Trigonometry-Free SVPWM Algorithm and FPGA Application Simulation

Kai Zhang[1], Caiyun Wu[1*]

[1]*Shenyang Ligong University, Shenyang, China*

Abstract

SVPWM algorithm is widely used in motor control systems, but the traditional SVPWM algorithm is complicated in the calculation of the normal vector and the calculation of the adjacent vector action time, and the existence of irrational factors causes the problem of poor data accuracy, to solve the above shortcomings, a TF-SVPWM (Trigonometry-Free SVPWM algorithm) algorithm is proposed. First, the three-phase voltage is obtained by introducing the Clarke inverse transform, thus changing the trigonometric operation into an addition and subtraction operation, and reducing the complexity of the operation by eliminating the irrational number factor; then, the TF-SVPWM algorithm is verified in the Simulink environment; finally, the TF-SVPWM algorithm is implemented in the FPGA, and the data accuracy is improved by avoiding the calculation of irrational numbers. Theoretical analysis and simulation results show that the TF-SVPWM algorithm makes the operation of sector and adjacent vector action time simple and easy to implement on FPGA, leaving enough resources for the implementation of the whole motor control system.

Keywords

SVPWM algorithm; Clarke inverter; three-phase voltage; FPGA

1. Introduction

Space Vector Pulse Width Modulation (SVPWM) is commonly used in the vector control of permanent magnet synchronous motors and brushless DC motors because of its low waveform distortion, high voltage utilization, and easy implementation in control systems [1-2].

Currently, motor control systems implement SVPWM algorithms mostly using microcontrollers or digital signal processors. However, the functions of both rely on the sequential execution of the program to achieve. Therefore, the algorithm execution delay is large and limited execution efficiency. Compared to them, FPGA (Field-Programmable Gate Array), field-programmable gate arrays, have the advantages of parallel architecture, flexibility, and low power consumption, which are more suitable for implementing SVPWM algorithms.

Literature [3] made a detailed analysis of the traditional SVPWM algorithm and gave simulation results, but it is only applicable to the Simulink environment and cannot be directly used in the actual control system; literature [4] proposed a sector judgment method simulating human eye recognition, and the sector judgment condition was simplified, but its adjacent vector action time calculation was not simplified; literature [5] proposes an SVPWM algorithm based on line voltage calculation, using the phase angle of the line voltage converted to phase voltage, and the phase angle is used as the input parameter of the SVPWM algorithm to calculate the sector free from the complex calculation formula, but the sine and cosine of the phase angle should be calculated when the adjacent vector action time, while the calculation formula becomes more complicated due to the existence of irrational number factor.

ISCIPT2022@7th International Conference on Computer and Information Processing Technology, August 5-7, 2022, Shenyang, China
EMAIL: Kaizhang0327@163.com (KaiZhang); *Corresponding author's email: wu_cai_yun@126.com (Caiyun Wu);

Considering the above literature, the SVPWM algorithm needs to calculate the complex trigonometric function problem in the sector calculation and adjacent vector action time, and the problem of poor calculation accuracy caused by the irrational number factor is also considered. Therefore, a TF-SVPWM (Trigonometry-Free SVPWM algorithm, TF-SVPWM) algorithm is proposed in this paper. The Clarke inversion is introduced, and the resulting three-phase voltage avoids the trigonometric operation and eliminates the irrational factor. The TF-SVPWM algorithm is simulated in a Simulink environment to verify the effectiveness of the algorithm. Finally, this paper uses FPGA to implement the TF-SVPWM algorithm to avoid floating point fixed points, reduce the number of operations, and make the accuracy of operations improved. The theoretical analysis and simulation results show that the TF-SVPWM algorithm proposed in this paper is simpler than the traditional SVPWM algorithm in terms of sector calculation formula and adjacent vector action time calculation formula, and the data accuracy is higher than that of the traditional algorithm. In addition, the algorithm proposed in this paper can be implemented on FPGA, which reduces the consumption of hardware resources and has certain theoretical and engineering application value.

2. Coordinate transformation of vector control

Vector control theory mainly decouples the control equations of the AC motor, controls the magnetic chain and torque of the motor separately and independently, and controls the AC motor with the same idea as controlling the DC motor. Since the magnetic chain, current, and voltage of the AC motor are rotating vector quantities, the coordinate transformation transforms each physical quantity of the motor from the stationary three-phase coordinate system to the rotating two-phase coordinate system, and the sinusoidal quantities input to the AC motor are decoupled into the direct flow. The input parameter of the conventional SVPWM algorithm is the two-phase cross-flow after the Park inversion, but irrational numbers are introduced in the calculation of the intermediate variables. To reduce the computational complexity, the irrational number factor is eliminated to facilitate FPGA implementation. For this reason, this paper will introduce Clarke's inverse transform based on the traditional SVPWM algorithm.

2.1. Park inverter

In motor control systems, to achieve the two-phase cross-flow required by the SVPWM algorithm, the current ones are used to obtain the AC quantities by the coordinate transformation of the straight flow rate regulated by the PID link using the Park inversion. The specific method is to transform the rotating coordinate system (d-q coordinate system) voltages U_d, U_q to the two-phase stationary coordinate system (α-β coordinate system) to obtain the input parameters U_α, U_β of the SVPWM algorithm. the Park inversion matrix is shown in equation (1).

$$\begin{bmatrix} U_\alpha \\ U_\beta \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \cos\theta & \sin\theta \end{bmatrix} \begin{bmatrix} U_d \\ U_q \end{bmatrix} \quad (1)$$

where: Ud, Uq is the voltages in the d-axis and q-axis, and θ is the angle through which the rotor turns.

The conventional SVPWM algorithm uses the voltages U_α and U_β of the Park inversion to calculate the sector and adjacent vector action time, thus introducing irrational numbers, which makes the formula for calculating the sector and adjacent vector action time complicated.

To eliminate the irrational numbers, simplify the sector judgment conditions, reduce the complexity of calculating the sector and adjacent vector action times, and avoid the degradation of accuracy when processing data in FPGA, Clarke's inverse transform is introduced in this paper.

2.2. Clarke inverters

After the Park inversion, the voltages U_α and U_β are again coordinate transformation. The coordinate transformation is carried out from a two-phase coordinate system (α-β coordinate system) to a three-phase coordinate system (a-b-c coordinate system), The three-phase voltages U_A, U_B, and U_C calculate the intermediate variables eliminating the irrational number factor and reducing the number of operations. The Clarke inverse transformation matrix is shown in equation (2).

$$\begin{bmatrix} U_A \\ U_B \\ U_C \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} \\ -\frac{1}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} U_\alpha \\ U_\beta \end{bmatrix} \tag{2}$$

where: U_α and U_β are the voltages of the α-axis and β-axis.

Although irrational numbers exist in Eq. (3), the TF-SVPWM algorithm does not affect the accuracy of the algorithm because it directly uses the re-obtained U_A, U_B, and U_C, avoiding the problem of irrational numbers introduced by the two-phase voltage calculation.

3. TF-SVPWM algorithm

The coordinate transformation is studied in the previous paper, and the conventional algorithm is studied in the following, and then the three-phase voltage obtained from the Clarke inversion is applied to the SVPWM algorithm, and the TF-SVPWM algorithm is obtained after theoretical derivation.

3.1. Traditional SVPWM algorithm

According to the literature [6], three variables A, B, and C are defined in the calculation of sectors in equation (3). then the variable N=4C+2B+A is defined, if A>0, N=1; if B>0, N=2; if C>0, N=4. The values of six sectors are obtained from the combination of A, B, and C.

$$\begin{cases} A = U_\beta \\ B = -\frac{\sqrt{3}}{2} U_\alpha - \frac{1}{2} U_\beta \\ C = \frac{\sqrt{3}}{2} U_\alpha - \frac{1}{2} U_\beta \end{cases} \tag{3}$$

Calculating the adjacent vector action time requires defining the three variables X, Y, and Z of equation (4).

$$\begin{cases} X = \frac{\sqrt{3} T_{PWM}}{U_{dc}} U_\beta \\ Y = \frac{\sqrt{3} T_{PWM}}{U_{dc}} \left(\frac{\sqrt{3}}{2} U_\beta + \frac{1}{2} U_\alpha\right) \\ Y = \frac{\sqrt{3} T_{PWM}}{U_{dc}} \left(-\frac{\sqrt{3}}{2} U_\beta + \frac{1}{2} U_\alpha\right) \end{cases} \tag{4}$$

where: T_{PWM} is the period of the PWM wave, U_{dc} is the bus voltage, and U_α and U_β are the voltages of the α-axis and β-axis.

From Eqs. (3) and (4), it can be seen that the sector calculation formula and the adjacent vector action time calculation formula have complicated structures and are not easy to implement on FPGA. For this reason, the TF-SVPWM algorithm is proposed below.

3.2. TF-SVPWM algorithm

The instantaneous value expression of the three-phase symmetric sine voltage is shown in equation (5).

$$\begin{cases} U_A = U_m \cos \omega t \\ U_B = U_m \cos(\omega t - \frac{2}{3}\pi) \\ U_C = U_m \cos(\omega t + \frac{2}{3}\pi) \end{cases} \tag{5}$$

where: Um is the peak value of phase voltage and ω is the angular frequency of phase voltage.

The vector of the three-phase voltage synthesis is shown in equation (6). The synthesized vector amplitude is equated to the output voltage amplitude, so the rotated vector after synthesis is multiplied by a factor of 2/3.

$$\begin{aligned} U_{out} &= \frac{2}{3}(U_A + U_B e^{j\frac{2}{3}\pi} + U_C e^{j\frac{2}{3}\pi}) \\ &= U_\alpha + jU_\beta \end{aligned} \tag{6}$$

where: $U_\beta = \frac{1}{\sqrt{3}}(U_B - U_C)$, $U_\beta = \frac{1}{\sqrt{3}}(U_B - U_C)$.

Redefine A, B, and C as shown in equation (7).

$$\begin{cases} A = U_B - U_C \\ B = U_A - U_C \\ C = U_B - U_A \end{cases} \tag{7}$$

Substituting the expressions of U_α and U_β into equation (4) yields the expressions of variables X, Y, and Z as shown in equation (8).

$$\begin{cases} X = \frac{T_{PWM}}{U_{dc}}(U_B - U_C) \\ Y = \frac{T_{PWM}}{U_{dc}}(U_A - U_C) \\ Z = \frac{T_{PWM}}{U_{dc}}(U_B - U_A) \end{cases} \tag{8}$$

From Eqs. (7) and (8), it can be seen that the trigonometric function operation is avoided, which reduces the computational effort; the structure of the sector calculation formula and the adjacent vector action time calculation formula obtained from the three-phase voltage is simpler compared to the two-phase voltage, and there are no irrational numbers in the formula thus improving the data accuracy. Therefore, the TF-SVPWM algorithm is easier to implement on FPGA than the traditional algorithm.

In this section, the improved SVPWM algorithm is obtained after theoretical derivation, and the TF-SVPWM algorithm is simulated in the Simulink environment below.

3.3. TF-SVPWM algorithm simulation

According to Equation (7), the simulation results of sectors are obtained as shown in Figure 1. From Figure 1, it can be seen that the sectors appear in the order of 3-1-5-4-6-2, which is by the law of sector appearance. Then, the duty cycle calculation method in the literature [6] is used in the TF-SVPWM algorithm, and the simulation results of the duty cycle are shown in Fig. 2, which are similar to the conventional algorithm as saddle wave. Therefore, the variables A, B, and C defined by the TF-SVPWM algorithm are correct.

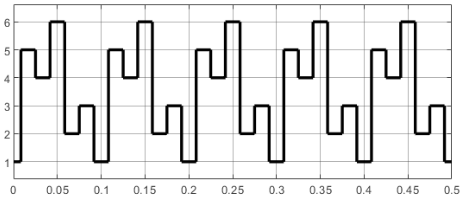

Figure.1: sector simulation results

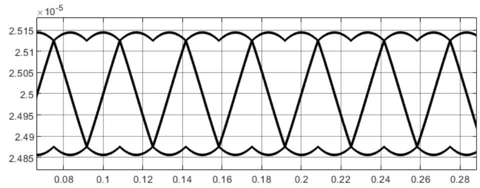

Figure.2: duty cycle

It can be seen from Fig. 1 and Fig. 2 that the sector and duty cycle curves obtained from the simulation of the TF-SVPWM algorithm are similar to those of the literature [6]; therefore, the algorithm proposed in this paper is correct. In the following, the coordinate transformation is implemented by FPGA, and then the TF-SVPWM algorithm is implemented.

4. FPGA Implementation of Coordinate Transformation

The algorithm proposed in this paper solves the problem of needing to calculate trigonometric functions in traditional algorithms, avoids calculating floating point numbers on FPGAs, and has higher data accuracy. the TF-SVPWM algorithm has a simple structure compared to traditional algorithms, is easier to implement on FPGA, and consumes fewer hardware resources.

4.1. FPGA implementation of the Park inverter

In this paper, we use the Cordic algorithm to realize the sine and cosine values needed for Park's inverse conversion and follow equation (1) to directly call the multiplier IP core to realize the multiplication of two direct flows with sine and cosine functions. The simulation results are shown in Figure 3, Sin and Cos are the output sine and cosine values, and finally, the two intersecting flows are obtained U_α and U_β.

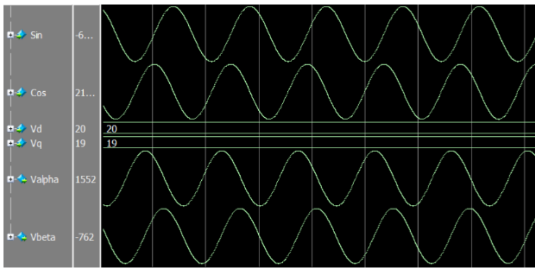

Figure.3: Park inverter conversion ModelSim simulation results

4.2. FPGA implementation of the Clarke inverter

To save the logic resources of the chip, the Clarke inversion is implemented by replacing the multiplication with a shift operation $\frac{\sqrt{3}}{2}$ by using equation (9). The simulation results are shown in Figure 4, and the three-phase voltages U_A, U_B, and U_C are obtained.

$$\frac{\sqrt{3}}{2} \approx 2^{-1} + 2^{-2} + 2^{-4} + 2^{-5} + 2^{-6} + 2^{-8} + 2^{-9} + 2^{-10} + 2^{-11} \tag{9}$$

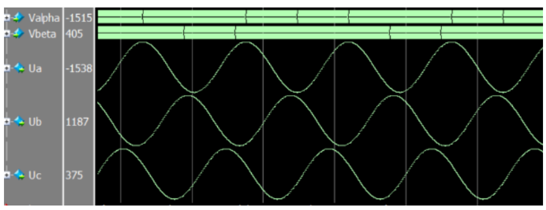

Figure.4: Clarke inversion simulation results

5. TF-SVPWM algorithm FPGA implementation

The core algorithm of TF-SVPWM consists of a sector calculation module, adjacent vector action time calculation module, three-way duty cycle calculation module, PWM signal generation module, and dead time setting module. From Figure 5, we can see that the sectors appear in the order of 3-1-5-4-6-2, which is to the design requirements; the simulation results of the three-way duty cycle are all saddle waves as shown in Figure 6, therefore, it is verified that the TF-SVPWM algorithm proposed in this paper is correct.

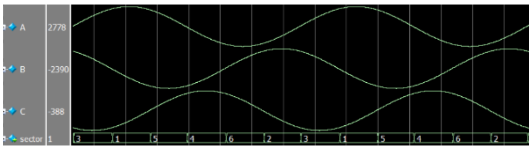

Figure.5: simulation results for sectors

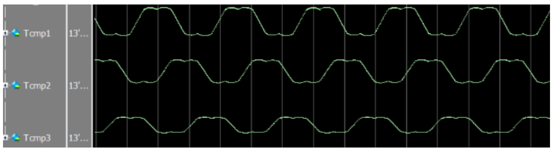

Figure.6: Three-way duty cycle

The six PWM waves after adding the dead time are shown in Figure 7. The three PWM waves of the upper bridge arm and the three PWM waves of the lower bridge arm do not conduct at the same time, indicating that the dead time added in the algorithm is effective.

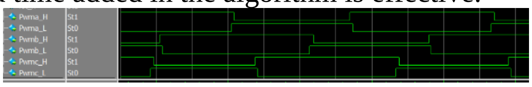

Figure.7: PWM wave with headband

For the traditional SVPWM core algorithm and TF-SVPWM core algorithm according to the top-down idea, the Rtl code is written on Vivado software using Verilog hardware description language for each module, the irrational numbers in the traditional SVPWM algorithm are in Q14 format, and the timing simulation is completed in ModelSim tool; Xilinx Zynq-7000 series of xc7z010clg400-1 model FPGA from Xilinx is selected to complete the layout wiring and comprehensive analysis; the resources consumed by TF-SVPWM algorithm and traditional SVPWM algorithm are shown in Table 1, the TF-SVPWM algorithm proposed in this paper consumes 258 lookup tables and 146 registers, compared with the traditional algorithm, the algorithm proposed in this paper saves more hardware resources.

Table.1

resource consumption of the SVPWM algorithm

Algorithm Type	Traditional Algorithm	TF-SVPWM Algorithm
LUTs	635	258
Register	383	146

6. Conclusion

To address the problem that the conventional SVPWM algorithm requires the calculation of complex trigonometric functions for the sector and adjacent vector action times, this paper proposes the TF-SVPWM algorithm and FPGA implementation. Firstly, the three-phase voltage is obtained by using the Clarke inversion, which eliminates the irrational number factor, simplifies the formula structure, reduces the computational complexity of the sector and the adjacent vector action time, and improves the data accuracy. Then, the TF-SVPWM algorithm is simulated in the Simulink environment and implemented on the FPGA platform to realize the TF-SVPWM algorithm without fixed-point operation. In addition, the proposed algorithm has advantages in terms of hardware resource consumption and data accuracy; it leaves enough hardware resources for the implementation of the whole motor control system; using the TF-SVPWM algorithm proposed in this paper for vector control of permanent magnet synchronous motor, and then completing the whole motor closed-loop control system is the next step to be done.

7. References

[1] L. Yu, C. Wang, H. Shi, R. Xin, and L. Wang, "Simulation of PMSM field-oriented control based on SVPWM," 2017 29th Chinese Control And Decision Conference (CCDC), 2017, pp. 7407-7411, doi: 10.1109/CCDC.2017.7978524.

[2] Vinayaka K U and Priya S., "Analysis of BLDC motor performance using space vector pulse width modulation," 2016 International Conference on Computation of Power, Energy Information and Communication (ICCPEIC), 2016, pp. 549-552, doi:10.1109/ICCPEIC.2016.7557292.

[3] Xin Zhou and J. X. Jin, "A universal simulation model for motor control based on SVPWM," 2011 International Conference on Applied Superconductivity and Electromagnetic Devices,2011, pp. 212215, doi:10.1109/ASEMD.2011.6145103.

[4] Q. Wang, P. Chen, F. Deng, M. Cheng, and G. Buja, "A Novel Method Simulating Human Eye Recognition for Sector Judgement of SVPWM Algorithm," in IEEE Access, vol. 8, pp. 90216-90224, 2020, DOI: 10.1109/ACCESS.2020.2993806.

[5] W. Zhang, Y. Chen and J. Wang, "A Novel Engineering Implementation of SVPWM Based on onLine Voltage Information, "2018 IEEE 2nd International Conference on Circuits, System and Simulation (ICCSS), 2018, pp. 52-55, doi:10.1109/CIRSYSSIM.2018.8525917.

[6] X.Liang, M. Luo, B. Zhao, Y. Yuan and Z. Chen, "Research and implementation of SVPWM control algorithm based on FPGA," 2016 IEEE International Conference on Mechatronics and Automation, 2016, pp. 22-26, doi:10.1109/ICMA.2016.7558528.