ENTSO-E Standardized control interface for HVDC SIL/HIL conformity tests

21 April 2020

Last update 15 July 2020

Index

1. Management summary

Within this document, the ENTSO-E standardized control interface for HVDC SIL/HIL (HIL includes all kind of hardware in the loop test systems not limited to C-HIL and P-HIL)[1] conformity tests is defined. The standardized control interface shall support the interoperability workstream for a secure operation of the future European Energy system. It will also support the realization of interaction studies for multi-vendor AC and multi-vendor multi-terminal DC systems as required in the COMMISSION REGULATION (EU) 2016/1447 of 26 August 2016.

The European power system has been entering new territory as moving from a generator based to a converter based system approach. The standard control interface should help to improve the robustness of systems with regard to possible interactions and at the same time prepare vendors and network operators for the implementation of the new requirements of European regulation, ensuring good quality interaction studies, avoiding expensive actions after commissioning and secure the operations of the energy system of tomorrow.

As required in the COMMISSION REGULATION (EU) 2016/1447 all related parties identified by the relevant TSO, shall contribute to the interaction studies and shall provide all relevant data and models as reasonably required to meet the purposes of the studies. Furthermore the relevant system operator shall assess the compliance of an HVDC system throughout the lifetime of the HVDC system. The document describes a complete and standardized description of the control interface of an HVDC system in order to easily maintain the HVDC control model during lifetime. The provision of relevant data is compliant with the COMMISSION REGULATION 2016/1447 and will be executed when justified at project level..

The standard control interface described builds on existing knowledge and deliverables of EU H2020 project ‘Best Paths’ (http://www.bestpaths-project.eu/). Due to the convincing open available results from this project, the higher control layers and interfaces are already described. Standard gigabit Ethernet on optical POF 850nm laser has been selected as a suitable hardware (HW) interface for HVDC SIL/PHIL/HIL[1] conformity tests due to the excellent market availability, the cost efficiency and the easy implementation possibility in state-of-the-art FPGA which is essential for the manufacturer-specific bridges to the ENTSO-E standard. Within the document, the signal convention (depending on the given number of modules per arm of the specific HVDC application) as well as the hardware and physical communication standard are defined.

Furthermore, a standardized software interface for HVDC SIL/PHIL/HIL conformity tests has been defined. To ensure consistent results in HIL and SIL, the HW interfaces and protocols built the fundament for the signal conventions utilized in the standardized software interface.

Based on a short introduction on modelling of power-electronic assets, the standardized software interface for HVDC SIL/HIL conformity tests is described. Some of the interfaces have been demonstrated within a HIL setup in combination with a generic control system.

[1] SIL – Software In the Loop; HIL - Hardware In the Loop; CHIL – Controller Hardware In the Loop; PHIL – Power Hardware In the Loop

2. Definition of standardized electric circuit diagram

HVDC converter stations using Modular Multilevel Converters (MMC) topology usually apply the same structure irrelevant for vendor implementation. On the AC side, optional transformer, circuit breaker, grid filters are identified (cp. Figure 1). On the DC side, optional grid filter and circuit breaker are located. The points (electrical) where measurements are performed are at the grid-connection point. More specifically, three measurements of grid instantaneous value current and three measurements of instantaneous value grid voltages (measured against earth) are performed. Additionally, three measurements for current and voltage on the secondary side of the three-phase side are required (e.g. for synchronization). In every arm of the MMC station, a current measurement is performed (six measurement points) and every capacitor voltage of each submodule is measured. These measurements are necessary for balancing strategies and protection issues. For this control layer the sum of all capacitor voltages per arm is given. On the DC side, two currents and two voltages are measured.

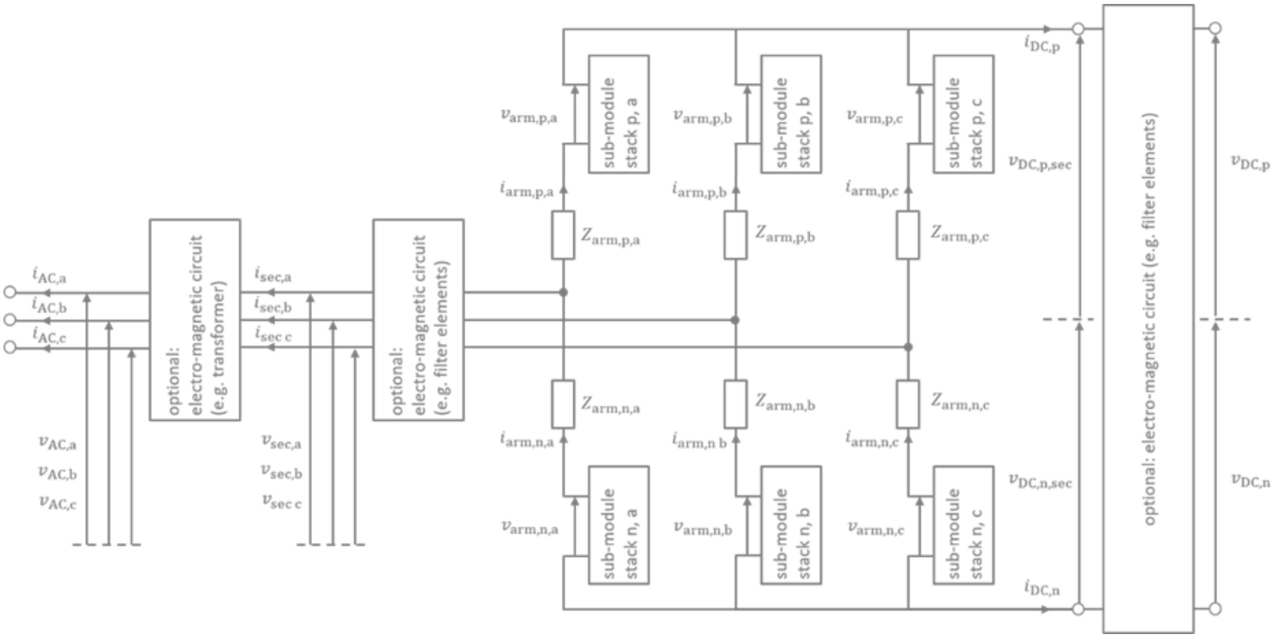

Figure 1: Electric-circuit diagram of HVDC with measurements

Besides the description of the measurements, the standardized control interface includes also the switching signals for the insulated-gate bipolar transistors (IGBTs) which are essential for the pulsed operation of the converter valves. The numbering of the IGBTs in the MMC modules includes four IGBTs with the numbers 1 to 4 in the first module, 5 to 8 in the second module and so on (cp. *Figure 2*, right-side diagram). In the case of half-bridge modules, only two IGBTs are in a module. Then, only the IGBT numbers 1 and 2 in the first module, 5 and 6 in the second module and so on are used (cp. *Figure 2*, left-side diagram). In the same manner, further converter topologies are considered if required by the network operator. Another possible module structure is a half-bridge module with a chopper IGBT included in each module. Here, the first three bits will be used.

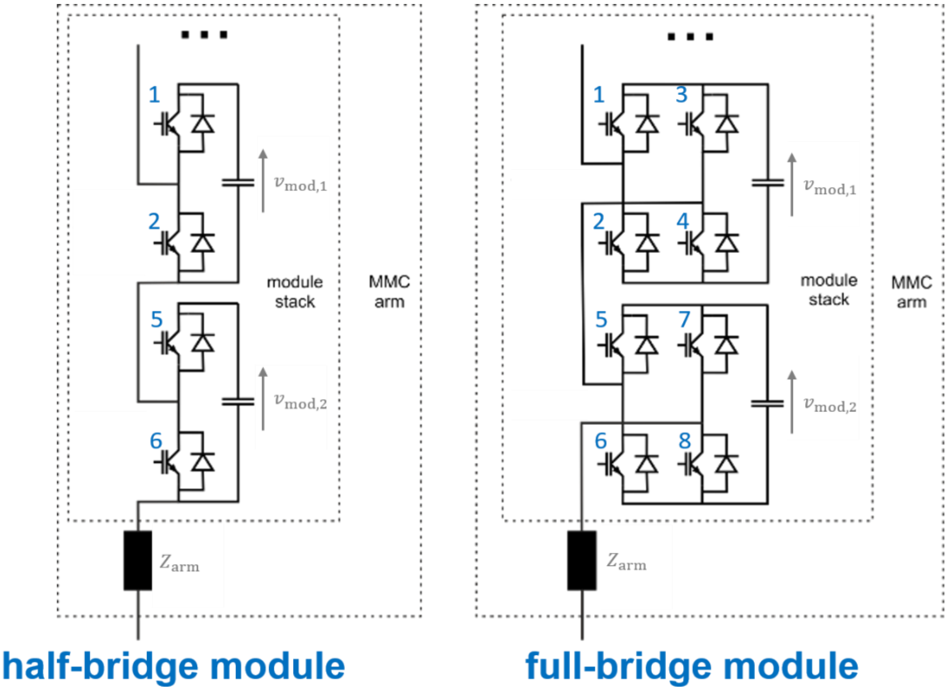

Figure 2: Numbering of IGBT inside the modules in every arm

3. Control layers and signals between the layers

In a typical HVDC station, four different control layers are used as given in the following description and Figure 3:

- **Grid-level control**
  Overlay control of the grid operator
- **Station-level control**
  Control of the complete station
- **Converter-near control**
  Voltage and current control of the HVDC
- **Module-near control and measurement-data acquisition**
  Control of the modules in the module towers including the module-voltage balancing

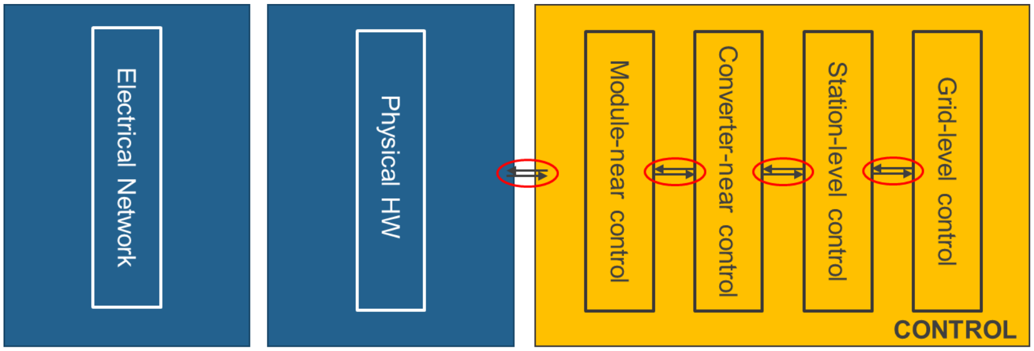

Figure 3: Control layers of HVDC station

Interface between Grid-level control and station-level control

This interface has been defined in the project ‘Best paths’ (http://www.bestpaths-project.eu/). The definitions achieved in this project shall be used (D9.3: BEST PATHS DEMO#2 Final Recommendations for Interoperability Of Multivendor HVDC Systems). The complete signal list with the master control is already defined under the standard IEC 61850 and will not be introduced in this document.

Interface between station-level control and converter-near control

For the interface between station-level control and converter-near control the following additional signals are considered:

Signals from station-level control to converter-near control

- AC-side active-power reference values: P_AC_pos_ref, P_AC_neg_ref
- AC-side reactive-power reference values: Q_AC_pos_ref, Q_AC_neg_ref
- AC-side voltage-magnitude reference values: U_AC_pos_ref, U_AC_neg_ref
- AC-side grid-frequency reference value: f_ref

- DC-side voltage-magnitude reference value: U_DC_ref
- Desired mode of operation

Signals from converter-near control to station-level control

- AC-side active-power actual values: P_AC_pos_act, P_AC_neg_act
- AC-side reactive-power actual values: Q_AC_pos_act, Q_AC_neg_act
- AC-side voltage-magnitude actual values: U_AC_pos_act, U_AC_neg_act
- AC-side grid-frequency actual value: f_act
- DC-side voltage-magnitude actual value: U_DC_act
- Actual mode of operation

Interface between converter-near control and module-near control and measurement data acquisition

Between the converter-near control and the module-near control and measurement data acquisition the switching signals for the circuit breakers and the reference values for the converter conversion ratios (collective modulation indices) are considered. In the opposite direction, the measurement values and the states of the towers and switches are considered.

Signals from converter-near control to module-near control and data acquisition

- Commands for Switch,AC,a, Switch,AC,b, Switch,AC,c, Switch,DC,p, Switch,DC,n
- Blocking command for module towers (for all six towers)
- Reference values for collective modulation indices (for all six towers)

Signals from module-near control and data acquisition to converter-near control

- Actual state of Switch,AC,a, Switch,AC,b, Switch,AC,c, Switch,DC,p, Switch,DC,n
- Actual blocking state of module towers (6)
- Actual AC-grid and secondary voltage-measurement values v_AC,a, v_AC,b, v_AC,c, v_AC,sec,a, v_AC,sec,b, v_AC,sec,c
- Actual AC-grid and secondary current-measurement values i_AC,a, i_AC,b, i_AC,c, i_sec,a, i_sec,b, i_sec,c
- Actual DC-grid and secondary voltage-measurement values v_DC,p, v_DC,n, v_DC,sec,p, v_DC,sec,n
- Actual DC-grid and secondary current-measurement values i_DC,p, i_DC,n, i_DC,sec,p, i_DC,sec,n
- Actual arm current-measurement values i_arm,p,a, i_arm,p,b, i_arm,p,c, i_arm,n,a, i_arm,n,b, i_arm,n,c
- Calculated collective module-capacitor measurement voltage v_mod_col,p,a, v_mod_col,b, v_mod_col,p,c, v_mod_col,n,a, v_mod_col,n,b, v_mod_col,n,c

Interface between module-near control and measurement data acquisition and the physical hardware

The physical hardware is not part of the control interface. The strict separation of the electrical power part from the control allows to investigate all possible fault scenarios within the electrical system in an absolute flexible manner. Furthermore, harmonics are very sensitive and can be influenced by physical hardware over lifetime. Therefore, harmonic control interaction studies as required in NC HVDC Art. 29 and Art. 70 needs separation of physical hardware and control. It shall be demonstrated that the standard control interface is implemented into the HVDC system model as described in detail in this document in accordance with article 54 COMMISSION REGULATION (EU) 2016/1447.

Signals from module-near control and measurement data acquisition to physical hardware

- Commands for Switch,AC,a, Switch,AC,b, Switch,AC,c, Switch,DC,p, Switch,DC,n
- Blocking command for module towers (for all six towers)
- 4 switching bits for all modules (for all six towers)

Signals from physical hardware to module-near control and measurement data acquisition

- Actual state of Switch,AC,a, Switch,AC,b, Switch,AC,c, Switch,DC,p, Switch,DC,n
- Actual blocking state of module towers (for all six towers)
- Actual AC-grid and secondary voltage-measurement values v_AC,a, v_AC,b, v_AC,c, v_AC,sec,a, v_AC,sec,b, v_AC,sec,c
- Actual AC-grid and secondary current-measurement values i_AC,a, i_AC,b, i_AC,c, i_sec,a, i_sec,b, i_sec,c
- Actual DC-grid and secondary voltage-measurement values v_DC,p, v_DC,n, v_DC,sec,p, v_DC,sec,n
- Actual DC-grid and secondary current-measurement values i_DC,p, i_DC,n, i_DC,sec,p, i_DC,sec,n
- Actual arm current-measurement values i_arm,p,a, i_arm,p,b, i_arm,p,c, i_arm,n,a, i_arm,n,b, i_arm,n,c
- All module-capacitor measurement voltages (for all six towers)

4. Definition of standardized hardware interface and protocols

The communication is completely digital; therefore, the measurement values are transformed with analogue-digital converters to digital values. In consequence, the measurement values are given in e.g. 12-bit values. All other values like the setpoints are also transformed to digital values (e.g. with 16 bit or 32 bit). Switching commands for IGBT, circuit breakers and disconnectors are transmitted in single bits. In the following subchapters, the hardware requirements and the proposed hardware and communication standard are given.

Hardware requirements

To identify the best hardware standard to be used for the transmission of the described signals between the different levels, first, the requirements determined by the bandwidth of the signals and typical sampling and delay times have to be identified. In the following listing, the number of bits which is transmitted between the layers is given (cp. Chapter 0 for more details of every communication signal).

Signals to be transmitted:

- Between station-level control and converter-near control
    - 288 bit (in both directions)
- Between converter-near control and module-near control and measurement data acquisition
    - 128 bit from converter-near control
    - 512 bit to converter-near control
- Between module-near control and measurement data acquisition and physical hardware (here for 1024 full bridge modules, switches and measurements)
    - 4096 bit + 32 bit from module-near control
    - 16384 bit + 416 bit to module-near control

Besides the number of bits, additionally, the typical sampling rate and acceptable delay times are crucial for the hardware definition since introduced delays – which are not there in reality – may result in stability issues. Both values are dependent on the functionality of every control level and are given in the following.

The station-level control typically has a sampling time of about 1 ms to100 ms. Based on the input and output values of this control layer, even a delay of several 100 µs is acceptable since the feedback loops in the station-level control have a relatively low bandwidth.

The converter near control has a typical control sampling time in the range of 10 µs to 400 µs. As the delay between this level and the station-level control can be several 100 µs, here, the defining factor is the delay from and to the module-near control and measurement-data acquisition. In typical HVDC applications the worst case delay is in sum about 80 µs. First, the measurement values have a typical delay time of about 10 µs. Additionally, in worst case, a complete control sampling time of 10 µs is added to the delay. In the pulse pattern, a statistical worst case delay of about 30 µs is added due to the number of modules to be switched and the IGBT switch time. The fourth factor is the communication delay, which is estimated in worst case to 30 µs. The total worst case delay time therefore is 80 µs. Based on this value, for the hardware protocol a maximum delay of 10 µs is demanded which is clearly lower than the worst case delay in real applications.

The module-near control and measurement acquisition has a typical sampling time in the range of 1 µs to 100 µs. The acceptable delay in the transmission of the pulse patterns is defined in the same range as in the converter-near control (10 µs). For the transmission of the measured capacitor voltages a maximum delay of about 20 µs is acceptable due to the maximum change in the capacitor voltages during one sampling time.

Based on these correlations, the aimed communication delay of 1 µs is proposed. This value is possible to be achieved with state-of-the-art signal processing hardware and is clearly lower than all values defined above. The highest number of bits which is needed to transmit is 16384 bit which is an unnecessary high requirement for the hardware solution. Instead, a transmission of 1024 bit of application data and 32 bit of control data in every 1 µs is proposed. To transmit more than 1024 bit of application data, several frames are combined (time-oriented multiplexing). E.g., in the first 1 µs, the first 1024 bit are transmitted via the interface. Subsequently, the second 1024 bit are transmitted via the interface. In consequence, after e.g. 16 µs the whole 16384 bit are transmitted. As this value is below 20 µs, this approach meets the requirements. This time-oriented multiplexing is used e.g. between converter-near and module-near control (details in chapter 5).

In addition to the 1024 bit, 32 bits are used in every frame for information concerning the frame number, the absolute number of frames and version information. The resulting requirement is a transmission of 1056 bit/ µs ≈ 1 Gbit/s which is a realistic requirement for FPGA implementation.

Besides the number of bits per µs, additional requirement should be met by the hardware solution. To minimize the EMC risk and, therefore, the damage risk, complete galvanic separation should be included. In consequence, an optical connection is optimal.

In order to reduce the project costs and the risk of a limited supply, cost-efficient market available components at least with a second-source supplier should be used. If possible, a hardware interface and physical communication protocol which is a common international standard should be used. Additionally, the protocol should be easy to implement in a FPGA due to the necessary realization of bridges between the ENTSO-E standard and the specific manufacturer interface for each level.

Hardware and communication standard

As described above, a suitable and market available hardware interface with a bit rate of 1024 bit/µs and optical connection has to be identified. As hardware interface the Network standard interface – 'standard SFP' (LC duplex, POF multimode, 850nm) is proposed. As communication protocol IEEE 802.3z for gigabit applications (8b/10) is proposed. These both selections fulfil all requirements and are widespread in many applications. Additionally, the full cost-efficient market availability is given. Besides, due to several available IC components from different manufacturers and available IP cores for all major FPGA manufacturers, sufficient sources and industrial solutions are available.

For the signal transmission the IEEE 802.3z point to point high-speed standard protocol is chosen. Based on these definitions, a prototype (only for the demonstrator) with market available FPGA, communication IC and optical transceiver is realized.

Data words

The first 32bit of every connection are used for addressing purposes. The first 12bit give the address, the next 12bit give the number of 32-bit words and the last 8bit give the version of the standard used.

With this definition the signals between all layers are transmitted in digital signals. These signals are given in 32-bit signed values. In every 32-bit signal, several signals like e.g. 32 switching signals, two 16-bit signals, one 32-bit signal or even the first or second half of a 64-bit signal can be included.

5. Interface signals on hardware

In this project the focus has been placed on the interfaces between the control layers as defined in chapter 5. In the following subchapters, the detailed definition of all signals and their position in the communication are given in tables.

The complete communication between the layers is realized with digital values. Here, a transmission of several 32-bit signals is used as standard. These 32bit can contain one single signal, two 16-bit signals or even more signals with single bits. The tables describing the signals have the following columns:

- **Signal name**
  In this column, the name of the signal is given
- **32 bit and 1 bit**
  The columns '32 bit' and '1 bit' contain the starting position of the signal. The entry in '32 bit' defines the number of the 32-bit value the signal is included in. In '1 bit' the starting position inside the 32-bit value is given.
- **Bit width**
  The column 'bit width' contains the information of the number of bits used for the signal.
- **Signal type**
  The signal type differs between 'real', 'int' and 'bit' values with the following characteristics:
  - Real: Is a value with floating point (which is only possible with at least 32 bit)
  - Int: Integer value
  - Bit: Single Bits
- **Default**
  Initial value of the signal
- **Min / Max**
  In the last two columns, the minimum and maximum value of the original signal is given. Based on this value and the bit width, the transmitted value is calculated, like given in this example: Bit width = 16, Min = -1000, Max = +1000 means the original signal is limited to Min and Max and then transformed to the 16-bit signal.

As described, the signals are transmitted in 32-bit blocks. In one transmission 33 of these 32-bit blocks are used (1056 bit are delivered in one transmission). If more than 1056 bits are needed, several frames are used. In every frame all 1056 bits are used sequently, first frame one, one step later frame two and so on. In consequence, if more than 1056 bits are needed, the signals are transmitted one frame after another.

Interface between Grid-level control and station-level control

This interface is the focus of the project 'Best paths' (http://www.bestpaths-project.eu/). The definitions achieved in this project can be used here.

Interface between station-level control and converter-near control

The signals between station-level control and converter-near control are assumed as 32-bit real values. Based on this definition several spares are still available in this interface which can be filled with additional values. If these additional values exceed the total number of 1024 bit, additional frames can be used.

Table 1: Signals from station-level control to converter-near control

Signal name	**32 bit**	**1 bit**	**Bit width**	**Type**	**Default**	**Min**	**Max**
Frame information	0	0	32	bit	0	0	2^32-1
Mode of operation	1	0	3	bit	0	0	8
Spare	1	3	29	bit	0	0	8
P_AC_pos_ref	2	0	32	real	0	-2	2
P_AC_neg_ref	3	0	32	real	0	-2	2
Q_AC_pos_ref	4	0	32	real	0	-2	2
Q_AC_neg_ref	5	0	32	real	0	-2	2
U_AC_pos_ref	6	0	32	real	0	-2	2
U_AC_neg_ref	7	0	32	real	0	-2	2
U_DC_ref	8	0	32	real	0	0	2
f_ref	9	0	32	real	0	0	2
Spare	10	0	32	real	0		
Spare	…	…	…	…	…	…	…
Spare	32	0	32	real	0		

Table 2: Signals from converter-near control to station-level control

Signal name	**32 bit**	**1 bit**	**Bit width**	**Type**	**Default**	**Min**	**Max**
Frame information	0	0	32	bit	0	0	2^32-1
Mode of operation	1	0	3	bit	0	0	8
Spare	1	3	29	bit	0	0	8
P_AC_pos_act	2	0	32	real	0	-2	2
P_AC_neg_act	3	0	32	real	0	-2	2
Q_AC_pos_act	4	0	32	real	0	-2	2
Q_AC_neg_act	5	0	32	real	0	-2	2
U_AC_pos_act	6	0	32	real	0	-2	2
U_AC_neg_act	7	0	32	real	0	-2	2
U_DC_act	8	0	32	real	0	0	2
f_act	9	0	32	real	0	0	2
Spare	10	0	32	real	0		
Spare	…	…	…	…	…	…	…
Spare	32	0	32	real	0		

Interface between converter-near control and module-near control and measurement data acquisition

Between the converter-near control and the module-near control and measurement data acquisition mainly the switching signals for the circuit breakers and the reference values for the converter voltage are transmitted. In the opposite direction, the measurement values and the states of the towers and switches are transmitted. With this definition, all signals can be transmitted in one frame.

Table 3: Signals from converter-near control to module-near control and data acquisition

Signal name	32 bit	1 bit	Bit width	Type	Default	Min	Max
Frame information	0	0	32	int	0	0	2^32-1
Switch,AC,a	1	0	1	bit	0	0	1
Switch,AC,a	1	1	1	bit	0	0	1
Switch,AC,a	1	2	1	bit	0	0	1
Switch,DC,p	1	3	1	bit	0	0	1
Switch,DC,n	1	4	1	bit	0	0	1
Block towers	1	5	6	bit	0	0	1
Spare (switches)	1	11	21	bit	0	0	1
rho,p,a	2	0	16	int	0	-1	1
rho,p,b	2	16	16	int	0	-1	1
rho,p,c	3	0	16	int	0	-1	1
rho,n,a	3	16	16	int	0	-1	1
rho,n,b	4	0	16	int	0	-1	1
rho,n,c	4	16	16	int	0	-1	1
Spare	5	0	32	Int	0		
Spare	…	…	…	…	…	…	…
Spare	32	0	32	int	0		

Table 4: Signals from module-near control and data acquisition to converter-near control

Signal name	32 bit	1 bit	Bit width	Type	Default	Min	Max
Frame information	0	0	32	int	0	0	2^32-1
Switch,AC,a	1	0	1	bit	0	0	1
Switch,AC,a	1	1	1	bit	0	0	1
Switch,AC,a	1	2	1	bit	0	0	1
Switch,DC,p	1	3	1	bit	0	0	1
Switch,DC,n	1	4	1	bit	0	0	1
Towers blocked	1	5	6	bit	0	0	1
Spare (switches)	1	11	21	bit	0	0	1
v_AC,a	2	0	16	int	0	-1000000	1000000
v_AC,b	2	16	16	int	0	-1000000	1000000
v_AC,c	3	0	16	int	0	-1000000	1000000
v_AC,sec,a	3	16	16	int	0	-1000000	1000000
v_AC,sec,b	4	0	16	int	0	-1000000	1000000
v_AC,sec,c	4	16	16	int	0	-1000000	1000000
i_AC,a	5	0	16	int	0	-10000	10000
i_AC,b	5	16	16	int	0	-10000	10000
i_AC,c	6	0	16	int	0	-10000	10000
i_AC,sec,a	6	16	16	int	0	-10000	10000
i_AC,sec,b	7	0	16	int	0	-10000	10000
i_AC,sec,c	7	16	16	int	0	-10000	10000
v_DC,p	8	0	16	int	0	-1000000	1000000
v_DC,n	8	16	16	int	0	-1000000	1000000
v_DC,sec,p	9	0	16	int	0	-1000000	1000000
v_DC,sec,n	9	16	16	int	0	-1000000	1000000
i_DC,p	10	0	16	int	0	-10000	10000
i_DC,n	10	16	16	int	0	-10000	10000
i_arm,p,a	11	0	16	int	0	-10000	10000
i_arm,p,b	11	16	16	int	0	-10000	10000
i_arm,p,c	12	0	16	int	0	-10000	10000
i_arm,n,a	12	16	16	int	0	-10000	10000
i_arm,n,b	13	0	16	int	0	-10000	10000
i_arm,n,c	13	16	16	int	0	-10000	10000
v_mod,p,a	14	0	16	int	0	-1000000	1000000
v_mod,p,b	14	16	16	int	0	-1000000	1000000
v_mod,p,c	15	0	16	int	0	-1000000	1000000
v_mod,n,a	15	16	16	int	0	-1000000	1000000
v_mod,n,b	16	0	16	int	0	-1000000	1000000
v_mod,n,c	16	16	16	int	0	-1000000	1000000
Spare	17	0	32	Int	0		

Signal name	32 bit	1 bit	Bit width	Type	Default	Min	Max
Spare	…	…	…	…	…	…	…
Spare	32	0	32	Int	0		

Interface between module-near control and measurement data acquisition and the physical hardware

The connection between the physical hardware and the module-near control and measurement data acquisition which includes the signals for switches and the measurement values also fit into one frame, leaving some spare signals.

Table 5: Signals from module-near control and measurement data acquisition to physical hardware (measurement cable)

Signal name	32 bit	1 bit	Bit width	Type	Default	Min	Max
Frame information	0	0	32	int	0	0	2^32-1
Switch,AC,a	1	0	1	bit	0	0	1
Switch,AC,a	1	1	1	bit	0	0	1
Switch,AC,a	1	2	1	bit	0	0	1
Switch,DC,p	1	3	1	bit	0	0	1
Switch,DC,n	1	4	1	bit	0	0	1
Block towers	1	5	6	bit	0	0	1
Spare (switches)	1	11	21	bit	0	0	1
Spare	2	0	32	Int	0		
Spare	…	…	…	…	…	…	…
Spare	32	0	32	Int	0		

Table 6: Signals from physical hardware to module-near control and measurement data acquisition (measurement cable)

Signal name	32 bit	1 bit	Bit width	Type	Default	Min	Max
Frame information	0	0	32	int	0	0	2^32-1
Switch,AC,a	1	0	1	bit	0	0	1
Switch,AC,a	1	1	1	bit	0	0	1
Switch,AC,a	1	2	1	bit	0	0	1
Switch,DC,p	1	3	1	bit	0	0	1
Switch,DC,n	1	4	1	bit	0	0	1
Towers blocked	1	5	6	bit	0	0	1
Spare for switches	1	11	21	bit	0	0	1
v_AC,a	2	0	16	int	0	-1000000	1000000
v_AC,b	2	16	16	int	0	-1000000	1000000
v_AC,c	3	0	16	int	0	-1000000	1000000
v_AC,sec,a	3	16	16	int	0	-1000000	1000000
v_AC,sec,b	4	0	16	int	0	-1000000	1000000
v_AC,sec,c	4	16	16	int	0	-1000000	1000000
i_AC,a	5	0	16	int	0	-10000	10000
i_AC,b	5	16	16	int	0	-10000	10000

Signal name	32 bit	1 bit	Bit width	Type	Default	Min	Max
i_AC,c	6	0	16	int	0	-10000	10000
i_AC,sec,a	6	16	16	int	0	-10000	10000
i_AC,sec,b	7	0	16	int	0	-10000	10000
i_AC,sec,c	7	16	16	int	0	-10000	10000
v_DC,p	8	0	16	int	0	-1000000	1000000
v_DC,n	8	16	16	int	0	-1000000	1000000
v_DC,sec,p	9	0	16	int	0	-1000000	1000000
v_DC,sec,n	9	16	16	int	0	-1000000	1000000
i_DC,p	10	0	16	int	0	-10000	10000
i_DC,n	10	16	16	int	0	-10000	10000
i_arm,p,a	11	0	16	int	0	-10000	10000
i_arm,p,b	11	16	16	int	0	-10000	10000
i_arm,p,c	12	0	16	int	0	-10000	10000
i_arm,n,a	12	16	16	int	0	-10000	10000
i_arm,n,b	13	0	16	int	0	-10000	10000
i_arm,n,c	13	16	16	int	0	-10000	10000
Spare	14	0	32	Int	0		
Spare	…	…	…	…	…	…	…
Spare	32	0	32	Int	0		

In the cable between the module-near control and the module tower all switching signals have to be delivered to the IGBT and all measured capacitor voltages to the module-near control. Due to the high number of modules in every tower of a HVDC station, this transmission is the biggest challenge in the interface definition. For every measurement signal 16 bit are used leading to the capacity to transmit 64 capacitor voltages per frame (with 32 available 32-bit blocks per frame, cp. *Table 1*).

Table 7: Signals from physical hardware to module-near control and measurement data acquisition (module cable)

32 bit	1 bit	Bit width	Type	Default	Min	Max	Frame 1	Frame 2	Frame 3	Frame 4
0	0	32	int	0	0	2^32-1	Frame information	Frame information	Frame information	Frame information
1	0	16	int	0	-10000	10000	v_c,1	v_c,65	v_c,129	v_c,193
1	16	16	int	0	-10000	10000	v_c,2			
2	0	16	int	0	-10000	10000	v_c,3			
2	16	16	int	0	-10000	10000	v_c,4			
3	0	16	int	0	-10000	10000	v_c,5			
3	16	16	int	0	-10000	10000	v_c,6			
4	0	16	int	0	-10000	10000	v_c,7			
4	16	16	int	0	-10000	10000	v_c,8			
5	0	16	int	0	-10000	10000	v_c,9			
5	16	16	int	0	-10000	10000	v_c,10			

32 bit	1 bit	Bit width	Type	Default	Min	Max	Frame 1	Frame 2	Frame 3	Frame 4
6	0	16	int	0	-10000	10000	v_c,11			
6	16	16	int	0	-10000	10000	v_c,12			
7	0	16	int	0	-10000	10000	v_c,13			
7	16	16	int	0	-10000	10000	v_c,14			
8	0	16	int	0	-10000	10000	v_c,15			
8	16	16	int	0	-10000	10000	v_c,16			
9	0	16	int	0	-10000	10000	v_c,17			
9	16	16	int	0	-10000	10000	v_c,18			
10	0	16	int	0	-10000	10000	v_c,19			
10	16	16	int	0	-10000	10000	v_c,20			
11	0	16	int	0	-10000	10000	v_c,21			
11	16	16	int	0	-10000	10000	v_c,22			
12	0	16	int	0	-10000	10000	v_c,23			
12	16	16	int	0	-10000	10000	v_c,24			
13	0	16	int	0	-10000	10000	v_c,25			
13	16	16	int	0	-10000	10000	v_c,26			
14	0	16	int	0	-10000	10000	v_c,27			
14	16	16	int	0	-10000	10000	v_c,28			
15	0	16	int	0	-10000	10000	v_c,29			
15	16	16	int	0	-10000	10000	v_c,30			
…	…	…	…	…	…	…	v_c,31			
…	…	…	…	…	…	…	v_c,32			
32	0	16	int	0	-10000	10000	v_c,63			
32	16	16	int	0	-10000	10000	v_c,64	v_c,128	v_c,192	v_c,256

To transmit the voltages of all 544 modules, 16 frames are needed, like given in the following listing.

- Frame 1: Modules 1 to 64
- Frame 2: Modules 65 to 128
- Frame 3: Modules 129 to 192
- Frame 4: Modules 193 to 256
- Frame 5: Modules 257 to 320
- Frame 6: Modules 321 to 384
- Frame 7: Modules 385 to 448
- Frame 8: Modules 449 to 512
- Frame 9: Modules 513 to 576
- Frame 10: Modules 577 to 640

- Frame 11: Modules 641 to 704
- Frame 12: Modules 705 to 768
- Frame 13: Modules 769 to 832
- Frame 14: Modules 833 to 896
- Frame 15: Modules 897 to 960
- Frame 16: Modules 961 to 1024

The four switching signals for every module are given in one bit per IGBT in the module. In consequence, in every frame the switching signals for 256 modules can be transmitted (with 32 32-bit blocks).

Table 8: Signals from module-near control and measurement data acquisition to physical hardware (module cable)

32 bit	1 bit	Bit width	Type	Default	Min	Max	Frame 1
0	0	32	int	0	0	2^32-1	Frame information
1	0	32	bit	0	0	1	switching,module,1 to 8
2	0	32	bit	0	0	1	switching,module,9 to 16
3	0	32	bit	0	0	1	switching,module,17 to 24
4	0	32	bit	0	0	1	switching,module,25 to 32
5	0	32	bit	0	0	1	switching,module,33 to 40
6	0	32	bit	0	0	1	switching,module,41 to 48
7	0	32	bit	0	0	1	switching,module,49 to 56
8	0	32	bit	0	0	1	switching,module,57 to 64
9	0	32	bit	0	0	1	switching,module,65 to 72
10	0	32	bit	0	0	1	switching,module,73 to 80
11	0	32	bit	0	0	1	switching,module,81 to 88
12	0	32	bit	0	0	1	switching,module,89 to 96
13	0	32	bit	0	0	1	switching,module,97 to 104
14	0	32	bit	0	0	1	switching,module,105 to 112
15	0	32	bit	0	0	1	switching,module,113 to 120
…	…	…	…	…	…	…	…
32	0	32	bit	0	0	1	switching,module,249 to 256

The switching signals to the IGBT of the modules are transmitted in 4 frames, like given in the following listing:

- Frame 1: Modules 1 to 256
- Frame 2: Modules 257 to 512
- Frame 3: Modules 513 to 768
- Frame 4: Modules 769 to 1024

6. Modelling of power-electronic assets

For the detailed analysis of e.g. HVDC interoperability, high-quality models are needed. Within this section, a brief introduction on modelling of power-electronic equipment is given.

In general, power-electronic equipment consist of components from three different types:

- Electric drive train (e.g. transformers, converters, inductors)
- Digital systems (e.g. control systems of HVDC converters)
- Control algorithms

To achieve a high modelling quality, each component type must be modelled according to its physical behaviour:

- Electric drive train: E.g. with differential equations or EMTP equations
- Digital system: Sampling system with input and output values and one set of control algorithms per system
- Control algorithms: Original algorithms have to be used
  (e.g. in compiled object like DLL in case of SIL)

Concerning the modelling of the digital systems and control algorithms, certain requirements have to be met. The original interfaces have to be used without changes since a standard interface is designed and the real interfaces have to be mapped on the standardized interface. It is recommended to use only algebraic calculations and not to use filters. Furthermore, for every physical control system, one digital system and a set of related control algorithm has to be considered. The set of control algorithms are e.g. compiled as 64bit DLL in case of Windows operating system.

Due to the relevant influence of interrupts of different control system, control algorithms located on different control systems are not allowed to share memory in the simulation. The signals shall be mapped in a transparent manner. In general, the control algorithms used in each DLL shall be identical to the control realization on the real asset control system.

7. General aspects of standard software interface

The standard control software interface shall provide a standard how relevant information shall be exchanged in order to simplify the compliance process for new HVDC connections. The standard software interface is fully in line with the COMMISSION REGULATION (EU) 2016/1447. As a consequence the provision of data is already given in the COMMISSION REGULATION 2016/1447 and shall not be treated in this document. The formal software definition for one control system is given. This definition has to be used for each relevant control system. These assignments can be found in the next section.

Within this section, the derivation of requirements for the standard software interface starting from the standard hardware interface defined is given. Subsequently, the necessary interface functions and the specific init, input and output structures are described.

Derivation of requirements for standard software interface

It is very important that no new signal convention is defined. The signal convention already defined for the standardized hardware interface has to be used. For a better reading comprehension, the relevant description is included subsequently.

In principle, the software interface has to emulate the hardware interface including the signal protocol. Furthermore, it is essential that the software interface is neutral. To put it into practice, the realization should not be optimised for a specific grid-calculation program / simulation. The aim is that the control algorithms can be used in any program which is able to call the specified library. In addition, functions have to be defined which emulate the real interrupts on a control hardware.

Interface of control functions

There are four functions which can be called by the simulation tool. In the interface header, these functions are only referenced but have to be included in the DLL. Every function receives a pointer to the structure control_data which includes a pointer to the input structure, the output structure and the internal structure. As described above, these structures are used to transfer input data and output data to the control algorithms in the DLL.
The needed memory is allocated outside of the DLL (excluding the internal structure which is not known outside the DLL). The complete allocation is finished before the first call of the functions. Therefore, all functions can access all structures directly.

- **int function_init (control_data_init const *args, control_data * self)**
  The function function_init is run once at the start of the simulation. Besides the pointer to the structure control_data, the structure control_data_init is referenced including all needed data for the initialization of the control software e.g. according to the specific asset parameters. The content of control_data_init is set before the call of the init-function. The included data is specific for each control.

- **int function_reset (control_data_init const *args, control_data * self)**
  function_reset can be run during the simulations. With this function the control software should be reset to the state valid after the initialization. The functions function_reset and function_init are separated because in function_init routines may be included which should not be executed one more time, like e.g. the allocation of memory.

- **int function_run (control_data * self)**
  function_run is the main function of the DLL and includes the calculation of the control-output data. This function only receives a pointer to the structure control data because there all needed information is included. The input data of the control is included in control_data_input while the actuating values and the output data has to be written to control_data_output. The call of this function is realized according to the sampling time set in the last run.
- **int function_release(control_data *self)**
  function_release is called once after all calculations. In this function, the memory is deallocated.

Data structures

In case of HVDC applications, several signals have to be exchanged. To optimize the handling of this large number of signals, the interface signals should be grouped according to their purpose (e.g. input, measurement, output). In hardware-near software, it is best practice to use vectors for grouping and handling many signals and structures for grouping and handling several vectors.

For the specific implementation, two structures are proposed – structure for init (control_data_init) and one structure for run (control_data). From this, the specific function calls with arguments results:

- **function_init (control_data_init *args_init, control_data *args_control)**
- **function_run (control_data *args_control)**

Content of interface structures

The structure control_data contains input, output and internal values (each in a new vector). Input and output are used for the signals in and out of this control. Every control can allocate internal values. With the pointer internal_structure the internal values can be addressed.

Values for initialization are given in control_data_init. The initialization algorithms and their parameters are very depending on the specific manufacturer realization. Thus, the values for initialization given in one vector.

Content of input structure

The inputs are separated in three vectors:

- **Measurement_data** contains the measurement values
- **Input_data** contains further input signals
- **System_data** contains e.g. information from the simulation tool
  (only used for optional information in this project, e.g. parametrization of converter-station information)

Content of output structure

The outputs are separated in five vectors

- **Output_data** is used for arbitrary outputs (e.g. in the grid-level control)
- **Modulation** is used for setpoints for converters
  (e.g. from converter-near control to module-near control)
- **Switching_word** is used for transmission of pulses to the converters, the circuit breakers and other switches
- **Status_data** is used for output of control status and comparable values (optional vector, maybe specified individually)
- **Sampling_time** is used to define the time when the control should be executed the next time

8. Assignment of defined signals to interface vectors

The standard hardware interface has already been defined. Several signals were defined with 1-bit, 16-bit and 32-bit values. Considering the requirement that the signals of hardware interface should be equal to the signals of the software interface, all signals from the hardware interface have to be assigned to the standard SW interface.

Float and integer values with 32-bit are used since hardware-near interfaces shall be chosen. In consequence, one 32-bit signal from the HW interface can be assigned directly to one value in the float vector. In analogy, two 16-bit signals from HW interface can be assigned to one value in the float vector (The first 16-bit signal is assigned to the lower 16 bit of the 32-bit float / integer, the second 16-bit-signal is assigned to the upper 16 bit). Additionally, 32 1-bit binary signals from HW interface can be assigned to one value in the float vector.

In the following subsection, the specific assignment for the different control layers is defined. Since multiplexing is necessary for exchange of the module-capacitor voltage measurements, several frames have to be addressed. In all levels, the frame information is the first entry in the vectors input_data and output_data.

Station-level control

- Input structure
    - Input_data
        - output_data of grid-level control
        - Output_data of converter-near control
- Output structure
    - Output_data
        - Input_data of grid-level control
        - Input_data of converter-near control

Control-near control

The converter-near control receives input signals from the station-level control and the module-near control and measurement acquisition. The calculated inputs are given in the vector input_data, the measurement values in measurement_data. The output signals are sorted according to their physical meaning. The signals to the station-level control are set in the vector output_data, the switching commands which are calculated in this level (e.g. for the circuit-breakers) are given in the vector switching_word while the modulation indices for the converter arms are given in the vector modulation.

- Input structure
    - Input_data
        - Output_data of station-level control
        - Output_data of module-near control (e.g. switching states)
    - Measurement_data

        - Output_data of module-near control
          (e.g. voltage and current measurements)
- Output structure
    - Output_data
        - Input_data of station-level control
    - Switching_word
        - Input_data of module-near control (e.g. circuit-breaker commands)
    - Modulation
        - Input_data of module-near control (modulation indices)

Module-near control

The module-near control receives the switching commands and the modulation from the converter-near control (all given in the vector input_data). Additionally, signals from the simulation tool are given in this vector (e.g. the switching states of the circuit breakers). The measurement values from the simulation tool are given in the vector measurement_data.

The output of this control level is transmitted to the converter-near control and the simulation tool in the vectors output_data and switching_word.

- Input structure
    - Input_data
        - Switching_word and modulation of converter-near control
        - SIL tool (e.g. switching states)
    - Measurement_data
        - SIL tool
          (e.g. voltage and current measurements, module-capacitor voltages)
- Output structure
    - Output_data
        - Input_data of converter-near control (e.g switching states)
        - Measurement_data of converter-near control
          (e.g. voltage, current and collective module voltages)
    - Switching_word
        - SIL tool
          (e.g. circuit-breaker commands and pulses for semiconductors)