S.D. Sudhoff, *Analysis of Electric Machinery and Drive Systems*, IEEE Press, 2002.

[11] C.A. Coello, E. H. Luna. and A. H. Aguirre, "A comparative study of encodings to design combinational logic circuits using particle swarm optimization," *NASA/DoD Conference on Evolvable Hardware*, pages 71-78, Seattle WA, 2004.

[12] P.W. Moore and G.K. Venayagamoorthy, "Evolving Digital Circuits Using Hybrid Particle Swarm Optimization and Differential Evolution," *International Journal of Neural Systems,* volume 16, number 3, pages 1-15, 2006.


Authorized licensed use limited to: University of Missouri. Downloaded on February 11, 2009 at 18:05 from IEEE Xplore. Restrictions apply.