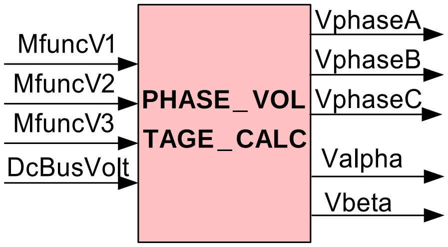
PHASE_VOLTAGE — *Phase Voltage Reconstruction*

Description

This software module calculates three phase voltages impressing to the 3-ph electric motor (i.e., induction or synchronous motor) by using the conventional voltage-source inverter. Three phase voltages can be reconstructed from the DC-bus voltage and three switching functions of the upper power switching devices in the inverter. In addition, this software module also includes the clarke transformation changing from three phase voltages into two stationary dq-axis phase voltages.



Availability

This IQ module is available in one interface format:

1) The C interface version

Module Properties

Type: Target Independent, Application Dependent

Target Devices: x281x or x280x

C Version File Names: volt_calc.c, volt_calc.h

IQmath library files for C: IQmathLib.h, IQmath.lib

Item	C version	Comments
Code Size□ (x281x/x280x)	144/144 words	
Data RAM	0 words•	
xDAIS ready	No	
XDAIS component	No	IALG layer not implemented
Multiple instances	Yes	
Reentrancy	Yes	

• Each pre-initialized "_iq" PHASEVOLTAGE structure consumes 22 words in the data memory

□ Code size mentioned here is the size of the ***calc()*** function

C Interface

Object Definition

The structure of PHASEVOLTAGE object is defined by following structure definition

```
typedef struct    { _iq  DcBusVolt;          // Input: DC-bus voltage
                    _iq  MfuncV1;            // Input: Modulation voltage phase A
                    _iq  MfuncV2;            // Input: Modulation voltage phase B
                    _iq  MfuncV3;            // Input: Modulation voltage phase C
                    Uint32  OutOfPhase;      // Parameter: Out of Phase adjustment (0 or 1)
                    _iq  VphaseA;            // Output: Phase voltage phase A
                    _iq  VphaseB;            // Output: Phase voltage phase B
                    _iq  VphaseC;            // Output: Phase voltage phase C
                    _iq  Valpha;             // Output: Stationary d-axis phase voltage
                    _iq  Vbeta;              // Output: Stationary q-axis phase voltage
                     void  (*calc)();        // Pointer to calculation function
                  } PHASEVOLTAGE;

typedef PHASEVOLTAGE * PHASEVOLTAGE_handle;
```

Item	Name	Description	Format*	Range(Hex)
Inputs	DcBusVolt	DC-bus voltage	GLOBAL_Q	80000000-7FFFFFFF
	MfuncV1	Switching function of upper switching device 1	GLOBAL_Q	80000000-7FFFFFFF
	MfuncV2	Switching function of upper switching device 2	GLOBAL_Q	80000000-7FFFFFFF
	MfuncV3	Switching function of upper switching device 3	GLOBAL_Q	80000000-7FFFFFFF
Outputs	VphaseA	Line-neutral phase voltage A	GLOBAL_Q	80000000-7FFFFFFF
	VphaseA	Line-neutral phase voltage A	GLOBAL_Q	80000000-7FFFFFFF
	VphaseA	Line-neutral phase voltage A	GLOBAL_Q	80000000-7FFFFFFF
	Valpha	Stationary d-axis phase voltage	GLOBAL_Q	80000000-7FFFFFFF
	Vbeta	Stationary q-axis phase voltage	GLOBAL_Q	80000000-7FFFFFFF

*GLOBAL_Q valued between 1 and 30 is defined in the IQmathLib.h header file.

Special Constants and Data types

PHASEVOLTAGE

The module definition is created as a data type. This makes it convenient to instance an interface to phase voltage reconstruction. To create multiple instances of the module simply declare variables of type PHASEVOLTAGE.

PHASEVOLTAGE_handle

User defined Data type of pointer to PHASEVOLTAGE module

PHASEVOLTAGE_DEFAULTS

Structure symbolic constant to initialize PHASEVOLTAGE module. This provides the initial values to the terminal variables as well as method pointers.

Methods

void phase_voltage_calc(PHASEVOLTAGE_handle);

This definition implements one method viz., the phase voltage reconstruction computation function. The input argument to this function is the module handle.

Module Usage

Instantiation
The following example instances two PHASEVOLTAGE objects
PHASEVOLTAGE volt1, volt2;

Initialization
To Instance pre-initialized objects
PHASEVOLTAGE volt1 = PHASEVOLTAGE_DEFAULTS;
PHASEVOLTAGE volt2 = PHASEVOLTAGE_DEFAULTS;

Invoking the computation function
volt1.calc(&volt1);
volt2.calc(&volt2);

Example

The following pseudo code provides the information about the module usage.

```
main()
{

}

void interrupt periodic_interrupt_isr()
{
        volt1.DcBusVolt = dc_volt1;          // Pass inputs to volt1
        volt1.MfuncV1 = M1_1;                // Pass inputs to volt1
        volt1.MfuncV2 = M2_1;                // Pass inputs to volt1
        volt1.MfuncV3 = M3_1;                // Pass inputs to volt1

        volt2.DcBusVolt = dc_volt2;          // Pass inputs to volt2
        volt2.MfuncV1 = M1_2;                // Pass inputs to volt2
        volt2.MfuncV2 = M2_2;                // Pass inputs to volt2
        volt2.MfuncV3 = M3_2;                // Pass inputs to volt2

        volt1.calc(&volt1);                   // Call compute function for volt1
        volt2.calc(&volt2);                   // Call compute function for volt2

        Vd1 = volt1.Valpha;                   // Access the outputs of volt1
        Vq1 = volt1.Vbeta;                    // Access the outputs of volt1

        Vd2 = volt2.Valpha;                   // Access the outputs of volt2
        Vq2 = volt2.Vbeta;                    // Access the outputs of volt2
}
```

Technical Background

The phase voltage of a general 3-ph motor (V_{an}, V_{bn}, and V_{cn}) can be calculated from the DC-bus voltage (V_{dc}) and three upper switching functions of inverter (S_1, S_2, and S_3). The 3-ph windings of motor are connected as the Y connection without a neutral return path (or 3-ph, 3-wire system). The overall system can be shown in Figure 1.

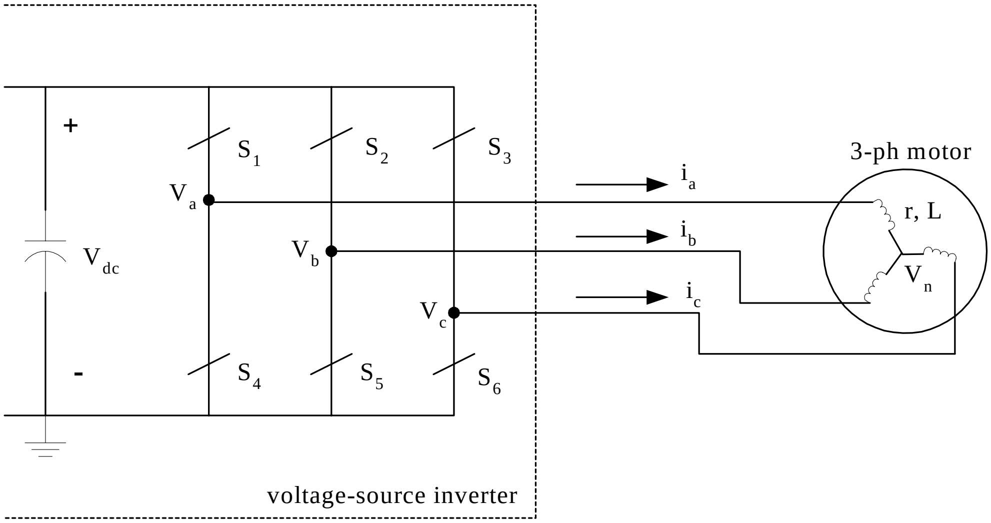


Figure 1: Voltage-source inverter with a 3-ph electric motor

Each phase of the motor is simply modeled as a series impedance of resistance and inductance (r, L) and back emf (e_a, e_b, e_c). Thus, three phase voltages can be computed as

$$V_{an} = V_a - V_n = i_a r + L\frac{di_a}{dt} + e_a \tag{1}$$

$$V_{bn} = V_b - V_n = i_b r + L\frac{di_b}{dt} + e_b \tag{2}$$

$$V_{cn} = V_c - V_n = i_c r + L\frac{di_c}{dt} + e_c \tag{3}$$

Summing these three phase voltages, yields

$$V_a + V_b + V_c - 3V_n = (i_a + i_b + i_c)r + L\frac{d(i_a + i_b + i_c)}{dt} + e_a + e_b + e_c \tag{4}$$

Without a neutral return path, according to KCL, i.e., $i_a + i_b + i_c = 0$, and the back emfs are balanced and symmetrical due to the 3-ph winding structures, i.e., $e_a + e_b + e_c = 0$, so (4) becomes

$$V_{an} + V_{bn} + V_{cn} = 0 \tag{5}$$

Furthermore, the neutral voltage can be simply derived from (4)-(5) as

$$V_n = \frac{1}{3}(V_a + V_b + V_c) \tag{6}$$

Now three phase voltages can be calculated as

$$V_{an} = V_a - \frac{1}{3}(V_a + V_b + V_c) = \frac{2}{3}V_a - \frac{1}{3}V_b - \frac{1}{3}V_c \tag{7}$$

$$V_{bn} = V_b - \frac{1}{3}(V_a + V_b + V_c) = \frac{2}{3}V_b - \frac{1}{3}V_a - \frac{1}{3}V_c \tag{8}$$

$$V_{cn} = V_c - \frac{1}{3}(V_a + V_b + V_c) = \frac{2}{3}V_c - \frac{1}{3}V_a - \frac{1}{3}V_b \tag{9}$$

Three voltages V_a, V_b, V_c are related to the DC-bus voltage (V_{dc}) and three upper switching functions (S_1, S_2, S_3) as the following relation.

$$V_a = S_1 V_{dc} \tag{10}$$

$$V_b = S_2 V_{dc} \tag{11}$$

$$V_c = S_3 V_{dc} \tag{12}$$

where S_1, S_2, S_3 = either 0 or 1, and S_4 = 1-S_1, S_5 = 1-S_2, and S_6 = 1-S_3. (13)

As a result, three phase voltages in (7)-(9) can also be expressed in terms of DC-bus voltage and three upper switching functions as follows:

$$V_{an} = V_{dc}\left(\frac{2}{3}S_1 - \frac{1}{3}S_2 - \frac{1}{3}S_3\right) \tag{14}$$

$$V_{bn} = V_{dc}\left(\frac{2}{3}S_2 - \frac{1}{3}S_1 - \frac{1}{3}S_3\right) \tag{15}$$

$$V_{cn} = V_{dc}\left(\frac{2}{3}S_3 - \frac{1}{3}S_1 - \frac{1}{3}S_2\right) \tag{16}$$

It is emphasized that the S_1, S_2, and S_3 are defined as the upper switching functions. If the lower switching functions are available instead, then the out-of-phase correction of switching functions is required in order to get the upper switching functions as easily computed from equation (13).

Next the clarke transformation changing from three phase voltages (V_{an}, V_{bn}, and V_{cn}) to the stationary dq-axis phase voltages (V_{ds}^{s} , and V_{qs}^{s}) are applied by using the following relationship. Because of the balanced system (5), V_{cn} is not used in clarke transformation.

$$V_{ds}^{s} = V_{an} \tag{17}$$

$$V_{qs}^{s} = \frac{1}{\sqrt{3}}(V_{an} + 2V_{bn}) \tag{18}$$

Figure 2 depicts the abc-axis and stationary dq-axis components for the stator voltages of motor. Notice that the notation of the stationary dq-axis is sometimes used as the stationary $\alpha\beta$-axis, accordingly.

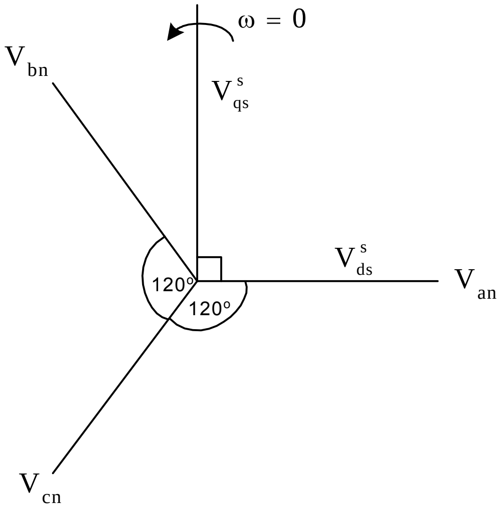


Figure 2: The abc-axis and stationary dq-axis components of the stator phase voltages

Next, Table 1 shows the correspondence of notations between variables used here and variables used in the program (i.e., volt_calc.c, volt_calc.h). The software module requires that both input and output variables are in per unit values.

	Equation Variables	**Program Variables**
Inputs	S_1	MfuncV1
	S_2	MfuncV2
	S_3	MfuncV3
	V_{dc}	DcBusVolt
Outputs	V_{an}	VphaseA
	V_{bn}	VphaseB
	V_{cn}	VphaseC
	V_{ds}^{s}	Valpha
	V_{qs}^{s}	Vbeta

Table 1: Correspondence of notations