On the Partitioning of MMC Control Systems Using Graph Theory

ILKA JAHN 1 (Member, IEEE), GERAINT CHAFFEY 2 (Member, IEEE), EDUARDO PRIETO-ARAUJO 3 (Senior Member, IEEE), MELANIE HOFFMANN 4, RODRIGO ALVAREZ 5, AND ANTONELLO MONTI 1,6 (Senior Member, IEEE)

1Institute for Automation of Complex Power Systems, RWTH Aachen University, 52074 Aachen, Germany
2EnergyVille/Electa Research Group, Electrical Engineering Department ESAT, KU Leuven, 3001 Heverlee, Belgium
3CITCEA-UPC, 08028 Barcelona, Spain
4Elenia Institute for High Voltage Technology and Power Systems, Technische Universität Braunschweig, 38106 Brunswick, Germany
5Siemens Energy Global GmbH & Co. KG, 91058 Erlangen, Germany
6Fraunhofer FIT, 52074 Aachen, Germany

CORRESPONDING AUTHOR: ILKA JAHN (e-mail: ilka.jahn@eonerc.rwth-aachen.de)

The work of I. Jahn was supported by the Excellence Strategy of the German Federal Government and the Länder (Junior Principal Investigator Grant at RWTH Aachen). The work of G. Chaffey was supported by the Belgian Energy Transition fund, Project Neptune. E. Prieto is Lecturer of the Serra Húnter Programme.

ABSTRACT Control systems of modular multilevel converters can be partitioned into several levels. Such partitioning is becoming relevant in multi-vendor high-voltage direct-current systems where one party may deliver the converter hardware and associated control, whereas another party may deliver the system-related control. However, the exact control system partitioning has not been subject to research. This paper presents a method for partitioning control systems into two levels with minimum dependencies between the two levels. Furthermore, the impact of specific control designs on the partitioning and integration effort is discussed using an example. The presented method is useful for the design of multi-vendor control systems for modular multilevel converters whose partitioning otherwise would be based on engineering intuition.

INDEX TERMS Graph theory, modular multilevel converters, control system partitioning.

I. INTRODUCTION

The all power electronics grid will likely have a backbone structure consisting of high-voltage direct-current (HVDC) transmission links or HVDC multi-terminal systems. Many single-vendor point-to-point HVDC links are in operation. However, the implementation of multi-vendor or multi-terminal HVDC systems requires careful coordination.

In this context, the control systems of the modular multilevel converters (MMCs) have to be designed to accommodate a multi-vendor approach, particularly if one party delivers the converter hardware and associated control, whereas another party delivers the system-related control. Such a setup is becoming a more commonly used approach in multi-vendor HVDC systems according to [1] stating that “...the external controller and the internal controller are usually provided by different vendors...”. Hierarchical control systems with partitioning into an upper and a lower level are nothing new, as many standards and publications show, e.g., [2]–[4]. Though it is usually not explicitly stated, it can be assumed that, ideally, the control functions placed in the upper level should be concerned with the MMC’s system behavior. In contrast, the control functions placed in the lower level should ideally be concerned with the MMC internal behavior and hardware requirements. In such an ideal scenario, the partitioning into upper level and lower level can be assumed to be straightforward, as shown in Fig. 1(a).

Unfortunately, some control functions usually placed in the lower level can impact the system behavior. Vice-versa, some upper-level control functions concerned with the MMC system behavior depend on the MMC lower-level control and hardware (see Fig. 1(b)). In other words, the MMC hardware dimensioning and associated lower-level control determine what functionality the MMC can deliver to the system, e.g., during fault. As a result, it is not straightforward to partition an MMC control system into an upper and a lower level.

A real example of an HVDC system with multi-vendor control is the Johan Sverdrup project with two parallel HVDC links by two vendors and a third-party device providing...
coordinating control [5], [6]. Another example is the Zhangbei project with vendor-specific MMC valves [7] and upper-level controls supplied by another vendor. Similarly, the three-terminal Nan’ao system in Fig. 2 has a vendor-specific valve and submodule control [8], [9]. Note that the circulating current control is found in the lower control level. However, it is not explicitly indicated whether the DC- and AC-side current controllers are placed in the lower or upper control level.

The placement of specific control functions into specific control levels has not been subject to research, and the underlying reasoning is not addressed in standards or information on real MMC control systems. This gap leads to several challenges. For example, if there are dependencies between the upper and lower control level, sensitive vendor-specific implementation aspects and hardware limits in the lower level may have to be disclosed for the optimal design of the upper level. Also, if the upper and lower control levels are designed by different parties (vendors, transmission system operators (TSOs), consultancies) with a non-optimal partitioning, that is, many dependencies, likely the integration effort will be unnecessarily high — which can be perceived as a blocker for the implementation of multi-vendor HVDC systems. Furthermore, it can be argued that the result of a non-optimal control system partitioning will be the overdesign of the converter hardware, such as to provide margins in case the upper-level control puts unexpectedly high requirements on the hardware. For all these challenges, the underlying and central unaddressed aspect is that the exact partitioning of control systems in multi-vendor setups seems subject to an engineering decision and not a thorough investigation.

This paper addresses this gap by proposing a graph-theoretic method for partitioning MMC control systems. In particular, the integration effort motivates partitioning, where a minimum dependency between an upper and a lower control part is achieved. A minimum dependency refers to physical interfaces (e.g., references) between the upper and lower control levels, as well as, information to be shared (e.g., about lower control level needed for the design of the upper level). Lastly, specific control functions can be manually assigned to either the upper or lower control level before partitioning with a minimum dependency.

This paper is structured as follows: Section II describes the main idea of how control systems can be partitioned using graph theory. Section III presents the graph-theoretic method for this purpose. Section IV illustrates the control system under study, and Section V the different test cases. Section VI presents the results of the partitioning. Section VII discusses these results from different perspectives. Finally, Section VIII draws the conclusion.

II. PROCEDURE FOR PARTITIONING

The proposed procedure for partitioning an MMC control system using graph theory requires the following definitions. The procedure is described in Section II-F.

A. GRAPH REPRESENTATION

An MMC control system is represented as an undirected graph consisting of nodes and edges.

A node is used to represent an MMC control function. Two additional nodes are used as criteria for partitioning the control system graph into two parts. In this paper, these two nodes are “system relevance” and “hardware relevance”.

An edge is used to represent an MMC control signal between two control functions. Furthermore, edges can be added to represent a conceptual connection between a control function and “system relevance” or “hardware relevance”.

B. MINIMUM CUT

The minimum cut between two graph nodes is achieved when a minimum sum of (weighted) edges is cut. In this paper, the cut is performed between node “system relevance” and node “hardware relevance”. The minimum cut can be interpreted as minimum dependency between two graph parts. As a result, the minimum cut can be used to numerically assess the integration effort of a specific control system partitioning.

C. SYSTEM RELEVANCE (NODE)

System relevance means that a control function has an impact on the MMC’s system behavior. System relevance is indicated with an edge from the control node in question to the system relevance node. An example is the active power control, whose design is relevant for the MMC’s system behavior [11].

D. HARDWARE RELEVANCE (NODE)

Hardware relevance means that a control function is closely connected with the converter hardware. Hardware relevance is indicated with an edge from the control node in question to the hardware relevance node. An example is the current control, whose design is relevant so as not to surpass the MMC’s current limits and the capacitor voltage limits [11].

E. GRAPH EDGE WEIGHT

The edge weight represents how strongly two nodes are connected. In this paper, all edges within the control system graph, i.e., control signals, are chosen to have an edge weight of one. The additional edges connecting a control system node to the “system relevance” or “hardware relevance” node
can be assigned a larger edge weight to model the degree of relevance a control system node has for either the system or the hardware. In this paper, two options are available to model the degree of relevance, as shown in Fig. 3. For hands-on examples, it is referred to Section IV-B.

1) PARAMETERIZATION-RELATED RELEVANCE (EDGE WEIGHT 10)
Parameterization-related relevance means that the parameterization (e.g., gains) of a control function is relevant for the behavior on the system or converter hardware level, whereas the specific implementation of that function (e.g., detailed block diagram, underlying hardware limits/configuration) are not relevant. A parameterization-related relevance could be described using the parameters and a suitable description of the implemented functionality (transfer function, representative block diagram). Modifying the parameters (e.g., gains) is sufficient to change the behavior of a function with parameterization-related relevance. In this paper and as an example, the parameterization-related relevance is assigned an edge weight of ten.

2) IMPLEMENTATION-RELATED RELEVANCE (EDGE WEIGHT 100)
Implementation-related relevance means that both the parameterization (e.g., gains) and additional implementation-related knowledge of a control function are relevant for the behavior on the system or converter hardware level. Implementation-related relevance includes the function’s parameterization, but in addition, an implementation aspect (e.g., the sizing and/or configuration of hardware components, control system sampling and/or delays, the detailed block diagram) is relevant for the function’s behavior. To analyze the behavior of a function with implementation-related relevance, the parameterization and its interplay with implementation-specific aspects have to be understood. To change the behavior of a function with implementation-related relevance, ultimately, not only the parameterization but also specific implementation aspects may have to be changed. In this paper and as an example, the implementation-related relevance is assigned an edge weight of one hundred (compared to ten for parameterization-related relevance). This higher edge weight was chosen because more aspects than the simple parameterization are relevant for the behavior of a function with implementation-related relevance.

F. STEPS
Using the previous definitions, it is proposed to partition an MMC control system using four steps as illustrated in Fig. 4.
JAHN ET AL.: ON THE PARTITIONING OF MMC CONTROL SYSTEMS USING GRAPH THEORY

III. GRAPH PARTITIONING METHOD

The graph partitioning method used in this paper is based on the minimum cut - maximum flow theorem [12]. The minimum cut between two graph nodes separating a graph into two parts corresponds to the maximum flow between those nodes1.

The maximum flow problem was first solved in 1956 and was presented and proven in [12]. The first algorithm that guaranteed termination was presented in the 1970s [14] with an upper runtime bound of $O(NE^2)$ with $N$ the number of graph nodes and $E$ the number of graph edges. Over the years, many more implementations with smaller upper runtime bounds were presented.

The “direct” minimum cut problem was first solved in 1995 [15] with an upper runtime bound of $O(N^2|E| + |N|^2\log|N|)$. Even though solving a maximum flow or minimum cut problem yields the same result, note that maximum flow algorithms are more established than minimum cut algorithms. Furthermore, maximum flow algorithms are readily available in different software, e.g., MATLAB, or R, Python, and C [16]. For that reason, a maximum flow algorithm is used for the method proposed in this paper. Note that algorithm runtime is not critical in the scope of this paper investigating small graphs (but would also favor maximum flow algorithms). For a better interpretation of the problem at hand, the remainder of this paper will present information using edge weights (not flows) even though a maximum flow algorithm is used for graph partitioning. For an overview on graph partitioning methods, it is referred to [17].

In power electronics, graph theory has been used for dynamic analysis [18], [19] (using switching flow graphs), cooperative control [20] (using directed communication graphs), converter topology simplification [21] (using node merging and path elimination), converter topology derivation [22] (using graph search), and chip layout optimization [23] (using constraint graphs). Another prominent example of using the minimum cut - maximum flow theorem is image segmentation. To the authors’ knowledge, graph-theoretic methods have not been used for control system partitioning, and vice versa, control system partitioning has not been subject to research.

IV. SYSTEM UNDER STUDY

The system under study is a typical control system of a single MMC-HVDC converter shown in Fig. 5. This implementation uses an energy-based approach and builds upon [10]. A standard MMC circuit is assumed, as presented, e.g., in [11], including a measurement distribution providing the required inputs for each control function block in Fig. 5, e.g., measured voltages, currents, angles, active/reactive powers, etc. Note that for better visibility, Fig. 5 only contains a strongly reduced representation of the measurement distribution.

Each control function block is represented by a node in a control graph and is assigned a number. The nodes are assigned a “system relevance” or “hardware relevance” (see Sections IV-A and IV-B, not shown in Fig. 5) with parameterization-related or implementation-related edge weight. Most nodes are assumed with a fixed degree of system or hardware relevance. However, some nodes are assumed with different degrees of system or hardware relevance depending on their design. Note that these assumptions serve the purpose of illustrating the developed partitioning method. Depending on a specific control system or control function design, other degrees of relevance (for the system or hardware) may apply.

1The flow of a graph can be calculated with, e.g., [13] and has additional constraints on edge capacity (maximum flow through an edge is the edge weight) and flow conservation (what flows into a node has to flow out).
FIGURE 5. Control system under study.

A. FIXED RELEVANCE DEGREE
The nodes 1 - 19, 21 - 23, 25, 28 - 29, and 31 - 37 are assumed with a parameterization-related relevance for the system because changing the parameters of the control functions is assumed to change the MMC’s system response.

The nodes 20 - 39 are assumed with an implementation-related relevance for the hardware because not only their parameterization but also their interplay with implementation aspects are assumed to determine the impact of those functions on the converter hardware.

B. RELEVANCE DEGREE DEPENDING ON FUNCTION DESIGN
To illustrate the developed graph method for control system partitioning, some nodes are assumed with parameterization-related or implementation-related relevance (for either system or hardware):

- 19 I_Q modulation and I_P/I_Q priority and capability limitation (on hardware),
- 20 Active power control (on system),
- 24 Current control (on system),
- 25 AC voltage limitation (on system).

For node 19, I_Q modulation and I_P/I_Q priority and capability limitation, an implementation-related hardware relevance is assumed if the underlying real hardware limitations have to be known to understand if the function will make the converter go into a forbidden operating point [11]. A parameterization-related hardware relevance is assumed if the design is in per unit instead of revealing actual hardware values.

For node 20 Mode 1: Active power control, a parameterization-related system relevance can normally be assumed. An implementation-related system relevance, however, may be assumed if software filtering is used such that the filter implementation becomes relevant for the function’s system behavior [11].

For node 24 Current control, a parameterization-related system relevance is assumed because the parameterization determines the answer to a disturbance. However, the underlying implementation (e.g., sampling and delays) may have an impact in higher frequency regions [24] and therefore also an implementation-related system relevance may be assumed.

For node 25 AC voltage limitations, it is assumed that this function is only noticeable on the system level if the converter runs into a forbidden operating point and limits the AC voltage to protect its hardware [11]. It is assumed that this function can be designed with parameterization-related system relevance (e.g., in p.u.) but also with implementation-related system relevance requiring revealing specific hardware aspects to fully understand the system behavior.

For node 29 DC voltage control or feedforward, a parameterization-related relevance for the system is assumed if a DC voltage controller is used. If instead a feedforward is used no relevance on the system is assumed [11].
TABLE 1. Results

|                      | Edge weight | Case 1 | Case 2 | Case 3 | Case 4 |
|----------------------|-------------|--------|--------|--------|--------|
| System under study   |             |        |        |        |        |
| parameterization-related system relevance | 10 | (1) – (19) | (1) – (25) | See 1a but manually assigned to lower level (20) (24) (25) |
| implementation-related system relevance | 100 | (20) (24) (25) | none | See 1a but manual partitioning according to standards and reference architectures, compare Fig. 5 |
| parameterization-related hardware relevance | 10 | (20) | (20) (19) |
| implementation-related hardware relevance | 100 | (19) (20) (24) | (20) (24) (25) |
| nodes in upper level | (same nodes are highlighted red in Fig. 6), indicated in light gray are the nodes with hardware relevance | 11 – 19 | 11 – 19 | see case 2 | 11 – 18 |
| nodes in lower level | (same nodes are highlighted black in Fig. 6), indicated in light gray are the nodes with system relevance | 10 – 19 | 10 – 19 | see case 2 | 10 – 19 |
| physical interfaces | 19 (19) (19) (19) | 20 (20) (20) (20) | 21 (21) (21) (21) | 22 (22) (22) (22) | see case 2 | 37 – 38 |
| physical signal cuts | 360 | 176 | 546 | 1412 |
| upper-level nodes with hardware relevance | 10 | 6 | 6 | 2 |
| lower level nodes with system relevance | 13 | 16 | 16 | 5 |
| total edge weight of minimum cut | 360 | 176 | 546 | 1412 |

V. TEST CASES

Four test cases will be investigated. In each test case, the number of physical signal cuts, the number of upper-level nodes with hardware relevance, the number of lower-level nodes with system relevance, as well as the total weight of the edge cut will be evaluated.

A. CASE 1: MOSTLY IMPLEMENTATION-RELATED RELEVANCE

In test case 1, the nodes 19, 20, 24, 25 are assumed with an implementation-related relevance for system or hardware, respectively (compare Section IV-B). Node 29 is assumed with a parameterization-related system relevance.

B. CASE 2: CHANGE FROM IMPLEMENTATION-RELATED TO PARAMETERIZATION-RELATED RELEVANCE

In test case 2, with the reasoning in Section IV-B, the nodes 19, 20, 24, 25 are changed such that they only have a parameterization-related relevance for system or hardware (instead of implementation-related). Also, node 29 is changed such that it has no more system relevance. Compared to test case 1, test case 2 has modifications corresponding to an improvement, meaning that the node relevance on either the system or the hardware is less than before. The assumption is that this makes the partitioning more straightforward.

C. CASE 3: MOSTLY STRUCTURAL IMPACT WITH MANUAL NODE ASSIGNMENT

In test case 3, the same node relevance for either system or hardware is assumed as in case 1. Additionally, the nodes that were placed in the upper control part in case 1 (and which still have a hardware relevance) are now manually assigned to the lower control level.

D. CASE 4: PARTITIONING ACCORDING TO STANDARDS / REFERENCE ARCHITECTURES

In test case 4, no graph theory is used for partitioning. The MMC control system is manually partitioned according to standards and published reference architectures [2], [8], [9]. The interface between the upper and lower control level is at the arm reference voltage with the circulating current control placed in the lower control level as shown in Fig. 5.

VI. RESULTS

The results for the different test cases are shown in Table 1. The resulting partitioned control system are shown in Fig. 6 for test case 1 and 2.

Test case 1 results in the partitioning into upper and lower control level as indicated in Fig. 6(a). Three nodes in the upper level have a hardware relevance, and 13 nodes in the lower level have a system relevance. The physical interface between the upper and lower control parts consists of 10 signals. The total weight of the minimum cut is 360.

Compared to test case 1, test case 2 results in a partitioning with nodes 20, 24 and 25 in the lower level as shown in Fig. 6(b). This is achieved by changing the design of node 19 from implementation-related to parameterization-related hardware relevance, nodes 20, 24, 25 from implementation-related to parameterization-related system relevance, and node 29 from parameterization-related to no system relevance. This means that the edges connecting these nodes with “system relevance” have less weight such that the “hardware
relevance” node has more influence resulting in the nodes being assigned to the lower level. No more node in the upper level have a hardware relevance. 16 nodes in the lower level have a system relevance. The physical interface between the levels consists of six signals. The total weight of the minimum cut is 176.

Test case 3 is based on test case 1. However, nodes 22 - 24 are manually assigned to the lower level such that no nodes with hardware relevance are present anymore in the upper level. The resulting partitioning is identical to test case 2. 16 nodes in the lower level have a system relevance. The physical interface between the upper and lower control levels consists of six signals. However, the total weight of the minimum cut is 546 and much higher than in test case 2 even though the identical partitioning is achieved.

Test case 4 with manual partitioning according to reference architectures and standards results in 15 nodes in the upper level with hardware relevance and five nodes in the lower level with system relevance. The physical interface between the upper and lower control level consists of only two signals, which is the lowest for all test cases. However, the total weight of the minimum cut is 1410 and the highest for all test cases.

VII. DISCUSSION
The presented method for partitioning of MMC control systems can be discussed from different perspectives.

A. LIMITATIONS
In the study done in this paper, the edge weight for a specific type and degree of relevance is chosen based on different observations (compare Section IV-B) on how the parameterization or implementation of a control function is relevant for either system or hardware. In this paper, the edge weights of 10 (parameterization-based relevance) and 100 (implementation-based relevance) were chosen as examples. Though it could not be shown using the test cases in this paper, the choice of edge weights, together with a function design, can impact the minimum cut.

A limitation when choosing the relevance edge weights is that there is currently no unified metric to model that relevance (and corresponding edge weight). For example, a high-frequency resonance in the power system related to the sampling (implementation-related relevance 24) could be numerically represented with worst-case negative damping, which could reflect in a specific edge weight. On the other hand, an active power control response related to a specific filter implementation (implementation-related relevance 20) would rather be represented with, e.g., a response time, which could then reflect in a specific edge weight. Though desirable, it is not obvious what a unified method for assigning relevance edge weights could look like.

Furthermore, it should be kept in mind that dealing with a function’s implementation aspects may be more challenging than dealing with a function’s parameterization for non-engineering reasons, e.g., intellectual property or trade secrets associated with implementation aspects. At the current research stage, the above aspects can only be reflected by manually choosing the relevance edge weights.

B. INTEGRATION EFFORT
The minimum cut corresponds to the minimum sum of edge weights between two graph parts. In this paper, these two parts correspond to “hardware relevance” and “system relevance”, resulting in a partitioning into an upper and a lower control level. The edge weight of the minimum cut can be interpreted as the dependencies between the two graph parts and, as a result, the integration effort of the resulting partitioning (one
vendor or integrator for system level, another vendor for hardware level) for the following three reasons:

Firstly, a different number of physical signal interfaces need to be provided as interface between lower and upper control parts, resulting in higher or lower integration effort.

Secondly, upper-level nodes with hardware relevance (case 1) must be well-understood when designing the lower control level and underlying hardware. In turn, the hardware design determines the design of the upper-level nodes with hardware relevance. Here, a distinction must be made between those nodes with parameterization-related hardware relevance and others with implementation-related hardware relevance. Node ⑨ for example, is designed with an implementation-related hardware relevance in test case 1 but a parameterization-related hardware relevance in test case 2. With implementation-related relevance, the specific implementation aspects (likely vendor-specific knowledge) may have to be disclosed. With parameterization-related relevance, the design is such that no vendor-specific implementation knowledge has to be disclosed, thus making the integration easier (and the total edge weight of the minimum cut smaller).

Thirdly, lower-level nodes with system relevance must be well-understood when designing the upper control level. In test case 1, nodes ②④⑤ have an implementation-related system relevance meaning that vendor-specific implementation aspects and hardware limits may have to be disclosed, thus leading to a high integration effort (and high total edge weight of the minimum cut). In turn, in test case 2, the fact that there is no node with implementation-related system relevance means that the integration effort is smaller compared to test case 1. This is reflected in a lower edge weight of the minimum cut.

A manual assignment of a node to the lower level (such as to avoid implementation-related relevances stretching from the upper control part to the hardware) is possible, as shown in case 3 while avoiding a function re-design. Here only the manual assignment of nodes to the lower level was investigated, but a node could equally well be manually assigned to the upper level. Other reasons for a manual assignment include a close connection to other functions, traditional placement in a certain control part, or intellectual property. Note, however, that the simple placement of nodes in a specific control part may not result in a decreased integration effort. In case 3 (based on case 1), three nodes are manually assigned to the lower level, resulting in a minimum cut of weight 546, indicating the increased integration effort (compared to case 2, weight 176). This is because, in case 3, lower-level functions with implementation-related system relevance have to be described to the upper-level control system to ensure proper design.

In summary, the main benefit of the proposed graph-theoretic method for control system partitioning is that the dependencies between the two control parts and the resulting integration effort can firstly be numerically assessed and secondly be minimized using the minimum cut, i.e., the sum of edge weights between two graph parts.

On another note, it can furthermore be argued that a partitioning with decreased dependencies between the upper and lower control level may allow the development of a more optimized hardware design because decreased dependencies could allow for reduced safety margins that otherwise account for potentially unknown and complex dependencies to the behavior of the upper-level control part.

C. PARTITIONING CRITERIA

This paper investigated the functional partitioning into “system relevance” and “hardware relevance”. However, other functional partitioning aspects, e.g., sub-partitioning on the system level could also be investigated. Non-functional options for partitioning include, e.g., the testing effort or partitioning into functions running on a digital signal processor (DSP) or a field-programmable gate array (FPGA). Vice-versa, the computing capacity or speed capability of the respective DSPs or FPGAs in an MMC control system may require manually assigning a control function to the upper level (likely implemented in a DSP) or lower level (likely implemented in an FPGA). Also, the partitioning method could be applied with additional protection functions and/or consider signal bandwidths. For example, a high signal bandwidth between two functions could be represented with a high edge weight, resulting in these two functions being so closely connected that they can be treated as a node sub-group. On this note, a manual sub-grouping of nodes, using an artificially high edge weight, may be done if two (or several) control functions are traditionally implemented together, or their separation with a communication interface would affect control performance.

Finally, the graph partitioning method could also be adapted to address the coordinated design of multiple MMCs, such as back-to-back MMC systems. For such applications, additional control layers are needed, in turn requiring a different method of control system partitioning [17], e.g., using several steps.

D. COMPARISON WITH STANDARDIZED PARTITIONING

When comparing the graph-based control partitioning with standardized approaches for control partitioning [2]–[4] or published control architectures [8], [9] it must be kept in mind that these partitioning solutions operate on generalized (standardized) and as a result simplified control systems. Still, it can be observed that the established partitioning solutions tend toward interfacing at the arm reference voltage with the circulating current control in the lower control level. However, suppose the relevance of control functions on the system and/or hardware is considered (as done in this paper). In that case, the partitioning is optimal in other locations that is not the arm reference voltage. This may result in more physical interfaces (e.g., six in case 2 compared to two in case 4) but fewer dependencies and less integration effort between upper and lower control levels (e.g., low minimum cut weight of 176 case 2 compared to 1412 in case 4).
E. HANDLING SENSITIVE INFORMATION

As regards multi-vendor setups, note that the proposed partitioning method operates on information whether or not there is an implementation-related or parameterization-related relevance of a function for hardware and/or system. The specifics of a function do not have to be revealed. As such, the method allows discussions between different parties (vendors, TSOs, research institutes) in an early project stage when detailed information about control functions is not yet relevant or cannot be revealed.

Furthermore, the proposed partitioning method gives means to understand how control design and the need for information disclosure interact. This is relevant not only in multi-vendor setups but even more so in suggested approaches using open-source upper-level MMC control [10] that would allow collaboration and re-configuration on the upper system level while keeping vendor-specific implementation aspects on the lower control level closed.

F. FUTURE WORK: SYSTEMS ENGINEERING, FUNCTIONAL SPECIFICATIONS, AND GRID CODES

In the future, it can be expected that the widespread application of complex HVDC systems will require a holistic design approach using the *Systems Engineering* methodology [25]. With this methodology, functional specifications will be defined for MMC control and protection based on requirements allowing a robust system design [26]. In a simplified view, the *Systems Engineering* methodology uses three steps:

The first step is the concept definition of the MMC control and protection system with the identification of neighboring systems and the basic operational conditions.

The second step is the system definition, where different specific MMC control and protection systems are evaluated regarding pre-defined requirements and the overall system goal. Here, the partitioning method proposed in this paper can be used to complement this evaluation, and the identification of the most suitable control system design, including the identification of physical signals and required information exchange between different vendors.

The third step is the component definition, where — based on the partitioning result in the system definition — the components’ specifications can be developed.

As a result, the proposed partitioning method is expected to complement the *Systems Engineering* methodology well because it contributes to an efficient concept, system, and component definition. On a related topic, the proposed partitioning method could also contribute to a better understanding of grid code definitions and how these translate into (groups of) specific control functions.

In summary, the application of the partitioning method is expected to be beneficial for studies and the efficient integration of MMC control and protection systems, especially in multi-vendor HVDC systems.

VIII. CONCLUSION

This paper presented a graph-theoretic method for partitioning MMC control systems into an upper and lower level, where the upper level has mainly system relevance, and the lower level has mainly hardware relevance. The partitioning is subject to minimized dependencies between the two control parts (minimum graph cut), and as a result, a minimized integration effort. The design of specific control functions has an impact on the partitioning. Suppose the design goal is to have no functions with hardware relevance in the upper level. In that case, it makes sense to design these functions with less system relevance, so they are more easily pulled to the lower level during partitioning. Furthermore, even though the manual assignment of specific functions to the lower level is possible, designing functions with less system relevance is desirable, such as minimizing the cut’s complexity and reducing the integration effort.

APPENDIX

The names and numbering of the control nodes is provided in Table 2.

| Node names                                      |
|------------------------------------------------|
| 1     Integrated AC/HVDC system control        |
| 2     Pole control                             |
| 3     Q mode                                   |
| 4     Voltage control                          |
| 5     $Q$ vs. $U$ curve                       |
| 6     $Q$ setpoint                             |
| 7     Ramped $Q$ setpoint                      |
| 8     Reactive power modulation                |
| 9     $I_Q$ calculation                        |
| 10    $P$ mode                                 |
| 11    $P$ setpoint                             |
| 12    Ramped $P$ setpoint                      |
| 13    Frequency sensitive mode                 |
| 14    $P$ vs. $f$ curve                       |
| 15    Active power modulation and stability functions |
| 16    $I_P$ calculation                        |
| 17    $U_{DC}$ control                        |
| 18    $U_{DC, ref} = U_{DC2, ref} + fref$      |
| 19    $I_Q$ modulation and $I_P/I_Q$ priority and capability limitation |
| 20    Mode 1: Active power control             |
| 21    Mode 2: DC voltage and/or total energy control |
| 22    Mode 3: $V/f$ control                    |
| 23    Current limitations                      |
| 24    Current control                          |
| 25    AC voltage limitations                   |
| 26    Zero phase sequence control              |
| 27    3rd harmonic injection                   |
| 28    Mode 1: DC current control or limitation |
| 29    Mode 2: DC voltage control or feedforward|
| 30    Mode 3: Total energy control             |
| 31    Mode 4: $I_d$ control for DC faults      |
| 32    DC voltage limitations                   |
| 33    Frequency selective filters              |
| 34    Weighting function                       |
| 35    Energy balancing with $P$ or $I +$ feedforwards |
| 36    Current limitations                      |
| 37    Circulating current control              |
| 38    Voltage order calculation                |
| 39    Voltage order limitation                 |

General meaning, do not confuse with power system.
ACKNOWLEDGMENT

The authors are grateful to CIGRE working group B4.85 led by Staffan Norrga for providing feedback to this work.

REFERENCES

[1] K. Ji et al., “A hierarchical small-signal controller stability analysis method for the MMCs,” IEEE Trans. Power Del., vol. 37, no. 4, pp. 2587–2598, Aug. 2022.

[2] Guide for the development of models for HVDC converters in a HVDC grid, CIGRE Technical Brochure 604 (Working Group B4.57), Dec. 2014.

[3] HVDC Grid Systems and Connected Converter Stations – Guideline and Parameter Lists for Functional Specifications, Part 1, CENELEC Standard 50654-1:2020, 2020.

[4] IEEE Guide for Control Architecture for High Power Electronics (1 MW and Greater) Used in Electric Power Transmission and Distribution Systems, IEEE Standard 1676-2010, IEEE, New York, NY, USA, 2010.

[5] K. Sharifabadi et al., “Parallel operation of multivendor VSC-HVDC schemes feeding a large islanded offshore oil and gas grid,” in Proc. CIGRE Session, Paris, 2018, pp. 1–13.

[6] S. Dennetière et al., “Technical solutions to predict and mitigate inadvertent interaction of two parallel connected VSC-HVDC schemes feeding an islanded offshore oil and gas grid,” in Proc. CIGRE Session, Paris, 2018, pp. 1–16.

[7] ABB, “Zhangbei – the world’s first DC-grid with HVDC Light,” 2020, Accessed: 8 Jun., 2022. [Online]. Available: https://new.abb.com/hvdc/references/zhangbei

[8] G. Bathurst and P. Bordignan, “Delivery of the nan’ao multi-terminal VSC-HVDC system,” in Proc. 11th IET ACDC 2015, Birmingham, 2015, pp. 1–6.

[9] H. Rao, “Architecture of Nan’ao Multi-terminal VSC–HVDC system and its multi-functional control,” CSEE J. Power Energy Syst., vol. 1, no. 1, pp. 9–18, Mar. 2015.

[10] I. Jahn et al., “An architecture for a multi-vendor VSC-HVDC station with partially open control and protection,” IEEE Access, vol. 10, pp. 13555–13569, 2022.

[11] K. Sharifabadi, L. Harnefors, H.-P. Nee, S. Norrga, and R. Teodorescu, Design, Control, and Application of Modular Multilevel Converters for HVDC Transmission Systems, 1st ed. Chichester, U.K.: Wiley-IEEE Press, 2016.

[12] L. R. Ford and D. R. Fulkerson, “Maximal flow through a network,” Can. J. Math., vol. 8, pp. 399–404, 1956.

[13] R. J. Wilson, “Matching, marriage and Menger’s theorem,” in Introduction to Graph Theory, 4th ed. Edinburgh Gate, Harlow, U.K.: Addison Wesley Longman Ltd., 1998, ch. 8, pp. 112–130.

[14] Y. Dinitz, “Dinitz’ algorithm: The original version and even’s version,” in Theoretical Computer Science: Essays in Memory of Shimon Even, O. Goldreich, A. L. Rosenberg, and A. L. Selman, Eds., Berlin Heidelberg, Germany: Springer, 2006, pp. 218–240.

[15] M. Stor and F. Wagner, “A simple min-cut algorithm,” J. ACM, vol. 44, no. 4, pp. 585–591, Jul. 1997.

[16] igraph, “igraph – The network analysis package,” 2022, Accessed: Jun. 20, 2022. [Online]. Available: https://igraph.org

[17] S. E. Schaeffer, “Graph clustering,” Comput. Sci. Rev., vol. 1, no. 1, pp. 27–64, Aug. 2007.

[18] K. Smedley and S. Cuk, “Switching flow-graph nonlinear modeling technique,” IEEE Trans. Power Electron., vol. 9, no. 4, pp. 405–413, Jul. 1994.

[19] Y. Ma and K. Ma Smedley, “Switching flow-graph nonlinear modeling method for multistate-switching converters,” IEEE Trans. Power Electron., vol. 12, no. 5, pp. 854–861, Sep. 1997.

[20] H. Behjati, A. Davoudi, and F. Lewis, “Modular DC–DC converters on graphs: Cooperative control,” IEEE Trans. Power Electron., vol. 29, no. 12, pp. 6725–6741, Dec. 2014.

[21] G. Zhang, Y. Liao, S. S. Yu, and Y. Zhang, “A graph-modeling approach to topology simplification in power converters,” IEEE Trans. Power Electron., vol. 37, no. 7, pp. 8248–8261, Jul. 2022.

[22] L. Mo, G. Chen, J. Jiang, X. Qing, Y. Hu, and X. He, “Graph theory-based programmable topology derivation of multtiport DC–DC converters with reduced switches,” IEEE Trans. Ind. Electron., vol. 69, no. 6, pp. 5745–5755, Jun. 2022.

[23] Y. Zhou, Y. Jin, Y. Chen, H. Luo, W. Li, and X. He, “Graph-model-based generative layout optimization for heterogeneous SiC multichip power modules with reduced and balanced parasitic inductance,” IEEE Trans. Power Electron., vol. 37, no. 8, pp. 9298–9313, Aug. 2022.

[24] C. Hirsching et al., “Passivity-based sensitivity analysis of the inner current controller in grid-following MMC-HVDC applications – an overview,” in Proc. 46th IEEE IECON, Singapore, 2020, pp. 1412–1417.

[25] M. Hoffmann, “Model-based systems engineering for offshore HVDC grid integration systems,” Ph.D. dissertation, TU Braunschweig, Braunschweig, Germany, 2021.

[26] M. Hoffmann, C. Wei, and M. Kurrat, “Requirement analysis with SysML for concept design of offshore wind farm grid connection,” in Proc. VDE ETG Congr, 2021, pp. 808–813.