Designing the U/F Control Circuit and Space Vector Modulation for Three-Phase Inverter based on DSPIC30F4011

Pham Van Minh*, Vu Anh Nam*, Nguyen Duc Dien**, Dinh Thi Hang*, Mai Van Duy*

*University of Economics - Technology for Industries, Viet Nam

Abstract

Three-phase inverters using space vector modulation (SVPWM) are among the most popular inverter architectures available today. The inverter design and manufacturing process usually goes through simulation, hardware design, and programming on the microcontroller and calibration. The paper presents the construction and fabrication of the force circuit and control circuit that implements the SVPWM spatial vector modulation algorithm and the U/F speed control method on the microcontroller for the inverter in detail. U/F control and SVPWM spatial vector modulation algorithm are tested on dsPIC30F4011 microcontroller and a small power inverter model to illustrate implementation and execution.

Keywords — SVPWM, Inverter, dsPIC30F4011.

I. INTRODUCTION

Spatial vector modulation (SVPWM) for three-phase inverters has led to constant evolution in inverter engineering, developing a generation of inverters that allow soft control of frequency, amplitude, and phase angle. Based on the SVPWM modulation platform for two-level three-phase inverters, there are recently inverters with multilevel SVPWM architecture and algorithm. However, SVPWM technology for two-level three-phase inverters is still the basic structure and is widely used today due to its simple structure; reliable performance; the implementation of Modulation algorithm is not too complicated. There have been many publications on SVPWM for two-level three-phase inverters [1-3]. However, most of the above works are only published in theory, simulation calculations, and there is mention of real system implementation, most of the authors only give experimental results. The detailed problems of the technique of implementing SVPWM on a microcontroller system have been "obscured" and rarely mentioned. This paper presents how to implement SVPWM for two-level three-phase inverters on the dsPIC30F4011 microcontroller while simultaneously implementing the U/F control algorithm.

The rest of the paper is organized as follows. The second section introduces the hardware structure. The third section is the spatial vector modulation algorithm. The fourth section builds the driver program on the dsPIC 30F4011. The experimental results will be given in the fifth section. Finally, the sixth section is the conclusions of this paper.

II. HARDWARE STRUCTURE

The structure diagram performing U/F control and SVPWM spatial vector modulation for the inverter is illustrated in figure 1. The U/F unit will perform the required frequency signal reception and value conversion of the corresponding voltage. The Converter block converts the required frequency and voltage signals into the amplitude and angle of rotation for the spatial vector modulation block. The U/F, Converter, SVPWM conversion blocks will be done by dsPIC30F4011. A U/F frequency control principle for a three-phase asynchronous motor in U(f) is implemented simply as U/F = const [4].

![Figure 1: Structure diagram](http://example.com/figure1.png)

A. Build a Force Circuit

The power supply unit for the circuit is responsible for creating the following DC sources: 300VDC, 15VDC power supply for the power circuit, and 5VDC power supply for the control circuit. The 300VDC source circuit has stages of filtering noise, protecting against short-circuit via PTC300; 15VDC power circuit following Flyback pulse source model. The principle of the power block circuit is designed, as shown in Figure 2.

The power circuit uses Mitsubishi's PS219B4-CST power IC [5]; Power ICs have built-in 6 IGBTs, the drive circuits provide power to the upper and lower floors. The schematic diagram of the internal principle of the power IC is illustrated in Figure 3.
Figure 2: Diagram of the source circuit principle

Figure 3: Block diagram of the PS219B4 power IC

The power circuit uses the overcurrent protection signals directly from the power IC and indirect overcurrent to report to the microcontroller. The power block principle circuit is designed in Figure 4.
The complete force circuit is constructed, as shown in Figure 5.

**B. Control Circuit Construction**

Pin diagram of the dsPIC30F4011 microcontroller as shown in Figure 6, in which the pins of PWM signal output for the force circuit are PWM1L/RE0 (38), PWM1H/RE1 (37), PWM2L/RE2 (36), PWM2H/RE3 (35), PWM3L/RE4 (34), PWM3H/RE5 (33).

The controller uses the dsPIC30F4011 microcontroller, one of the most advanced and modern microcontrollers available today [6]. dsPIC30F4011 has specialized functions to control the motor, implement control algorithms, especially since the price is getting cheaper. The dsPIC30F4011 microcontroller has the same basic parameters as Table 1.
The control circuit has the function of implementing U/F control rules and spatial vector modulation algorithm. For the convenience of the test run, the control circuit features push buttons and a 16x2 LCD. Principle of the controller circuit using the dsPIC30F4011 microcontroller KIT, as shown in Figure 7.

![Figure 7: Principle of the control circuit](image1)

The experimental circuit picture of KIT dsPIC30F34011 is shown in Figure 8.

![Figure 8: Experimental model of the control circuit](image2)

### III. SPACE VECTOR MODULATION ALGORITHM

Document [7] has shown that the two-level voltage source inverse control (Figure 9) follows the spatial vector modulation method. In this paper, SVPWM drives are considered a single stage with the rotation angle Vs, input signal and the phase angle θ. Figure 10 shows the voltage vector Vs and the modulation space, where the amplitude modulation coefficient γ = [0.1], and Figure 11 illustrates the modulation of the vector Vs at sector 1.

---

**Table 1: Resources of the dsPIC30F microcontroller series**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>dsPIC30F2010</td>
<td>28</td>
<td>12K/4K</td>
<td>512</td>
<td>1024</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>6 ch</td>
<td>6 ch</td>
<td>Yes</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>dsPIC30F3010</td>
<td>28</td>
<td>24K/8K</td>
<td>1024</td>
<td>5</td>
<td>4</td>
<td>2</td>
<td>6 ch</td>
<td>6 ch</td>
<td>Yes</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>dsPIC30F4012</td>
<td>28</td>
<td>48K/16K</td>
<td>2048</td>
<td>1024</td>
<td>5</td>
<td>4</td>
<td>6 ch</td>
<td>6 ch</td>
<td>Yes</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>dsPIC30F3011</td>
<td>40</td>
<td>24K/8K</td>
<td>1024</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>6 ch</td>
<td>9 ch</td>
<td>Yes</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>dsPIC30F4011</td>
<td>40</td>
<td>48K/16K</td>
<td>2048</td>
<td>1024</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>6 ch</td>
<td>9 ch</td>
<td>Yes</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>dsPIC30F5015</td>
<td>64</td>
<td>66K/22K</td>
<td>2048</td>
<td>1024</td>
<td>5</td>
<td>4</td>
<td>8</td>
<td>16 ch</td>
<td>Yes</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>dsPIC30F6010</td>
<td>80</td>
<td>144K/48K</td>
<td>8192</td>
<td>4096</td>
<td>5</td>
<td>8</td>
<td>8</td>
<td>8 ch</td>
<td>16 ch</td>
<td>Yes</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>
The voltage vector position \( \mathbf{v}_s \) can be any of the sectors on the static coordinate system \( \alpha \beta \). Vector of the applied voltage will be synthesized from the known reference vectors of the inverse circuit. The idea of the spatial vector modulation method is to create a continuous displacement of the equivalent space vector of the inverter voltage vector (vector \( \mathbf{v}_s \)) on a circular orbit. With the regular displacement of the spatial vector on a circular orbit, the high-order harmonics are eliminated. The relationship between the control signal and the output amplitude becomes linear.

Steps to perform spatial vector modulation:
- Step 1: Locate the voltage vector \( \mathbf{v}_s \)
- Step 2: Determine the time (or modulation factor) to execute two reference vectors in each \( T \) modulation cycle.
- Step 3: Calculate the time (or modulation coefficient) to perform the reverse valve branch in each cycle \( T \)

The modulation scheme and algorithm are performed in Figure 12 and Table 2.

![Figure 9: Diagram of the principle of the independent reverse flow force circuit of the three-phase voltage source](image9)

![Figure 10: Modulation space \( V_s \)](image10)

![Figure 11: Modulation of vector \( V_s \) in sector 1](image11)

![Figure 12: SVPWM modulation algorithm](image12)

<table>
<thead>
<tr>
<th>Sector</th>
<th>Upper Switches ( (S_1, S_2, S_3) )</th>
<th>Lower Switches ( (S_4, S_5, S_6) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( S_1 = T_1 + T_2 + T_0 / 2 )</td>
<td>( S_4 = T_0 / 2 )</td>
</tr>
<tr>
<td></td>
<td>( S_3 = T_2 + T_0 / 2 )</td>
<td>( S_6 = T_1 + T_2 + T_0 / 2 )</td>
</tr>
<tr>
<td></td>
<td>( S_5 = T_0 / 2 )</td>
<td>( S_7 = T_1 + T_2 + T_0 / 2 )</td>
</tr>
<tr>
<td>2</td>
<td>( S_1 = T_1 + T_2 + T_0 / 2 )</td>
<td>( S_4 = T_0 / 2 )</td>
</tr>
<tr>
<td></td>
<td>( S_3 = T_2 + T_0 / 2 )</td>
<td>( S_6 = T_1 + T_2 + T_0 / 2 )</td>
</tr>
<tr>
<td></td>
<td>( S_5 = T_0 / 2 )</td>
<td>( S_7 = T_1 + T_2 + T_0 / 2 )</td>
</tr>
<tr>
<td>3</td>
<td>( S_1 = T_1 / 2 )</td>
<td>( S_4 = T_0 / 2 )</td>
</tr>
<tr>
<td></td>
<td>( S_3 = T_2 + T_0 / 2 )</td>
<td>( S_6 = T_1 + T_2 + T_0 / 2 )</td>
</tr>
<tr>
<td></td>
<td>( S_5 = T_0 / 2 )</td>
<td>( S_7 = T_1 + T_2 + T_0 / 2 )</td>
</tr>
<tr>
<td>4</td>
<td>( S_1 = T_1 + T_2 + T_0 / 2 )</td>
<td>( S_4 = T_0 / 2 )</td>
</tr>
<tr>
<td></td>
<td>( S_3 = T_2 + T_0 / 2 )</td>
<td>( S_6 = T_1 + T_2 + T_0 / 2 )</td>
</tr>
<tr>
<td></td>
<td>( S_5 = T_0 / 2 )</td>
<td>( S_7 = T_1 + T_2 + T_0 / 2 )</td>
</tr>
<tr>
<td>5</td>
<td>( S_1 = T_2 + T_0 / 2 )</td>
<td>( S_4 = T_0 / 2 )</td>
</tr>
<tr>
<td></td>
<td>( S_3 = T_1 + T_2 + T_0 / 2 )</td>
<td>( S_6 = T_1 + T_2 + T_0 / 2 )</td>
</tr>
<tr>
<td></td>
<td>( S_5 = T_0 / 2 )</td>
<td>( S_7 = T_1 + T_2 + T_0 / 2 )</td>
</tr>
<tr>
<td>6</td>
<td>( S_1 = T_1 + T_2 + T_0 / 2 )</td>
<td>( S_4 = T_0 / 2 )</td>
</tr>
<tr>
<td></td>
<td>( S_3 = T_2 + T_0 / 2 )</td>
<td>( S_6 = T_1 + T_2 + T_0 / 2 )</td>
</tr>
<tr>
<td></td>
<td>( S_5 = T_0 / 2 )</td>
<td>( S_7 = T_1 + T_2 + T_0 / 2 )</td>
</tr>
</tbody>
</table>

IV. BUILD DRIVER PROGRAM ON THE DSPIC30F4011

Programming for the dsPIC30F4011 microcontroller using MPLAB X IDE v3.40 programming software and XC16 compiler.

A. Description of MCPWM Settings

The dsPIC 30F4011 microcontroller has a dedicated function block [8]: Motor control PWM. The motor control PWM (MCPWM) module simplifies generating multiple, synchronized pulse width modulated outputs. In particular, the following power and motion control applications are supported: Three-Phase AC Induction Motor, Switched Reluctance (SR)
Motor, Brushless DC (BLDC) Motor, Uninterruptable Power Supply (UPS).

MCPWM unit has many operation modes: Edge aligned mode, Center aligned mode. Center aligned mode with double updates, Single event mode. In this article using the center-aligned mode, this mode’s principle of operation is illustrated in Figure 13.

Hình 13: Center aligned PWM

The issues we need to consider when setting the operation mode of MCPWM unit in modulation are as follows:

- Operating in interrupt mode, the interval between timer interrupts is the modulation value ($T_1, T_2, T_0$) update cycle.
- Set the conduction time between two valves on a series branch (dead-time).
- The modulation carrier scheme must be performed in isosceles triangle mode (a counter cycle consists of two counting steps up then counting down).
- Counting pulse cycle setting includes counting count pulses, balancing modulation frequency value, and modulation resolution.

Determining the dead-time should be based on the ON/OFF time of the IGBT capacity valve provided by the manufacturer. Most of the time parameters of Turn-ON and Turn-OFF delay of IGBTs are below 100 ns and 500 ns. However, to increase the factor of safety, the dead-time must be chosen greater than the sum of Turn-ON and Turn-OFF.

Experience has shown that it is acceptable to choose a dead-time ratio between 1.5s and 4µs. In this paper, the authors use a dead-time time of 2.5µs. The DTCON2 register contains control bits that allow the two programmable dead times to be assigned to each of the complementary outputs. There are two dead time assignment control bits for each of the complementary outputs.

The MCPWM unit setup program generates a 5kHz carrier frequency signal as follows:

```
void MCPWM_Init(void)
{
    PTPER = FCY/(2*PWM_FREQ) - 1;
    SEVTCMP = PTPER;
    PWMCON1 = 0;
    PWMCON1bits.PEN1H = 1;
    PWMCON1bits.PEN1L = 1;
    PWMCON1bits.PEN2H = 1;
    PWMCON1bits.PEN2L = 1;
    PWMCON1bits.PEN3H = 1;
    PWMCON1bits.PEN3L = 1;
    PWMCON1bits.PMOD1 = 0;
    PWMCON1bits.PMOD2 = 0;
    PWMCON1bits.PMOD3 = 0;
    OVDCON = 0xFF00;
    PDC1 = 0;
    PDC2 = 0;
    PDC3 = 0;
    DTCON1 = DEADTIME;
    PWMCON2 = 0x0F00;
    PTCON = 0x8002;
}
```

### B. Standardize Programming Data

Before programming, SVPWM modulation algorithm needs to standardize data. This stage is the quantization of data for calculation on the MCU. It is necessary to standardize 4 parameters: time variable $T$, modulation coefficient, sine function value in the range from 0 to $\pi/3$; rotation angle $\theta$.

Table 3: Standardization of data

<table>
<thead>
<tr>
<th>NO.</th>
<th>Parameters</th>
<th>Data range</th>
<th>Standardized data range</th>
<th>Bit number</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Value</td>
<td>Unit</td>
<td>Hex</td>
</tr>
<tr>
<td>1</td>
<td>Time</td>
<td>[0, $T_{pwm}$] s</td>
<td>[0x0000, 0x03FF]</td>
<td>[0, 1023]</td>
</tr>
<tr>
<td>2</td>
<td>Modulation factor $\gamma$</td>
<td>[0, 1]</td>
<td>[0x0000, 0x7FFF]</td>
<td>[0, 32767]</td>
</tr>
<tr>
<td>3</td>
<td>Sine function</td>
<td>[0, 1]</td>
<td>[0x0000, 0x7FFF]</td>
<td>[0, 32767]</td>
</tr>
<tr>
<td>4</td>
<td>Phase angle $\phi$</td>
<td>[0, 2$\pi$] rad</td>
<td>[0x0000, 0xFFFF]</td>
<td>[0, 65535]</td>
</tr>
</tbody>
</table>

Typically the PWM carrier frequency is from 2.5 kHz to 20 kHz [7], is much greater than the three-phase sine wave frequency. In 1 sine wave cycle, there will be $m = f_{PWM}/f_{PWM}$ PWM carrier cycle. If using the sine function available in the library to calculate, the calculation time is too large to keep up with updating data. So often, use the lookup table to calculate the sine function. Amount of updated angles during a PWM carrier cycle:

$$\Delta \phi(k) = \frac{f(k)}{f_{pwm}} \times 0xFFFF = 65535 \frac{f(k)}{f_{pwm}}$$ (1)
C. Programming Algorithm SVPWM

For the convenience of using SVPWM modules, we create an SVM subroutine. The SVM subprogram will receive the rotation angle and voltage amplitude value to perform the modulation. The program is written as follows:

```c
void SVM(int volts, unsigned int angle)
{
  unsigned int angle1, angle2;
  unsigned int half_t0,t1,t2,tpwm;
  tpwm = PTPER << 1;
  if(volts > VOLTS_LIMIT) volts = VOLTS_LIMIT;
  if(angle < VECTOR2)
  {
    angle2 = angle - VECTOR1;
    angle1 = SIXTY_DEG - angle2;
    t1 = sinetable[(unsigned char)(angle1 >> 6)]; // Look up values from table
    t2 = sinetable[(unsigned char)(angle2 >> 6)];
    t1 = ((long)t1*(long)volts) >> 15; // Scale t1 to by the volts variable.
    t1 = ((long)t1*(long)tpwm) >> 15; // Scale t1 for the duty cycle range.
    // Scale t2 time
    t2 = ((long)t2*(long)volts) >> 15;
    t2 = ((long)t2*(long)tpwm) >> 15;
    half_t0 = (tpwm - t1 - t2) >> 1; // Calculate half_t0
  }
  else if(angle < VECTOR6)
  {
    /* ...
  }
  else
  {
    /* ...
  }
  The SVM program will be executed at the Timer interrupt program, and the interrupt time is set to be equal to the end time of 1 PWM cycle.
  void __attribute__((__interrupt__, __auto_psv__)) _T1Interrupt(void)
  {
    voltageControl = M*VOLTS_LIMIT;
    SVM(voltageControl, Phi);
    Phi = Phi + deltaPhi;
  }
}
```

Every time there is an interrupt, the new angle will be updated. From this new angle value, we will compute the sine values of the angles normalized against the current sector from the built-in sine 600 tables. We will then multiply this sine value by the maximum value of the normalized PWM execution time. Finally, the PDCx registers will be updated with new values. These values are calculated based on the current SVPWM pulse patterns and the sector on which the angle is located. The generated sine table has 1024x (60/360)=171 elements, so we use a 10-bit integer pointer to access this table. Using the angle variable to look up the value of the sine table, but the angle variable is an unsigned 16-bit integer from 0x0000 → 0xFFFF, we have to use a 6-bit shift to get the top 10 bits of this angle variable to look up the sine table. Since the table has only 171 elements, to save memory, we use the unsigned char data type command for the normalized angle variables at angles 1 and 2. Finally, we scale the variable tpwm. For the best accuracy, we use long pressed multiplication (32-bit integer). Each order only takes 1 instruction cycle in dsPIC. And so, to perform multiplication takes only a few instruction cycles, completely satisfying the interrupt condition that needs to be computed quickly to load the value into the PDC.

D. Execute U/F Control Rules

Nguyên lý điều khiển tốc độ bằng cách cho động cơ không đồng bộ ba pha pha theo quan hệ U(f), được triển khai một cách đơn giản là U/f=const. Chương trình thực hiện sẽ nhận tín hiệu từ tần số đt và chuyển đổi sang giá trị góc quay và biên độ di động áp, chương trình sẽ được thực hiện từ chương trình chính tại vòng lập while (1).

freq = 5+(unsigned int){ADCValue8/1023.0*95}; // Frequency 5Hz - 100Hz
deltaPhi=(unsigned int){(65535.0/PWM_FREQ)*freq};
duty=(unsigned int){freq/2.0}; //U/F
M = duty/100.0;
```

V. EXPERIMENTAL RESULTS

Completed experimental circuit model as shown in Figure 15. In which the test motor load has parameters as shown in Figure 16.
Figures 17 and 18 are the line voltage and line voltage when modulating with frequency f = 50Hz. Figures 19 and 20 are the experimental results of measuring the modulated PWM pulse of S1, S3, and measuring dead-time. The results showed that the sine wave voltage pattern (not filtered) is consistent with the theory, the dead-time time value is 2.5µs, and the PWM frequency is 5kHz.

VI. CONCLUSIONS

The paper presented the problem of constructing and fabricating the force circuit and control circuit implementing the SVPWM spatial vector modulation algorithm and implementing the U/F speed control method on the dsPIC 30F4011 microcontroller the variable frequency in detail. In which, emphasis on quantization engineering and programming to implement SVPWM modulation algorithm. Test results on the real model show that the system works in accordance with the principles and functions of SVPWM and ensures the expansion and adjustment characteristics when necessary.

ACKNOWLEDGMENT

This study was supported by the University of Economics - Technology for Industries, Viet Nam; http://www.uneti.edu.vn/.

REFERENCES


