A Quick-Simulation Tool for Induction Motor Drives Controlled Using Advanced Space-Vector-Based PWM Techniques

V. S. S. Pavan Kumar Hari and G. Narayanan
Department of Electrical Engineering, Indian Institute of Science, Bangalore 560 012, INDIA
Email: pavan@ee.iisc.ernet.in, gnar@ee.iisc.ernet.in

Abstract—Space-vector-based pulse width modulation (PWM) for a voltage source inverter (VSI) offers flexibility in terms of different switching sequences. Numerical simulation is helpful to assess the performance of a PWM method before actual implementation. A quick-simulation tool to simulate a variety of space-vector-based PWM strategies for a two-level VSI-fed squirrel cage induction motor drive is presented. The simulator is developed using C and Python programming languages, and has a graphical user interface (GUI) also. The prime focus being PWM strategies, the simulator developed is 40 times faster than MATLAB in terms of the actual time taken for a simulation. Simulation and experimental results are presented on a 5-hp ac motor drive.

I. INTRODUCTION

Voltage source inverter (VSI)-fed induction motor stands among the most sought-after industrial drive configurations. A two-level VSI realized with IGBTs is shown in Fig. 1. Pulse width modulation (PWM) of the switches is required to control the output voltage and frequency of a VSI, for a given DC bus voltage \( V_{DC} \).

Generation of PWM waveforms by comparing three-phase sinusoidal modulating waves \( (m_R, m_Y \) and \( m_B) \) against a common triangular carrier wave is popularly known as sine triangle PWM (STPWM) [1]. Considering a unipolar triangular carrier with peak value \( V_p \), the sinusoidal modulating function for R-phase is defined as

\[
m_R = 0.5 + \frac{V_m}{2V_p} \sin \omega t \tag{1}
\]

where \( V_m \) is the peak value of sinusoid. Modulating signals \( m_Y \) and \( m_B \) are phase shifted by 120° and 240°, respectively, with respect to \( m_R \).

Addition of third harmonic to the sinusoidal modulating functions increases the highest possible ac voltage output of the VSI. The DC bus utilization with such third harmonic injection PWM (THIPWM) is maximum when the amplitude of the third harmonic added is one sixth of the fundamental amplitude. This DC bus utilization is matched by the space-vector-based PWM (SVPWM) strategies [1]. Further, THIPWM also reduces the total harmonic distortion (THD) in the output current of the VSI. The THD is minimum if the third harmonic amplitude is 25% of the fundamental amplitude [2]. Conventional space vector PWM (CSVPWM) results in THD, which is quite close to this [1], [2]. Bus-clamping PWM (BCPWM) methods yield lower THD than CSVPWM at high modulation indices for a given average switching frequency [3]. Advanced bus-clamping PWM (ABCPWM) techniques, proposed recently, outperform BCPWM methods in terms of THD at high modulation indices [4]. ABCPWM methods have also been shown to have other advantages such as reduced pulsating torque [5], improved converter efficiency [2],[6] and low acoustic noise [7] under various operating conditions.

This work was supported by the Department of Heavy Industry, Government of India, under a project titled “Off-line and Real-time Simulators for Electric Vehicles / Hybrid Electric Vehicle Systems”.

Fig. 1. Two-level voltage source inverter using insulated gate bipolar transistors.

Fig. 2. Voltage vectors of a voltage source inverter. Magnitudes of the vectors are normalized with respect to DC bus voltage \( V_{DC} \). I, II, III, IV, V and VI are sectors [2].
Most SVPWM methods synthesize the desired voltage reference vector \( \mathbf{V_{REF}} \) (see Fig. 2) by applying the two nearest active vectors and the zero vector. For the reference vector shown in Fig. 2, the active vector 1 and the active vector 2 are applied for durations \( T_1 \) and \( T_2 \), respectively, as given by (2) [8], [9].

\[
T_1 = \frac{V_{REF} \sin(60^\circ - \alpha)}{\sin 60^\circ} T_s = m_R - m_Y \quad (2a) \\
T_2 = \frac{V_{REF} \sin \alpha}{\sin 60^\circ} T_s = m_Y - m_B \quad (2b)
\]

where \( T_s \) is the duration of a sub-cycle, and \( \alpha \) is the angle of reference vector from the start of a sector. The zero vector is applied for the rest of sub-cycle duration as given below.

\[
T_z = T_s - T_1 - T_2 \quad (3)
\]

CSVPWM applies the voltage vectors in a sequence starting with one zero state and ending with the other zero state in each sub-cycle; the two zero states are applied for 0.5\( T_z \) each [1]. The switching sequences 0-1-2-7 and 7-2-1-0 are applied in alternate sub-cycles in sector I [see Fig. 4(a)]. ABCPWM schemes apply only one zero state for the entire duration \( T_z \), but apply an active state twice as illustrated in Fig. 3(a) to Fig. 3(d). The ABCPWM method, shown in Fig. 4(b), employs sequences 0-1-2-1 and 1-2-1-0 in alternate sub-cycles in the first half of sector I. Sequences 7-2-1-2-7-2-1-0 are applied in the second half [4], [10]. Fig. 4(c) shows a hybrid PWM method which employs sequences 1012 and 2721 along with the conventional sequence 0127. Here, sequence 1012 is applied for \( 0^\circ < \alpha < 14^\circ \), 0127 is applied for \( 14^\circ < \alpha < 46^\circ \) and 2721 is applied for the remaining duration in sector I.

To simulate an induction motor drive fed from a pulse width modulated two-level VSI, the switching function needs to be generated for each switch in the inverter. This reduces to generation of switching functions for the three legs (\( S_R \), \( S_Y \) and \( S_B \)) since the two switches in each leg are complementary in nature.

The switching functions corresponding to continuous PWM schemes (STPWM, THIPWM and CSVPWM) and BCPWM schemes can easily be generated by comparing three-phase modulating signals with triangular carrier. However, generating \( S_R \), \( S_Y \) and \( S_B \) is quite involved in case of ABCPWM strategies. A method for determining the switching functions for ABCPWM schemes in discussed in section II.

A tool for simulation of VSI-fed induction motor drive, controlled with different PWM methods, is developed in this work. This tool is capable of simulating continuous PWM methods, BCPWM methods and a variety of ABCPWM methods. The details of the simulator developed are discussed in section III.

While there are commercial and open-source tools available (e.g. MATLAB[11], Octave[12] and SEQUEL[13]) for simulating a wide range of dynamic systems, the tool developed...
Fig. 5.

(a) Screenshot of the simulator developed.

(b) Simulation results: Direct on-line start of a 5hp induction motor.

Choice of Pulse Width Modulation (PWM) Method: ○ Sine Triangle PWM (STPWM) ○ STPWM with Third Harmonic Injection ■ Space Vector PWM (SVPWM)
Fig. 6. Motor line current waveforms at a fundamental frequency of 40Hz with different PWM techniques: (a) Simulated with MATLAB, (b) Simulated with MATLAB and (c) Experimentally measured. (1)-CSVPWM, (2)-Advanced 30° clamp PWM and (3)-Hybrid PWM. Average switching frequency \( f_{sw} \) is 1kHz.

II. Space Vector Based PWM: Realization

The first step to realize a space vector PWM is identification of the sector where \( V_{REF} \) falls in. When the magnitude and angle of \( V_{REF} \) are available, the sector can easily be determined from the angle. However, the reference is rarely available in magnitude-angle form. On the other hand, the voltage reference is usually provided by the controller as \( d \)-axis and \( q \)-axis reference in the synchronous reference frame, or as \( \alpha \)-axis reference and \( \beta \)-axis reference in the stationary reference frame, or as three-phase voltage references. The voltage reference is assumed to be available as a three-phase quantity here. (Even otherwise, the voltage references can be transformed to three-phase quantities.)

The sector can be identified by comparing \( m_R \), \( m_Y \) and \( m_B \). For example, \( m_R > m_Y > m_B \) indicates that \( V_{REF} \) is in sector I. Quite often, a sector may be divided into a number of sub-sectors, each employing a different switching sequence [see Fig. 4(b) and Fig. 4(c)]. Boundaries separating the different sub-sectors in a sector need to be represented in terms of \( m_R \), \( m_Y \) and \( m_B \). Once the sub-sector is identified, the switching sequence is known. For the four ABCPWM sequences, the switching functions can be determined as discussed below.

Consider the sequence 0121 in Fig. 3(a). PWM pulse of the single-switching phase \( S_R \) can be generated by comparing one modulating function \( c_R \) against the carrier. Two modulating functions \( c_{Y_1} \) and \( c_{Y_2} \) corresponding to \( Y \)-phase generate the PWM pulses \( S_{Y_1} \) and \( S_{Y_2} \), respectively. The final PWM output of \( Y \)-phase \( S_Y \) is an XOR function of \( S_{Y_1} \) and \( S_{Y_2} \). B-phase gets clamped to negative DC bus in sector I (i.e. \( S_B = 0 \)) with the sequence 0121. Similar procedure can be followed for the remaining sequences [Fig. 3(b) to Fig. 3(d)]. For instance, double-switching of \( R \)-phase with 1012 can be achieved by generating the pulses \( S_{R1} \) and \( S_{R2} \) (see Fig. 3(c)) and passing them through an XNOR function. All the modulating functions \((c_R, c_{R1}, c_Y, c_{Y2} \text{ etc.})\) can be expressed in terms of the dwell-times of inverter states. The dwell-times are functions of \( m_R, m_Y \) and \( m_B \) as shown by (2). For example, \( c_{B2} = (0.5T_2) + T_z \) for the sequence 2721 [see Fig. 3(d)].
### III. Simulation Tool Developed

The squirrel cage induction machine is a sixth order system when stator and rotor are modelled in their respective coordinate systems [14]. Thus, numerical simulation of an open-loop induction motor drive can be viewed as solving of a system of six ordinary differential equations (ODE) at each time step. A large number of numerical methods are available to solve a system of ODEs. In general, reduced step-size improves the accuracy of solution; Runge-Kutta methods are shown to be stable for smaller step sizes [15]. The simulator developed uses the *Explicit Runge-Kutta 4th Order* method to solve the system of ODEs. The simulator does not provide any option to the user to choose the ODE solver method. Further, the maximum integration step-size is limited to 1\(\mu s\).

A screenshot of the simulation tool developed, “motor1r” is shown in Fig. 5(a). The single-window graphical user interface (GUI) is written in PyGTK [16]. Details of the induction machine and VSI and other simulation parameters have to be entered by the user. The user has to select one among the three PWM methods, namely, STPWM, THIPWM and SVPWM. In case of SVPWM, total number of switching sequences in a sector has to be specified, which is same as the number of sub-sectors. Boundaries separating the sub-sectors have to be given in terms of angle. Finally, the sequence employed in each sub-sector has to be specified. The simulation data can be loaded from and saved to a text file. The text file is taken as an input to the main simulation routine, written in C programming language [17]. Output of the C program is plotted using NumPy and Matplotlib of the Python programming language [18]. The plotting window of Matplotlib has basic features such as x-axis zoom, y-axis zoom and rectangular zoom. Further, the plot can be saved in many formats including PNG, SVG, EPS, PDF, JPEG and EMF.

### IV. Simulation and Experimental Results

Configuration of the computer used for testing the simulation tool is Intel Core i3-3120 3.2 GHz processor with 4

---

Fig. 7. (a) Motor line-to-line voltage, (b) and (c) Harmonic spectra of line-to-line voltage at a fundamental frequency of 40Hz with different PWM techniques: (a) and (b) Simulated with motor1r, (c) Experimentally measured [PWM of R-phase minus PWM of Y-phase]. (1)-CSVPWM, (2)-Advanced 30\(^\circ\) clamp PWM and (3)-Hybrid PWM. Average switching frequency \(f_{sw}\) is 1kHz.
threads, 4GB 1333MHz DDR3 RAM and Fedora 19 Linux operating system. The experimental setup consists of a 10kVA IGBT-based two-level VSI connected to a 5hp, 400V, 50Hz, 4-pole squirrel-cage induction motor. The PWM techniques are implemented on ALTERA Cyclone II field programmable gate array (FPGA)-based digital controller [19].

Fig. 5(b) shows the simulation results when the 5hp induction motor is started direct on-line. The simulation step-size is 1µs, and the simulation end time is 1s. The actual time taken for simulation is 7.12 seconds. The total number of time points in the simulation above is 10^5. When only steady-state results are of interest (as in the case of comparing different PWM schemes), only the last significant data points need to be stored. This results in reduced time for simulation.

Simulation of the PWM methods, shown in Fig. 4, is carried out with mATrtr and MATLAB R2013a. Measured parameters of the induction machine are given as inputs. The fundamental frequency is 40Hz, and the average switching frequency fsw is 1kHz. Simulation step size is 500ns, and the simulation end time is 1.5s. Only the last 10^5 data points are stored to improve the simulation speed. Actual time taken for simulation by mATrtr and MATLAB are tabulated in Table I. It can be observed that the simulator developed performs 40 times faster than MATLAB.

Motor line current waveforms with mATrtr and MATLAB are presented in Fig. 6(a) and Fig. 6(b), respectively, for different PWM techniques. Further, experimental motor current waveforms under the same conditions are presented in Fig. 6(c). The simulation results from mATrtr are in excellent agreement with those from MATLAB and also with experimental observations.

Simulated waveforms of line-to-line voltage (v_{lY}) of the motor are shown in Fig. 7(a) for different PWM methods. Fig. 7(b) shows harmonic spectra of v_{LY} simulated with mATrtr. The measured harmonic spectra of line-to-line voltage are presented in Fig. 7(c). The simulated harmonic spectra are close to their experimental counterparts. Thus, the accuracy of the simulator is verified and the fastness is evaluated.

## V. Conclusions

A quick simulation tool for VSI-fed induction motor drives with a variety of PWM techniques is developed. The PWM techniques include continuous PWM methods, discontinuous or bus-clamping PWM methods, and advanced bus-clamping PWM methods. Generation of switching functions in case of ABCPWM schemes is explained. Simulation results from the tool developed, those from MATLAB, and experimental results on a 5-hp motor drive are presented. The simulation speed of the developed tool is 40 times that of MATLAB. The simulator developed has a simple GUI and is useful for the study of various PWM techniques.

## ACKNOWLEDGMENT

Pavan Kumar Hari would like to thank V. Seshadri Sranavan Kumar, Department of Electrical Engineering, Indian Institute of Science and Arvind Iyer, Department of Aerospace Engineering, Indian Institute of Science for their valuable inputs towards programming in C and Python.

## REFERENCES


