Implementation of the imperix SS-PWM for MMC

TN160 | Posted on October 11, 2021 | Updated on May 7, 2025

Jessy ANÇAY
Sales & Project Engineer
imperix • in

Table of Contents

This technical note offers insights into the SS-PWM peripheral block, which is part of the standard firmware of imperix controllers. The SS-PWM block provides multilevel Pulse Width Modulation (PWM) as well as an integrated submodule balancing mechanism derived from the Sort-&-Select approach initially proposed in [1].

Background and motivations

The generation of modulated multilevel waveforms is a long-established topic for multilevel converters. Historically, cascaded converters mostly used modulation techniques that rely on multiple carriers. The latter are then carefully arranged (e.g. phase-shifted) so that the resulting waveform features multiple voltage steps as well as an increased apparent switching frequency [1].

In this first case, the balancing of the submodule voltages is a separate concern, entirely distinct from the modulation, which is addressed using dedicated control loops (e.g. local voltage control loops). This approach has been used since the late ’90 for cascaded converters with separate DC sources [2]. It was first applied to Modular Multilevel Converters as well in 2008 [3].

Independently from these conventional approaches, Modular Multilevel Converters were immediately proposed with a special modulation and balancing approach, which dynamically attributes the switching actions to suitable submodules, as a function of their relative capacitor voltage and power flow direction [4].

The elegance of the Sort-&-Select PWM technique lies in its simplicity and hence robustness. For balancing, only the relative ranking and the arm current polarity are required. The strategy can be hence easily applied to various converter topologies and modulation techniques (conventional PWM, nearest level modulation, selective harmonic elimination, etc.).

Sort-&-select balancing and PWM for MMC

One of the challenges associated with MMCs is to balance the capacitor voltages among all submodules. Adequate balancing ensures that the total blocking voltage is equally spread among submodules so that individual ratings are not exceeded. To some extent, submodule voltages balancing also contributes to the quality of the output voltage waveform. The overall balancing problem can generally be split into three distinct sub-problems:

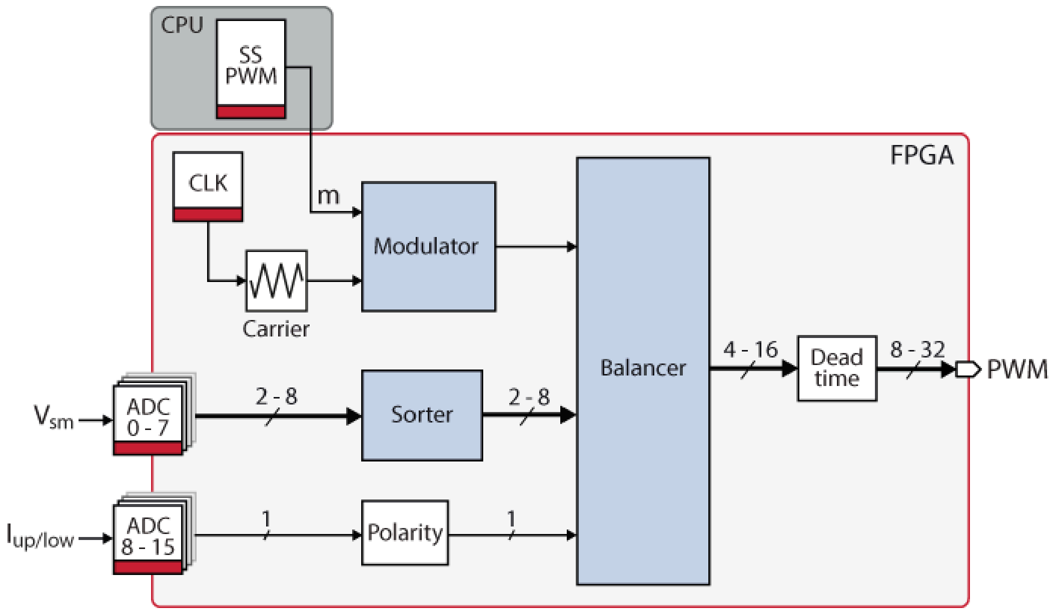
- Submodule-level balancing, namely between submodules of the same arm.
- Converter-level balancing, namely balancing between the different arms.
- Overall energy control.

The Sort-&-Select modulation deals exclusively with submodule-level balancing. As such, the converter-level balancing and overall energy control are left as further control problems. The latter fall outside the scope of this note and are addressed in dedicated articles.

Algorithm of the imperix SS-PWM for MMC

Operating principles

This section details the operation of the SS-PWM hardware as implemented within imperix controllers. The diagram below introduces the main blocks and gives an overview of the FPGA implementation. Documentation regarding how to use the SS-PWM blocks is given in SS-PWM – Multilevel PWM with Sort-&-Select balancing.



Structure and main data paths of the imperix SS-PWM peripheral.

The FPGA-based SS-PWM for MMC essentially contains three subsystems:

- The **Modulator** is responsible for generating/synthesizing the multilevel waveform. Its output is the exact desired waveform, defined with a very small sampling period (here 4ns).
- The **Sorter** is aimed at ranking the submodules as a function of their charge level (capacitor voltage). Its output is a list of indices.
- The **Balancer** is tasked to identify the level transitions within the desired multilevel waveform and to dynamically assign the corresponding switching events to suitable submodules.

Modulator

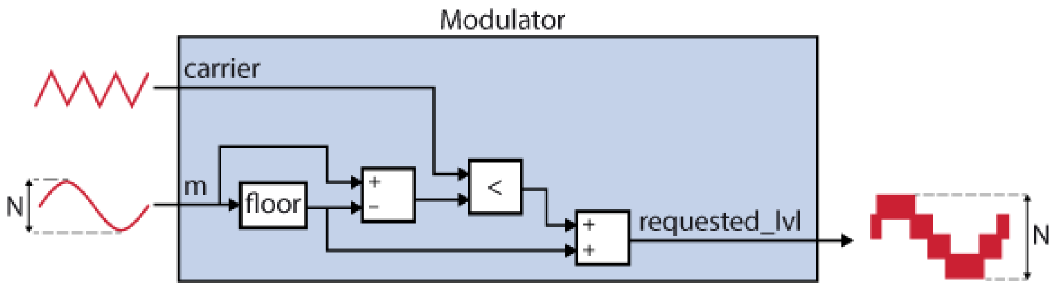
The input of the modulator is the modulation index `m`. Its range is [0;N] for operation half-bridge submodules, respectively [-N;N] for operation with full-bridge submodules. N is the number of usable submodules par arm.

Based on the real-valued index `m`, the modulator generates the discrete value `requested_lvl`, which is modulated with the configured `carrier` signal. This defines, at every instant, how many modules are inserted (and apply their voltage) to the arm.

As shown in the diagram below, the output waveform is in fact the sum of two parts:

- The lower integer part of `m` represent modules that are inserted all along a given switching period.
- The fractional part of `m` is equivalent to a duty-cycle that is applied to an additional module, only inserted during part of the switching period.

This way, the average number of modules inserted during a given switching period is equal to the modulation index `m`. Interestingly, this approach is also compatible with staircase modulation, provided that `m` is an integer staircase waveform by itself.



Implementation of the multilevel modulation

Sorter

The sorter has direct access to the digitally-converted values of the capacitor voltages. By sorting their values from the highest to the lowest, the sorter is able to generate a list of submodule indices, ranked in descending order.

When a large number of submodules must be compared and sorted, the algorithmic complexity of the sorting algorithm may become a challenge. In this case, it may be sufficient to only identify the (few) most, respectively least charged submodules. However, as the number of available I/Os on imperix controllers is limiting the number of submodules to N≤ 8 per arm anyway, the implemented sorting algorithm is a full sort (namely a simple even and odd sorting algorithm).

Balancer

The balancer is responsible for identifying the submodule to be switched (i.e. inserted or bypassed) as a function of the sorting results, arm current polarity, and current switching states. This process has multiple steps:

First, four candidate submodules are identified:
– The least charged module that can be inserted, ie. accept a voltage rising switching event `smallest_can_rise`
– The least charged one that can be bypassed, i.e. accept a voltage falling event `smallest_can_fall`

– The most charged one that can be inserted `largest_can_rise`
– The most charged one that can be bypassed `largest_can_fall`.

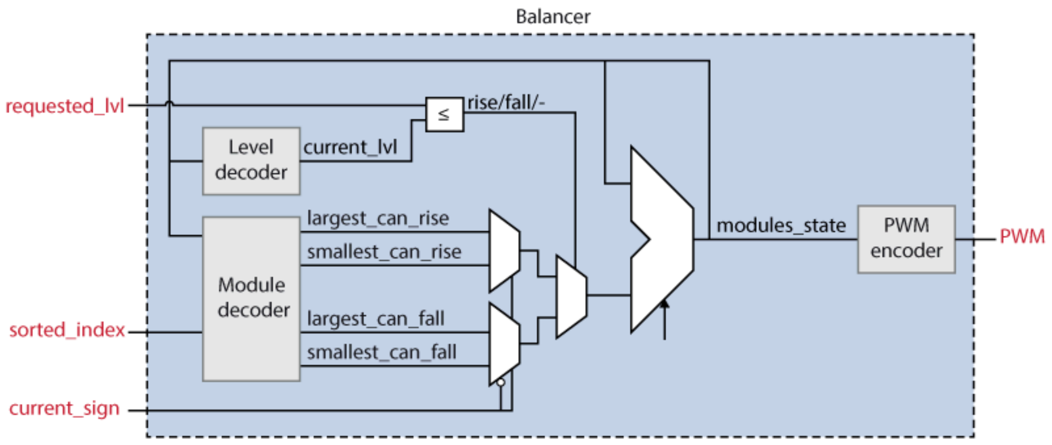
Second, according to the arm current polarity `current_sign`, only one submodule is selected for each type of switching transition (voltage rising or falling event).

In parallel, the current module states are decoded to identify the current level, i.e. the number of submodules that are currently inserted inside the arm. For half-bridge submodules, `current_lvl` is comprised within the range [0;N], respectively [-N;N] for operation with full-bridge submodules.

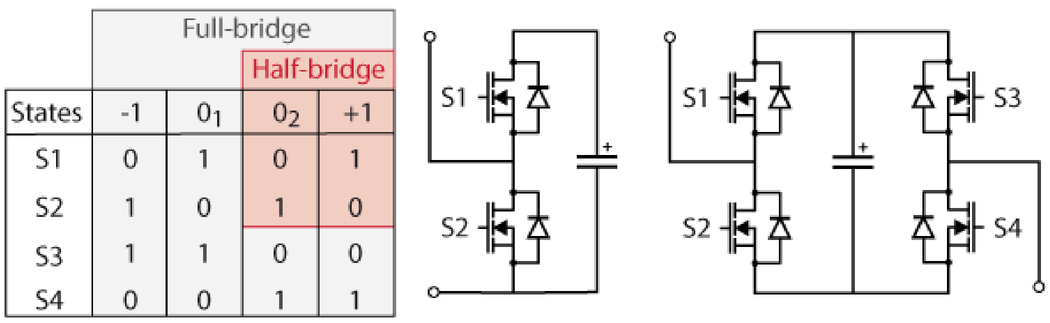
In parallel also, by comparing the instantaneous values of `current_lvl` and `requested_lvl`, the subsequent sampling period can be flagged with a transition type: `rise` if the requested level is higher, respectively `fall` if the requested level is lower. In case `current_lvl` and `requested_lvl` are equal, the transition type is – (none).

Third, based on the detected transition type, only one submodule is eventually selected among the two that were previously identified during the second step. This final candidate submodule is the only one that will see its state modified during the sampling period.

Finally, after defining the new state for the selected submodule, the PWM encoder defines the gate-driving signals for each power semiconductor using a basic truth table following table. When transitions occur, a configurable dead-time is introduced between complementary signals.



Implementation of the Balancer within the SS-PWM algorithm



	Full-bridge			
			Half-bridge	
States	-1	0_1	0_2	+1
S1	0	1	0	1
S2	1	0	1	0
S3	1	1	0	0
S4	0	0	1	1

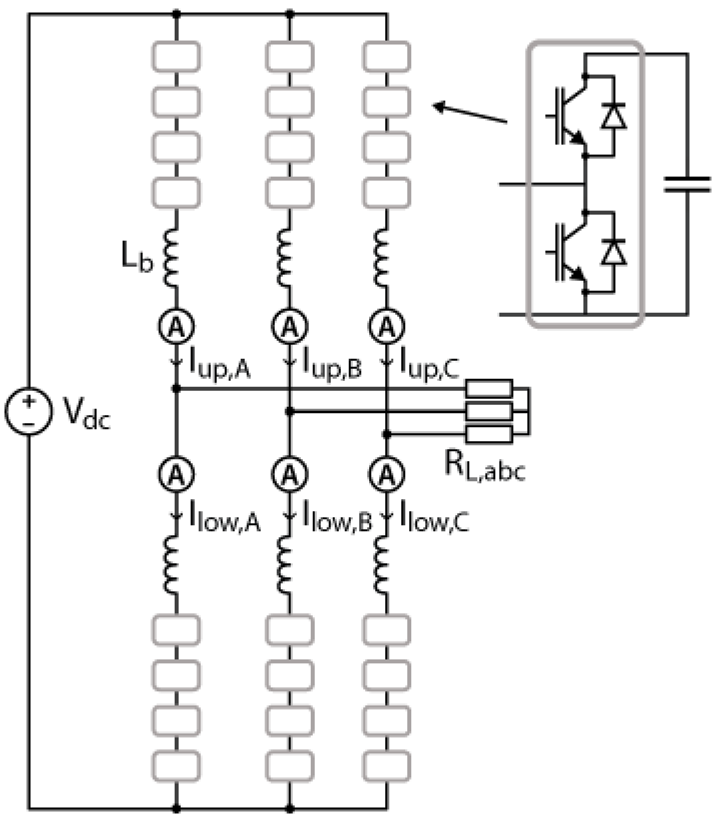
PWM generation from the submodules switching states

A few remarks are worth noting regarding the operation of the Balancer:

- Since only one submodule is switched at once, it is guaranteed that there are no “effectiveless” switching events, i.e. switching actions that result in no direct change of the arm voltage.
- This principle also guarantees that the ratio between the average switching frequency seen by each submodule and the apparent switching frequency is optimum.
- Due to this also, in case the `current_lvl` and `requested_lvl` differ by more than one level (such as in case of a very fast transient) only one change of the output level can be achieved per sampling period (4ns). Therefore, switching from the level -8 to +8 would require 48ns, which remains almost instantaneous for most applications.

Open-loop operation of SS-PWM for MMC

As an exemplary use case, Sort-&-Select modulation is applied to a Modular Multilevel Converter connected to a passive three-phase load. No converter-level balancing or energy control is performed, which also means that the so-called *circulating currents* are not shaped in a specific manner. Overall, this system is therefore controlled in a purely open-loop manner in the sense that there are no P, PI, or PID controllers regulating quantities explicitly.



Schematic of the use case example

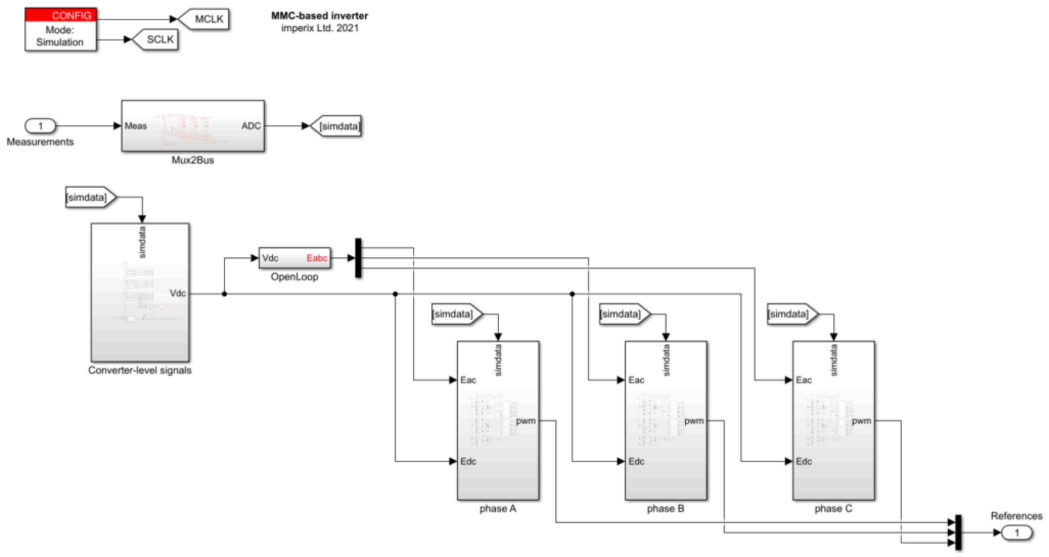
Software resources

The provided Simulink files contain the control algorithm and a model of the plant, allowing for both simulation and automated code generation for this scenario.

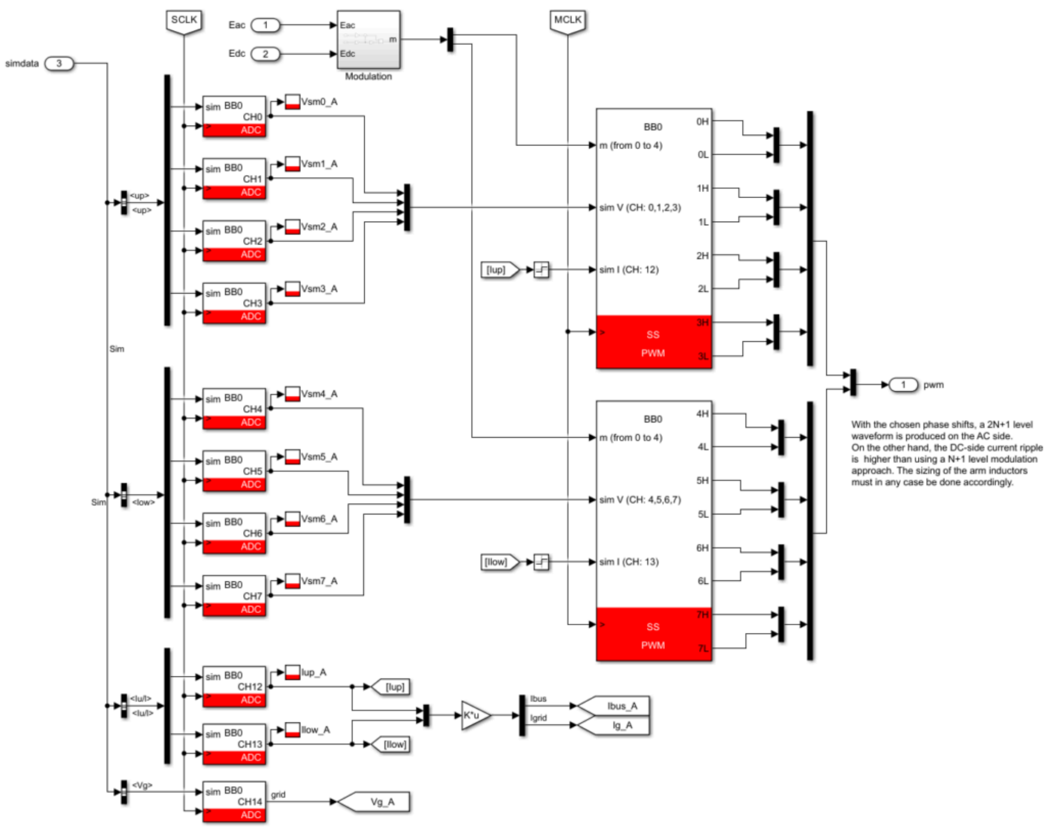
MMC_SSPWM_RloadDownload

Hardware and sofware configuration

The control model is organized using three main blocks, one for each phase-leg, which contain all the ADC and PWM configurations. In addition, the control model includes a basic "open-loop" waveform generation block that produces the requested AC voltages as a function of the available DC bus voltage.



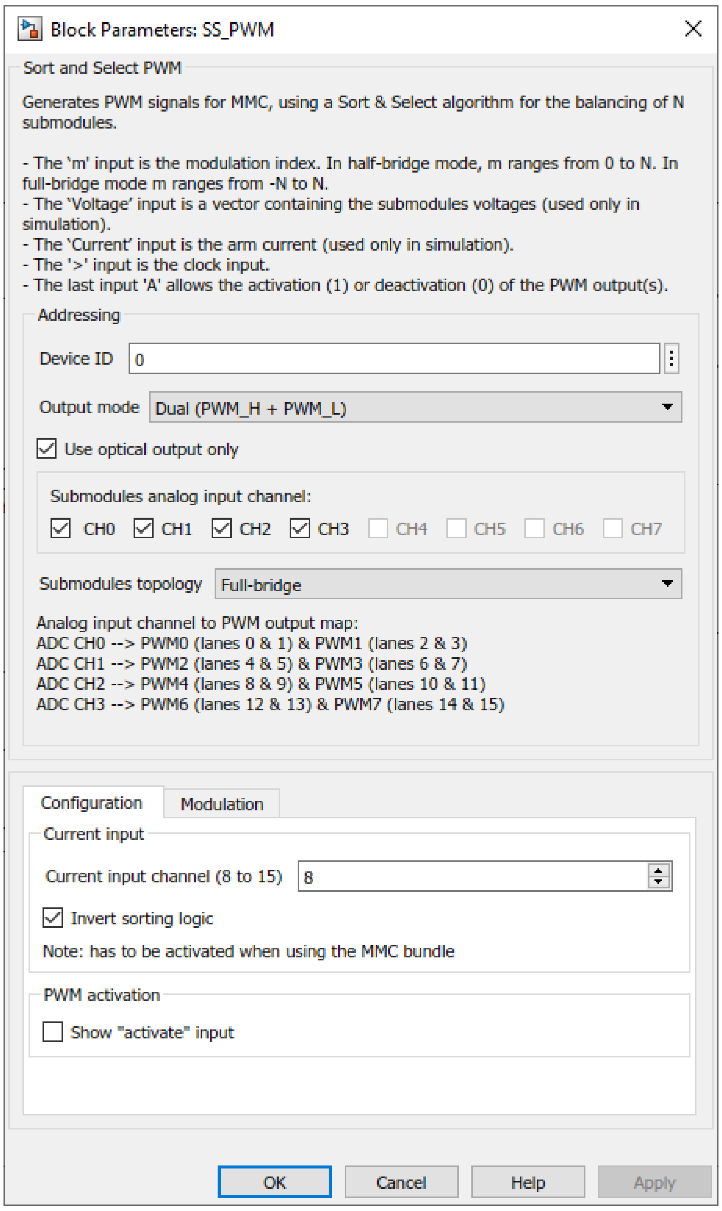
Simulink model overview



With the chosen phase shifts, a 2N+1 level waveform is produced on the AC side. On the other hand, the DC-side current ripple is higher than using a N+1 level modulation approach. The sizing of the arm inductors must in any case be done accordingly.

Simulink implementation of the balancing and modulation for each MMC phase-leg.

As seen above, one SS-PWM block is used for each MMC arm (meaning two blocks per phase-leg). They are configured as shown below:


Block Parameters: SS_PWM

Sort and Select PWM

Generates PWM signals for MMC, using a Sort & Select algorithm for the balancing of N submodules.

- The 'm' input is the modulation index. In half-bridge mode, m ranges from 0 to N. In full-bridge mode m ranges from -N to N.
- The 'Voltage' input is a vector containing the submodules voltages (used only in simulation).
- The 'Current' input is the arm current (used only in simulation).
- The '>' input is the clock input.
- The last input 'A' allows the activation (1) or deactivation (0) of the PWM output(s).

Addressing

Device ID: 0

Output mode: Dual (PWM_H + PWM_L)

☑ Use optical output only

Submodules analog input channel:
☑ CH0 ☑ CH1 ☑ CH2 ☑ CH3 ☐ CH4 ☐ CH5 ☐ CH6 ☐ CH7

Submodules topology: Full-bridge

Analog input channel to PWM output map:
ADC CH0 --> PWM0 (lanes 0 & 1) & PWM1 (lanes 2 & 3)
ADC CH1 --> PWM2 (lanes 4 & 5) & PWM3 (lanes 6 & 7)
ADC CH2 --> PWM4 (lanes 8 & 9) & PWM5 (lanes 10 & 11)
ADC CH3 --> PWM6 (lanes 12 & 13) & PWM7 (lanes 14 & 15)

Configuration | Modulation

Current input

Current input channel (8 to 15): 8

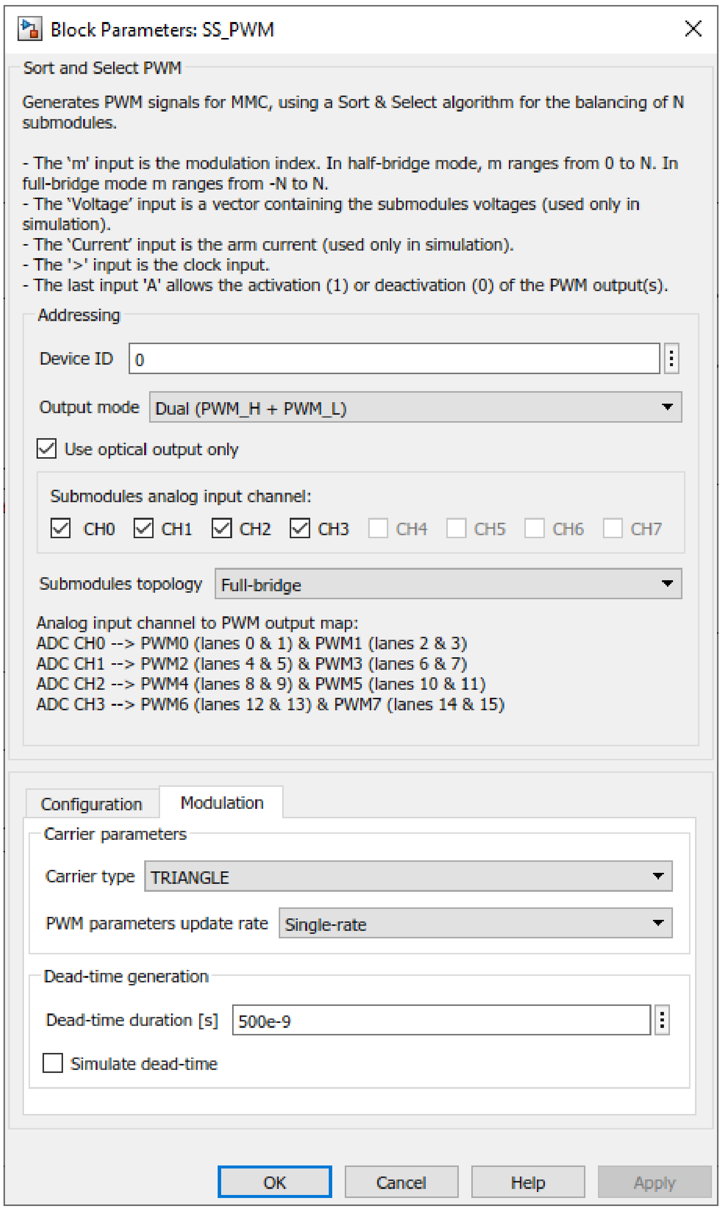
☑ Invert sorting logic

Note: has to be activated when using the MMC bundle

PWM activation

☐ Show "activate" input

Block mask of the SS-PWM (configuration tab)


Block Parameters: SS_PWM

Sort and Select PWM

Generates PWM signals for MMC, using a Sort & Select algorithm for the balancing of N submodules.

- The 'm' input is the modulation index. In half-bridge mode, m ranges from 0 to N. In full-bridge mode m ranges from -N to N.
- The 'Voltage' input is a vector containing the submodules voltages (used only in simulation).
- The 'Current' input is the arm current (used only in simulation).
- The '>' input is the clock input.
- The last input 'A' allows the activation (1) or deactivation (0) of the PWM output(s).

Addressing

Device ID: 0

Output mode: Dual (PWM_H + PWM_L)

[✓] Use optical output only

Submodules analog input channel: [✓] CH0 [✓] CH1 [✓] CH2 [✓] CH3 [ ] CH4 [ ] CH5 [ ] CH6 [ ] CH7

Submodules topology: Full-bridge

Analog input channel to PWM output map:
ADC CH0 --> PWM0 (lanes 0 & 1) & PWM1 (lanes 2 & 3)
ADC CH1 --> PWM2 (lanes 4 & 5) & PWM3 (lanes 6 & 7)
ADC CH2 --> PWM4 (lanes 8 & 9) & PWM5 (lanes 10 & 11)
ADC CH3 --> PWM6 (lanes 12 & 13) & PWM7 (lanes 14 & 15)

Configuration | Modulation

Carrier parameters

Carrier type: TRIANGLE

PWM parameters update rate: Single-rate

Dead-time generation

Dead-time duration [s]: 500e-9

[ ] Simulate dead-time

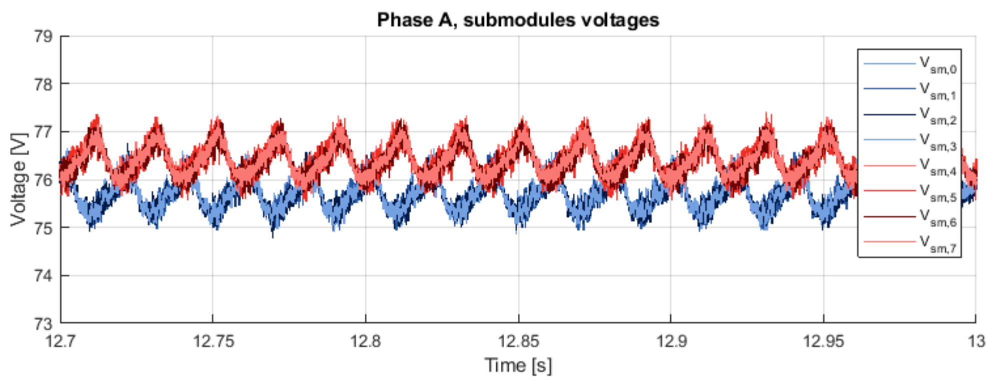
Block mask of the SS-PWM (modulation tab)

The `Submodules analog input channel` corresponds to the analog inputs and PWM outputs to which the submodules are connected. The block is configured to work with half-bridge modules and the `invert sorting logic` parameter is enabled since the modules used are PEH2015 (for further details on the configuration, please refer to the software documentation: SS-PWM).

Also, note that the inverted carrier is configured in all the modulators driving the lower arms. This way, a 2N+1 waveform is produced on the AC side which results in less filtering required, with the downside of having larger current ripples on the DC side.

Experimental results

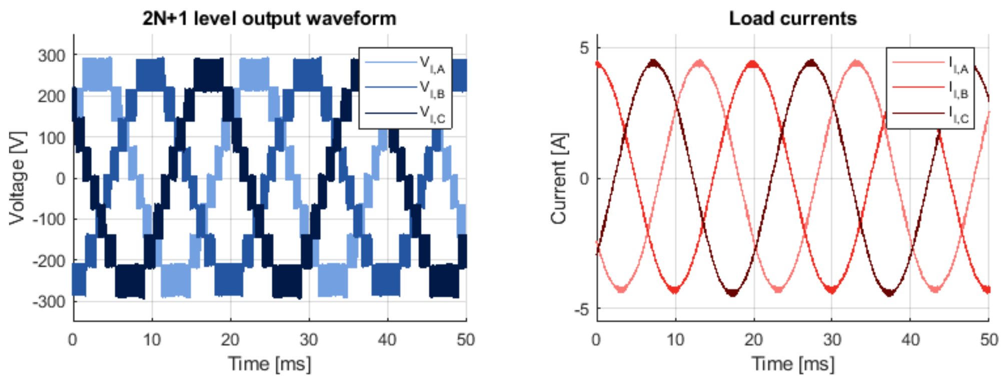
The typical performance of the SS-PWM for MMC peripheral block is shown below.



Balancing of the submodule voltages of an MMC.

It is however clear that the submodules belonging to the upper and lower arms are not well balanced together. Elementary techniques for converter-level balancing are introduced on the Three-phase MMC converter page.

The following plots show the PWM output waveforms of the MMC inverter. As expected, phase-to-neutral voltages show 2N+1 voltage levels, which results in very low ripple load currents.



Voltage and current waveforms for open-loop operation of a Modular Multilevel Converter.

References

[1] D.G. Holmes, T. Lipo “Pulse Width Modulation for Power Converters”, Wiley, ISBN 0471208140, 2003.
[2] P. W. Hammond, “Medium voltage PWM drive and method,” U.S. Patent 5625545, March 1994.
[3] M. Hagiwara and H. Akagi, “PWM Control and Experiment of Modular Multilevel Converters,” in Proc. PESC Conference, Rhodes, Greece, 2008 (read here).
[4] R. Marquardt, “Stromrichterschaltungen mit verteilten Energiespeichern,” German Patent DE10103031A1, Jan. 24, 2001.