Example n°7

FIELD ORIENTED CONTROL OF A PERMANENT MAGNET SYNCHRONOUS MOTOR (PMSM)

Written by: Imperix Ltd, Rue de l'Industrie 17, 1950 Sion, Switzerland

Requirements:
- Matlab/Simulink ≥ 2015a
- Simulink Coder, Matlab Coder, Embedded Coder
- PLECS viewer ≥ 4.0.1 – free download here.
- BoomBox ACG blockset ≥ 1.1 – request trial and license here.

1 INTRODUCTION

This example shows the essential elements of the control and operation of a three-phase permanent magnet synchronous motor (PMSM, Figure 1). The control is meant to be implemented using one BoomBox unit, through the automated code generation process (ACG).

Apart from the classical V/f control approach, field oriented control is certainly one of the most applied control technique for synchronous motors. The related Simulink files contains not only the control implementation, but also a model of the plant. Hence, these files can be simultaneously used for simulation purposes as well as for automated code generation for the BoomBox control platform.

The parameters defined in the model correspond to a setup made of imperix's PowerTrench 800V power modules as well as ModuLink 800V external voltage sensors. As such, this example is also meant to serve as a ready-made template for any control developments intended for this hardware.

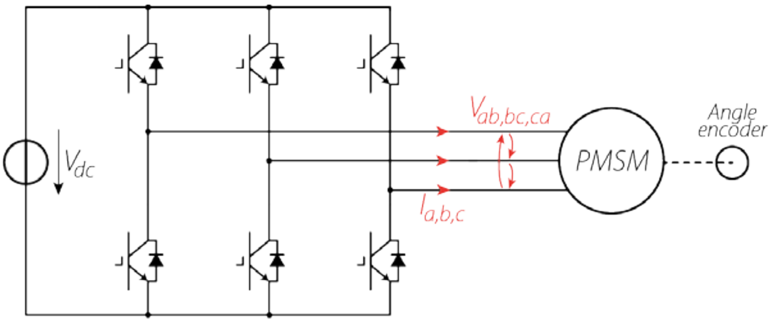


Figure 1: Electrical scheme of the considered system

2 GETTING STARTED WITH THE FILES

Apart from the minimum software requirements (Simulink/PLECS Viewer/ACG blockset), the complete simulation and control files are contained in file PMSM_FOC.slx. As usual with imperix's Simulink files, the control part is clearly separated from the plant model, which is only used in simulation.

All Simulink sheets are commented so that their operation can be easily understood.

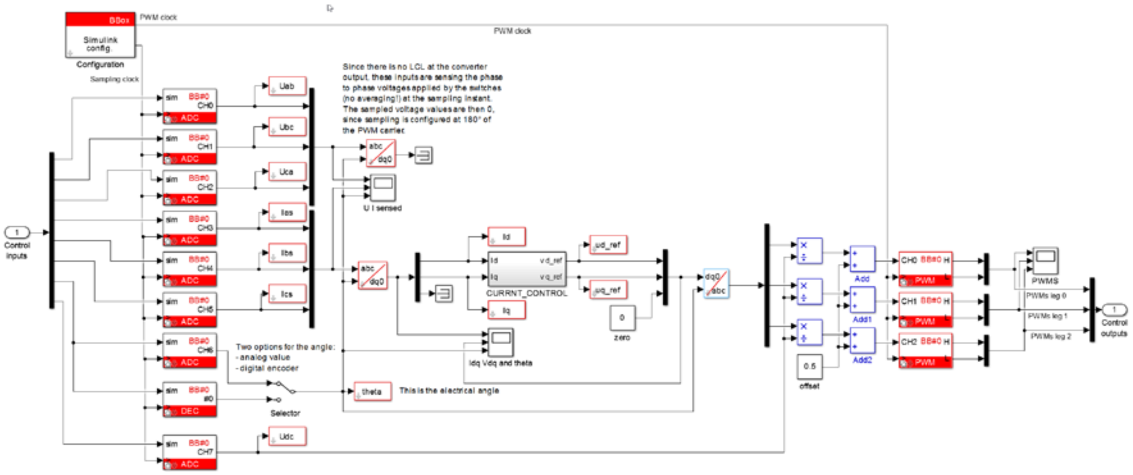
2.a PRINCIPLES OF OPERATION

The choice of the execution mode can be made in the Simulink config. block, located in the control section (Figure 2). Depending on the execution mode (Simulation/Code Generation), the appropriate variant for each subsystem is used, leading to the corresponding behavior:

- **Simulation**, Simulink runs the plant model as well as the variants corresponding to each input/output peripheral (modulators, analog inputs, etc.). This leads to an accurate simulation of the overall system, taking into account the exact sampling instants and PWM patterns.
- **Code Generation,** Simulink disregards the plant model and executes the appropriate C/C++ driver functions of the BoomBox(es). Real-time control code is automatically generated, compiled and downloaded into the master BoomBox.

This configuration block also controls some key parameters, such as:

- **Simulation step size**, corresponds to the time step between each iteration of the simulation solver. As such, only fixed-step solvers can be used. This step size is usually much smaller than the switching period, such that PWM-related phenomena can be highlighted by the simulation (PWM patterns, switching ripple, etc.).
- **Switching frequency**, corresponds to the carrier frequency utilized for all PWM modulators. In most cases, this also corresponds to the frequency at which the control algorithms are run (discrete control frequency).
- **Relative phase**, corresponds to the phase delay actually applied between the control routine and the sampling instant (that is common on all analog inputs – i.e. simultaneous sampling). With a triangular PWM modulation, configuring a relative sampling phase of 0.5 (180°) guarantees that midpoints (average value) of all signals are sampled.



Since there is no LCL at the converter output, these inputs are sensing the phase to phase voltages applied by the switches (no averaging!) at the sampling instant. The sampled voltage values are then 0, since sampling is configured at 180° of the PWM carrier.

Two options for the angle:
- analog value
- digital encoder

This is the electrical angle

Figure 2 : Top-level scheme of the implemented control strategy.

2.b PLANT MODEL

The plant model is implemented using PLECS. It can be run using the freely-available PLECS Viewer license (available on PLEXIM website). Each phase leg of the inverter is implemented using a half-bridge topology, modeled with ideal switches and anti-parallel diodes. The PMSM is loaded by an external constant torque and an angle sensor provides the shaft angle measurement.

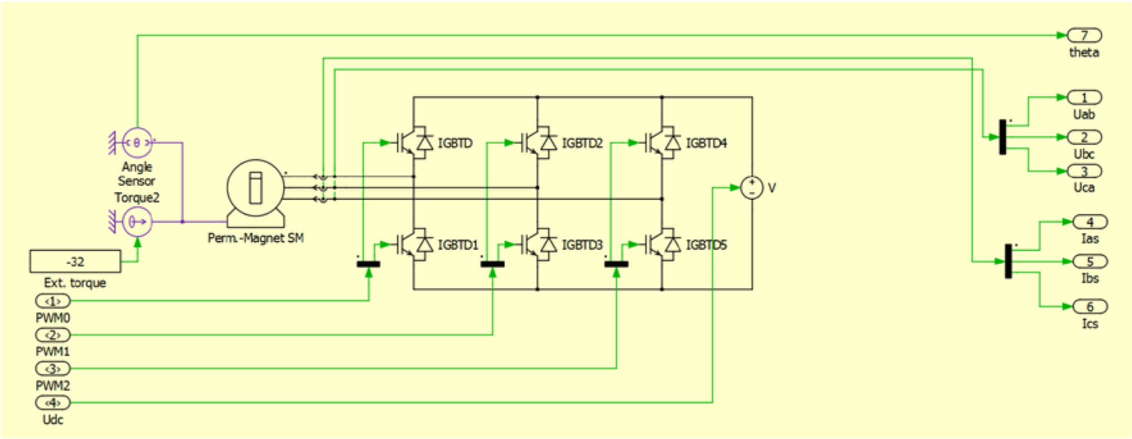


Figure 3 : Top-level schematic of the PLECS-based plant model

3 HARDWARE AND SOFTWARE CONFIGURATION

The configuration of all analog inputs of the BoomBox is contained on the top-level sheet of the control implementation (Figure 2). Angle sensing is implemented twice on this model, in order to depict the options available with the BoomBox.

On the one hand, analog inputs can be used to directly acquire the angle value. For example, such variant fully supports absolute encoders (e.g. Megatron MAB40A) as well as wide-spread sin/cos angle sensors.

On the other hand, a digital sensor (incremental encoder) can also be used. With its outputs wired on the digital inputs connector of the BoomBox, the angle information (in radian) can be retrieved by the use of special function block (DEC) in Simulink.

4 IMPLEMENTATION OF THE CONTROL APPLICATION

The DQ-current control can be found in a sub-block. It is a simple version of the DQ-current control, without the full axes decoupling. Both PI controllers are fed with a reset signal. In the code-generation mode, doing this allows to reset (and thus disable) the integration while the BoomBox controller is either not operating or in fault mode. As a consequence, this avoids all startup transients that are to a saturated integrator.

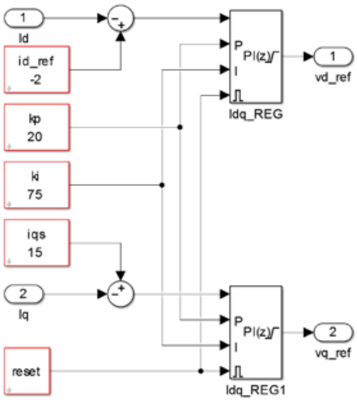


Figure 4 : Content of each phase-related subsystem

5 ILLUSTRATIVE SIMULATION RESULTS

Figure 5 shows some simulation results from the presented example. Full power reversal is applied at simulation time = 0.25s.

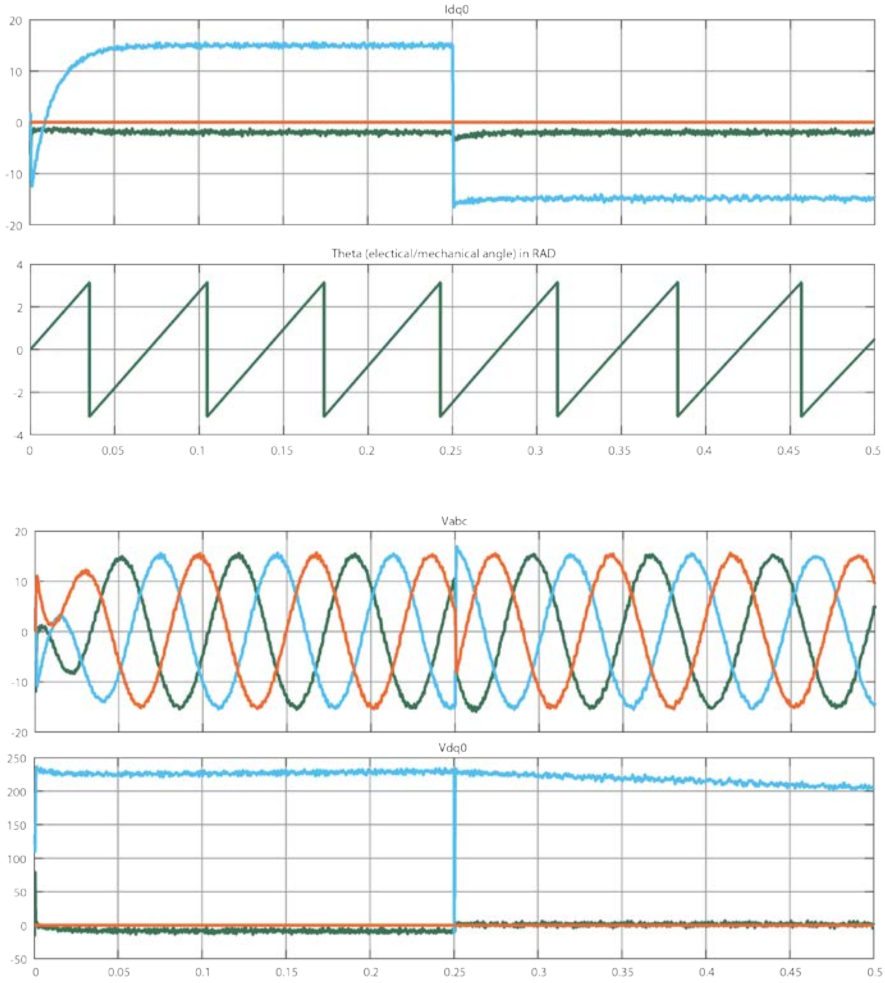


Figure 5 : Excerpt of the simulation results, showing a complete power reversal at t=0.25s.