

This blog shows a derivation of a time domain model of a BLDC PMSM motor. The block diagram is shown in Figure 1.

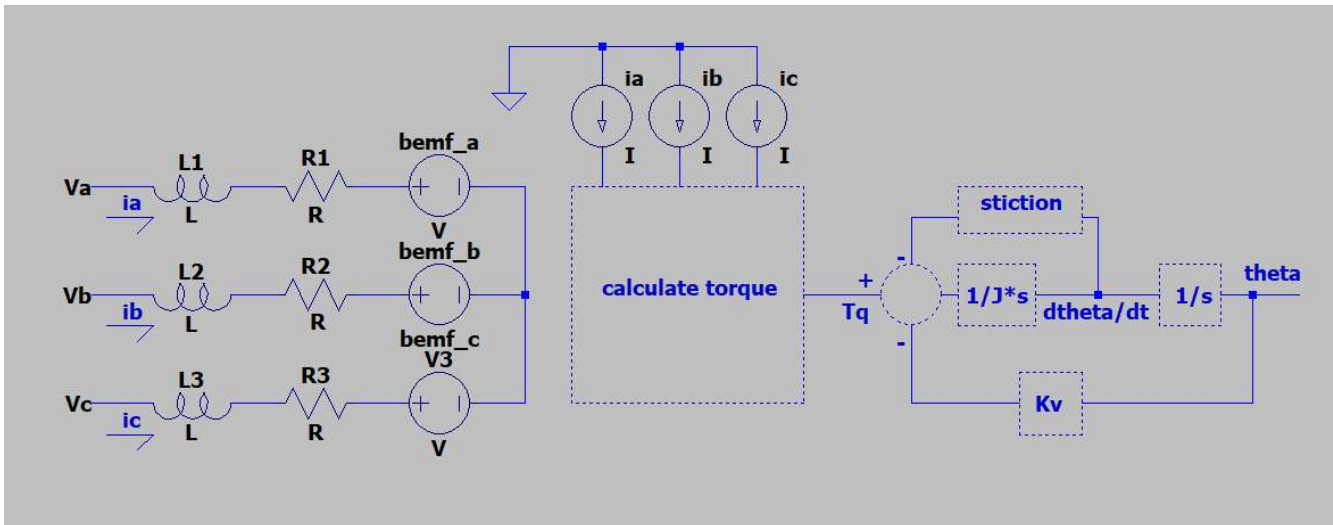


Figure 1 – BLDC PMSM model

Derivation

The model assumes a 3-phase N_p pole motor is driven with a sinusoidal drive which can an averaged model of Space Vector Modulation (SVM) to reduce simulation time, or a switched SVM model.

The equations can be divided into two sets, the first is the electrical input and the second is the mechanical generation of torque. The electrical equations are given here by using Kirchoff's Law and summing voltages around a closed loop and net current into a node. In these equations all currents and voltages are functions of time, they do not yet assume what the input voltage waveform is.

$$\begin{aligned}
 V_a - L \cdot \frac{di_a}{dt} - i_a \cdot R - bemf_a + bemf_b + i_b \cdot R + L \frac{di_b}{dt} &= V_b \\
 V_c - L \cdot \frac{di_c}{dt} - i_c \cdot R - bemf_c + bemf_a + i_a \cdot R + L \frac{di_a}{dt} &= V_a \quad [1] \\
 i_a + i_b + i_c &= 0
 \end{aligned}$$

The inputs for the electrical portion are the bus voltages, $V_a(t)$, $V_b(t)$ and $V_c(t)$. The outputs are the phase currents $i_a(t)$, $i_b(t)$, and $i_c(t)$. The bemf voltages are coupled from the mechanical model.

The mechanical model is given by equations [2],

$$\dot{\omega}(t) = \frac{torque(t)}{J} - \frac{Kv}{J} \omega(t) \quad [2]$$

Equations [1] and [2] are rewritten below in a form to explicitly show the state variables we will solve. The states of the system are $i_a(t)$, $i_b(t)$, $i_c(t)$, $\theta(t)$ and $\omega(t)$.

$$\begin{aligned}\frac{di_a}{dt} &= \frac{1}{3L} \cdot (2V_a - V_b - V_c - 3i_aR - 2 \cdot bemf_a + bemf_b + bemf_c) \\ \frac{di_b}{dt} &= \frac{1}{3L} \cdot (-V_a + 2V_b - V_c - 3i_bR + bemf_a - 2bemf_b + bemf_c) \\ \dot{\omega} &= \frac{Torque(t)}{J} - \frac{Kv}{J}\omega \\ \dot{\theta} &= \omega \\ 0 &= i_a + i_b + i_c\end{aligned}\tag{3}$$

We will talk about the torque and bemf terms next. Both of these are also dependent on the rotor flux produced by the magnets.

Torque and Bemf

When the motor winding's are not connected and the motor is spun externally, the rotor magnets will induce a sinusoidal voltage in the three stator winding's (armature) which we denoted above as $bemf_a$, $bemf_b$ and $bemf_c$. These three voltage are displaced from one another by 120 electrical degrees. The rotor flux is a vector Ψ_{rotor} rotating at the rotor speed.

When the stator coils A, B & C are energized there will be a net flux Ψ_{arm} generated by the coils that attracts the opposite polarity rotor flux. When the coils are driven with sinusoidal voltages displaced 120 degrees apart then Ψ_{arm} will be a vector that rotates at the frequency determined by the controller. This will move the motor until the rotor flux and armature flux are aligned and the torque is at a null. The torque that spins the motor is maximum when the armature flux is pointing 90 electrical degrees from the rotor flux. In general the bemf is given by Faraday's law as

$$\varepsilon = -n \frac{d\phi}{dt}\tag{4}$$

When the angle between the rotor and stator is λ , the shaft torque generated by the motor will be

$$Torque_q = (\Psi_{rotor})(\Psi_{stator})\sin(\lambda)\tag{5}$$

The q subscript on Torque is to denote torque along the quadrature axis. This is the shaft torque that generates work.

When λ is 90 electrical deg the generated torque will be maximized. The bemf generated by a sinusoidal rotor flux is given by the following,

$$\begin{aligned}
 bemf_a(t) &= K_b \cdot \omega \cdot \sin(\omega t - \lambda) \\
 bemf_b(t) &= K_b \cdot \omega \cdot \sin(\omega t - \frac{2}{3}\pi - \lambda) \\
 bemf_c(t) &= K_b \cdot \omega \cdot \sin(\omega t - \frac{4}{3}\pi - \lambda)
 \end{aligned}
 \tag{6}$$

The conservation of power across the air gap holds so the product of winding current and bemf voltage equals the product of the rotor speed and torque generated by that winding. As a consequence the winding currents will also be sinusoidal.

$$\begin{aligned}
 i_a(t) &\propto \sin(\omega t - \lambda) \\
 i_b(t) &\propto \sin(\omega t - \frac{2}{3}\pi - \lambda) \\
 i_c(t) &\propto \sin(\omega t - \frac{4}{3}\pi - \lambda)
 \end{aligned}
 \tag{7}$$

SVM (Space Vector Modulation)

Before introducing FOC control I will discuss the type of inverter used to drive the motor. The inverter takes in a DC link voltage and using control signals, generates a 3-phase output to the BLDC motor as shown in Figure 2a.

Figure 2a – Six Switch Inverter

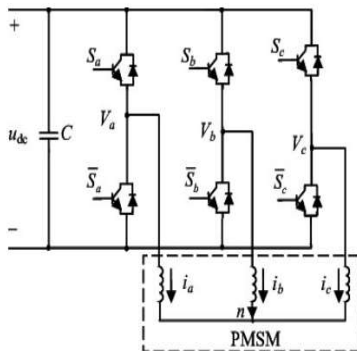
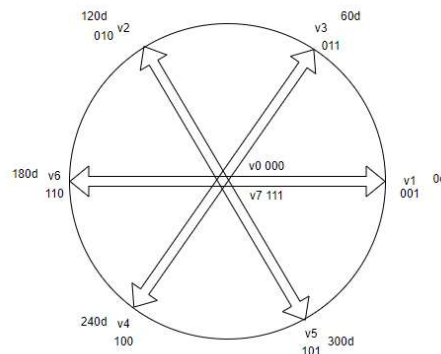


Figure 2b – Space Vector Diagram



The six switches have $2^3 = 8$ possible states. The states can be drawn as vectors in space, hence the name Space Vector (SV). The vector angles are drawn with respect to winding A and represent the winding angle on the rotor over 360 electrical degrees. The 8 possible states are shown in figure 2b. The vector magnitude with all upper switches ON or all OFF is zero and shown at the origin. The remaining six states represent the *basis* vectors that make up SVM.

A table of the states is shown in Table 1. The state is three binary digits representing the three switch pairs. A bit value of 1 represents the switch is ON and connected to the DC link voltage. A value of 0 represents the switch is OFF and connected to ground (0v). The state 000 has outputs A,B and C connected to 0v while a state 111 has the outputs all connected to the link voltage. States 001 and 110

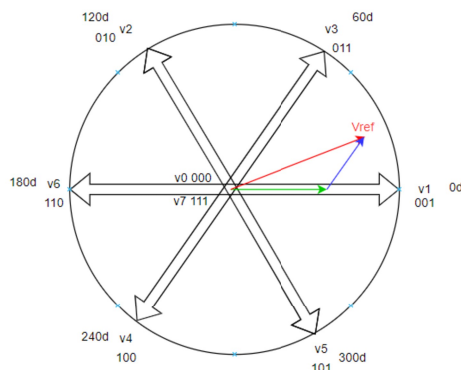
have currents of equal but opposite polarity in the coils. In the SVM Vector space these are vectors v1 and v2 which point in opposite directions.

Table 1 – SVM basis vectors

state	vector
000	v0
001	v1
010	v2
011	v3
100	v4
101	v5
110	v6
111	v7

For SVM an arbitrary voltage vector is applied that is a combination of two adjacent *basis* vectors. This is shown in Figure 3. The **red** vector is what we want at a given instant in time. This can be achieved by a portion of the v1 vector (**green**) and a portion of the v3 vector (**blue**).

Figure 3 – Arbitrary Vref Vector



The arbitrary vector Vref can be generated by pulse width modulating (PWM) between basis vectors v1 and v3. To adjust α and β the time periods of each need to have the correct ratio but in addition the total time must add up to the PWM period. For this reason a portion of state v0 or v7 is added to meet the PWM time criteria.

$$V_{ref} = \alpha \vec{v}_1 + \beta \vec{v}_3 + \gamma \vec{v}_7/0 \tag{8}$$

Another way of writing this with the PWM times is follows:

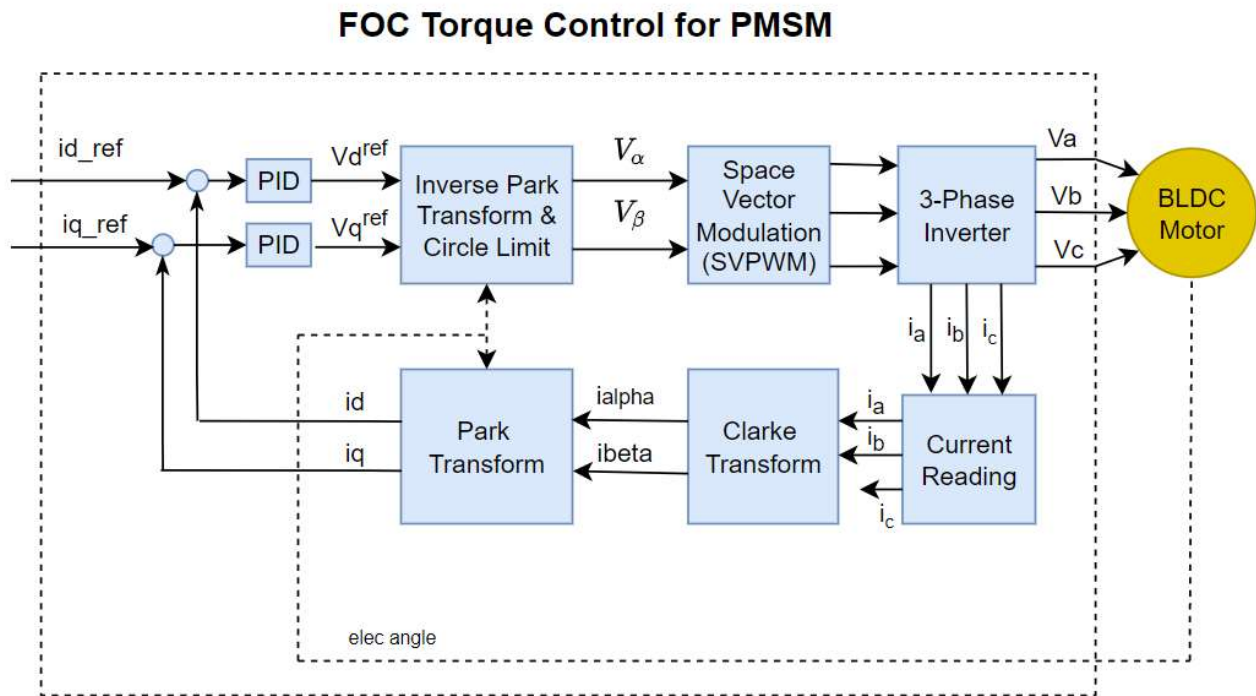
$$V_{ref} = \frac{T_1}{T_{pwm}} \vec{v}_1 + \frac{T_2}{T_{pwm}} \vec{v}_3 + \frac{T_3}{T_{pwm}} \vec{v}_{7/0} \quad [9]$$

$$T_3 = T_{pwm} - T_1 - T_2$$

Field Oriented Control

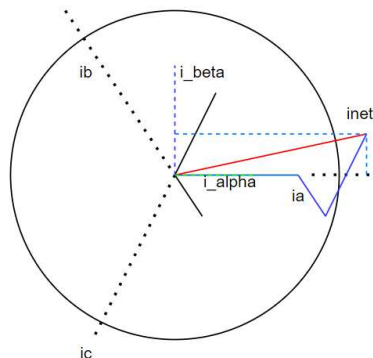
The block diagram for FOC control of the BLDC PMSM motor is shown below:

Figure 4 – Block Diagram – FOC Control



The motor current can be shown as three vectors spaced 120 electrical degrees apart but in a two-dimensional space three vectors are redundant. Knowing any two of the three vectors we use the current constraint to calculate the third vector. The following diagram shows the i_a and i_b vectors resolved into two orthogonal vectors i_α and i_β .

Figure 5 – Resolving Current Vectors



The individual winding current (i_a, i_b, i_c) are added as vectors to result in the net current vector i_{net} (shown red). This resulting vector is resolved into the two orthogonal components along i_α and i_β . This allows us to electrically measure two currents and then in firmware compute two orthogonal current values. This is done using the **Clarke** Transformation. The assumption here is the three winding currents add to zero.

$$\begin{aligned}i_\alpha &= i_a \\i_\beta &= \frac{1}{\sqrt{3}}i_a + \frac{2}{\sqrt{3}}i_b \\i_c &= -i_a - i_b\end{aligned}\quad [10]$$

The two orthogonal vectors are referenced to the stator. What we want to do next is to change the reference frame to the rotor. To do this we need prior knowledge of the rotor and θ . Then we can use the **Park** transform to generate two orthogonal vectors i_q and i_d as follows:

$$\begin{aligned}i_d &= i_\alpha \cos \theta + i_\beta \sin \theta \\i_q &= -i_\alpha \sin \theta + i_\beta \cos \theta\end{aligned}\quad [11]$$

In [11] i_q is the quadrature torque component we talked about earlier which generates the shaft torque. In many applications we will want to control the direct torque, i_d , to near zero. The angle θ can either be measured directly with a shaft encoder or can be computed in firmware using an estimator and observer model.

The values of i_d and i_q can now be used to control the two torque components in the motor. Two PID loops are shown in the block diagram which take the errors ($i_{d_ref} - i_d$) and ($i_{q_ref} - i_q$). These error signals drive the two PID blocks and generate the two respective outputs V_{d_ref} and V_{q_ref} . These two values are still in the rotor reference plane and are needed to compute the inputs to the SVM block, V_α and V_β . These two latter values are in the stator reference plane.

Our remaining task is to use all the above material as the basis for a simulation model to demonstrate a working FOC Motor control on a BLDC PMSM motor.