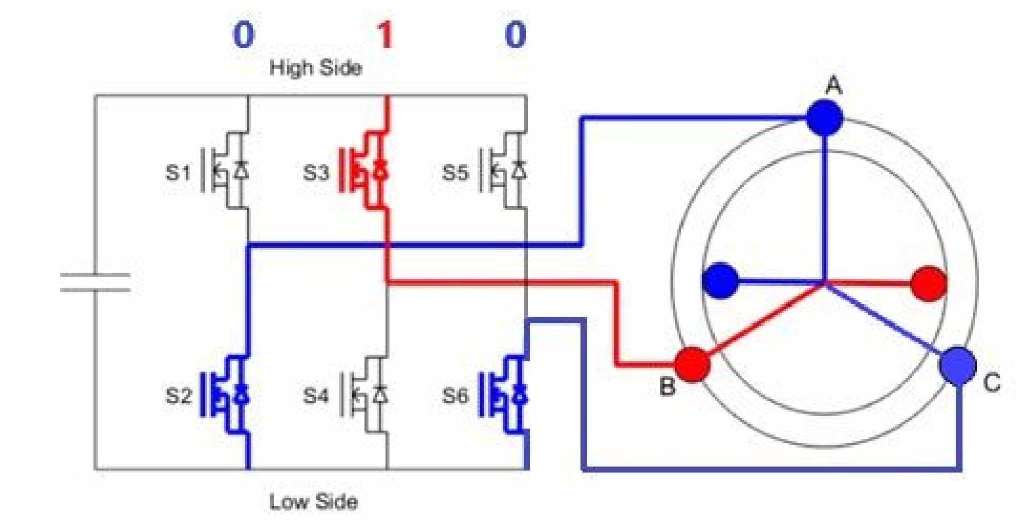
Space Vector Modulation (SVM) for Motor Control

What Is Space Vector Modulation?

Space vector modulation (SVM) is a common technique in field-oriented control for induction motors and permanent magnet synchronous motors (PMSM). Space vector modulation is responsible for generating pulse width modulated signals to control the switches of an inverter, which then produces the required modulated voltage to drive the motor at the desired speed or torque. Space vector modulation is also known as space vector pulse width modulation (SVPWM). You can use MATLAB® and Simulink® to implement space vector modulation techniques or leverage prebuilt SVM libraries for motor control applications.

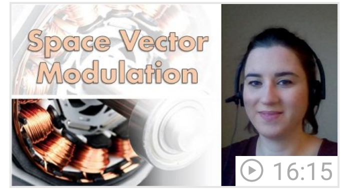
SVM Objective

Consider the concept of space vector modulation for motor control on a three-phase inverter with six switches represented by the following equivalent circuit. Note, there are eight valid switching configurations.



Three-phase inverter circuit connected to the stator windings of a motor.
* The states of switches S2, S4, and S6 are complimentary to S1, S3, and S5 respectively.

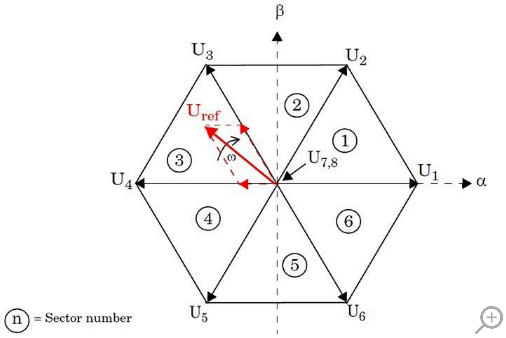
Each switching configuration results in a specific voltage applied to the motor terminals. The voltages are basic space vectors and represent their magnitude and direction in a space vector hexagon.



Space Vector Modulation | Motor Control, Part 5

BLDC Speed Control Using PWM | Motor Control, Part 3

How to Design and Simulate Motor Control Algorithms | Field-Oriented Control of PMSM with Simulink, Part 2



Space Vector	S1	S3	S5
U1	1	0	0
U2	1	1	0
U3	0	1	0
U4	0	1	1
U5	0	0	1
U6	1	0	1
U7	0	0	0
U8	1	1	1

Space vector hexagon with basic vectors U1-U8.

Three-phase inverter circuit connected to the stator windings of a motor.

The switching states that correspond to the basic space vectors (for direction) and the null vectors (for magnitude) are combined to approximate a voltage vector of any magnitude, at any position, within the space vector hexagon. For example, for every pulse width modulation (PWM) period, the reference vector 'Uref' is averaged by using a switching sequence of two adjacent space vectors (U3 and U4 in the figure) for a specified duration of time and a null vector (U7 or U8) for the rest of the period.

By controlling the switching sequence, and consequently the ON time duration of pulses, any voltage vector with varying magnitude and direction is achievable for every PWM period. The objective of space vector modulation technique is to generate switching sequences that correspond to the reference voltage vector for every PWM period to achieve a continuously rotating space vector.

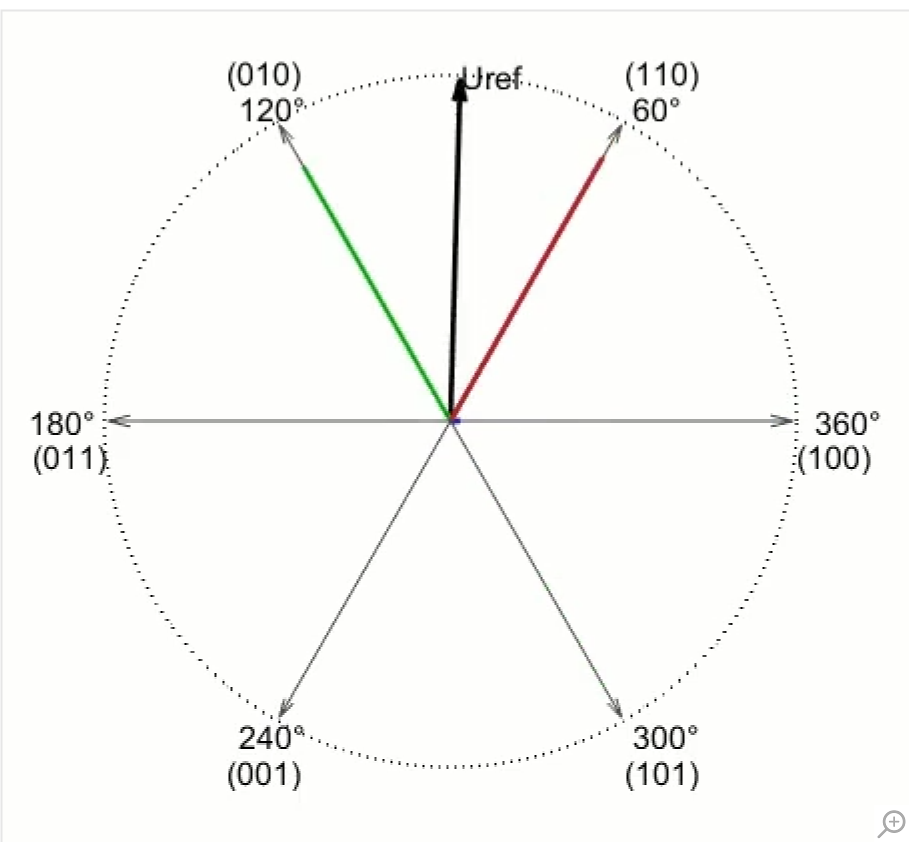


Illustration of a rotating reference space vector.

SVM Operation

The space vector modulation technique operates on the reference voltage vector to generate appropriate gate signals for the inverter every PWM period, with the objective of achieving a continuously rotating space vector.

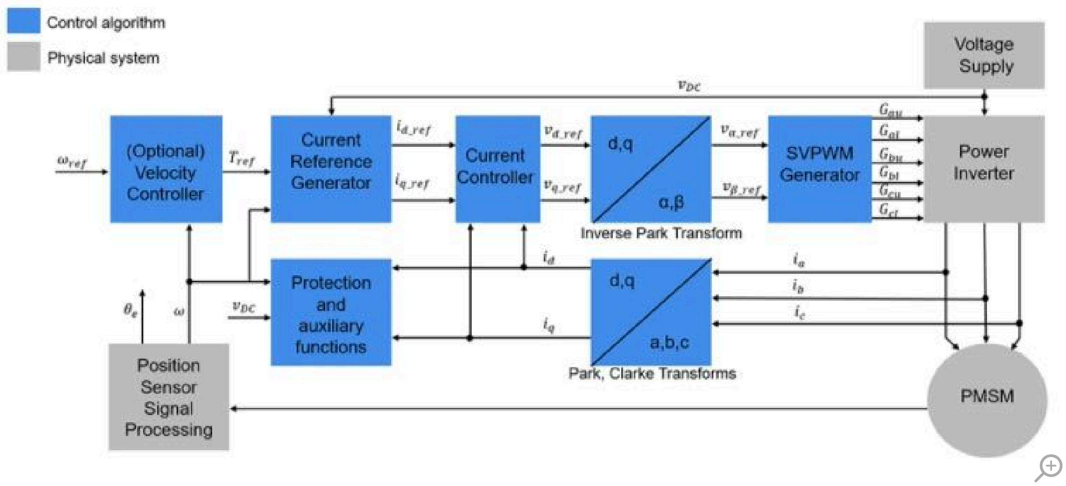
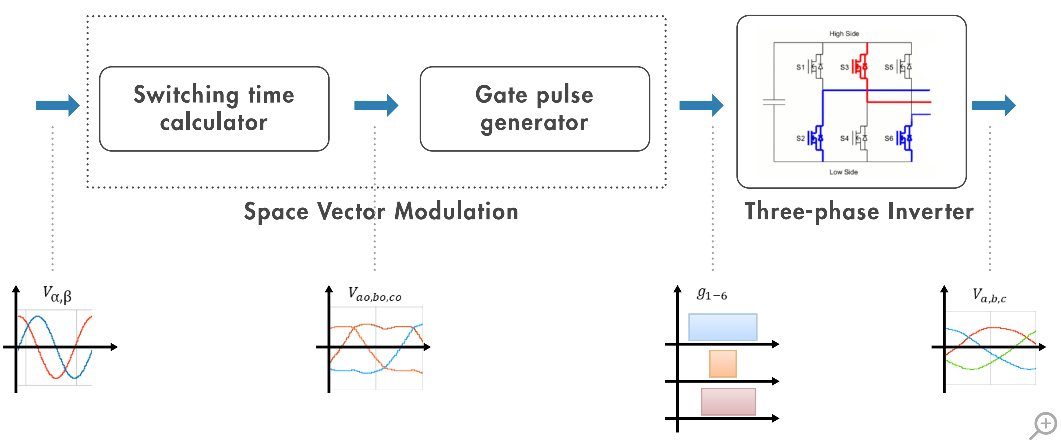


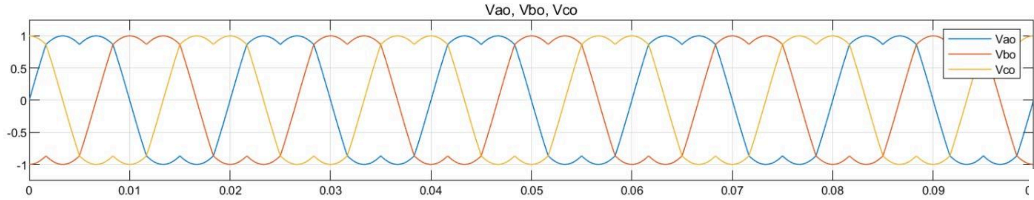
Illustration of field-oriented control architecture with space vector modulation.



Block diagram illustrates one example of a space vector modulation workflow.

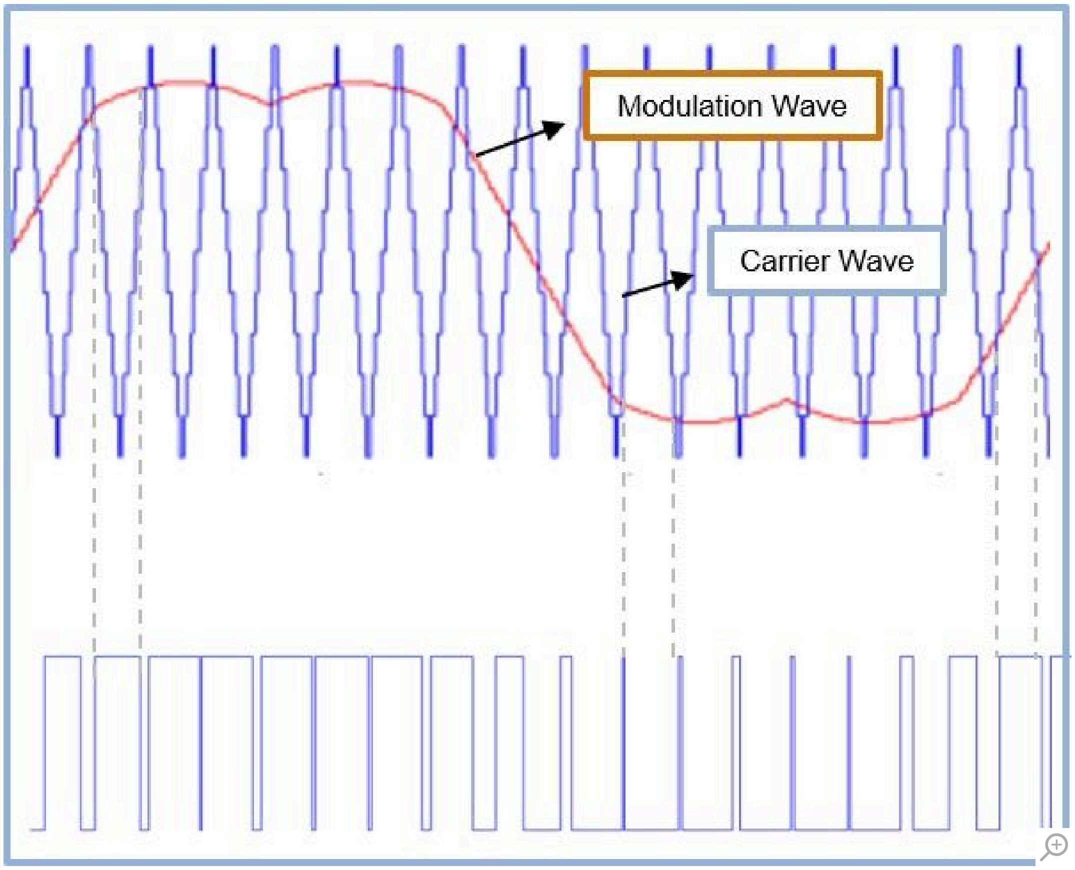
For every PWM period, with voltage vector as input reference, the SVM algorithm:

- Calculates on- and off-gating times based on the reference voltage vector
- Uses gating times to generate the double hump modulation waveforms
- Uses gating times to generate appropriate gate pulses for the inverter switches



Space vector modulated voltage signals generated by SVM algorithm.

The nature of the generated modulation wave with a double hump maximizes the utilization of the available DC bus voltage. This provides a better rated voltage output when compared with Sinusoidal Pulse Width Modulation (SPWM) technique.



Gate pulse generation as a result of comparing the modulation wave and the carrier wave.

You can then apply the generated gate signals to the switches of the three-phase inverter to drive the motor at the desired speed or torque.

PWM Hardware Support

Hardware boards, such as Arduino, Raspberry Pi, and TI boards, generate gate pulses to drive the power inverter by receiving the modulation waveforms.

To learn more about implementing field-oriented control with SVM on TI hardware, watch this video: How to Deploy Control Algorithm to a Microcontroller | Field-Oriented Control of PMSM with Simulink | Part 3 (4:52).

Motor control algorithms with PWM techniques employed typically require execution at higher frequencies in the order of few kHz depending on the design requirements. It is important to evaluate the correctness of the control architecture early, before committing to the expense of hardware testing. One such approach is to use a simulation environment. For example, with Simulink, you can simulate and verify the control architecture, including pulse width modulation techniques such as space vector modulation, against a modeled motor and rectify errors at an early stage.

To use SVM in Simulink, refer to Space Vector Generator block.

To learn more about how to design and implement motor control algorithms see, Motor Control Blockset and Simscape Electrical.

Examples and How To

- Induction Machine Direct Torque Control with Space Vector Modulator - Example
- Asynchronous Machine Direct Torque Control with Space Vector Modulator - Example
- Space Vector PWM-DTC Induction 200 HP Motor Drive - Example
- Three-Phase PMSM Drive - Example
- Field-Oriented Control of Permanent Magnet Synchronous Machine - Example
- ATB Technologies Cuts Electric Motor Controller Development Time by 50% Using Code Generation for TI’s C2000 MCU - Customer Story
- Eaton Corporation Speeds Development of a Medium-Duty Hybrid Truck - Customer Story

Software Reference

- SVPWM Generator (Two-Level) - Documentation
- PWM Generator (Three-Phase, Two-Level) - Documentation
- PWM Generator (Three-Phase, Three-Level) - Documentation
- Space Vector PWM VSI Induction Motor Drive - Documentation

See also: Simscape Electrical, Embedded Coder, Clarke and Park Transforms, Field-Oriented Control, Motor Control Design with Simulink, Powering Electrification with MATLAB, Simulink, and Simscape, Motor Simulation for Motor Control Design, Induction Motor Speed Control, Field-Weakening Control