



**HAL**  
open science

## Supervisory control of post-fault restoration schemes in reconfigurable HVDC grids

Lucas Molina-Barros, Miguel Romero-Rodriguez, Laurent Pietrac, Emil Dumitrescu

► **To cite this version:**

Lucas Molina-Barros, Miguel Romero-Rodriguez, Laurent Pietrac, Emil Dumitrescu. Supervisory control of post-fault restoration schemes in reconfigurable HVDC grids. EPE'21 ECCE Europe, Sep 2021, Ghent, Belgium. 10.23919/EPE21ECCEurope50061.2021.9570641 . hal-03621801

**HAL Id: hal-03621801**

**<https://hal.science/hal-03621801v1>**

Submitted on 28 Mar 2022

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Supervisory control of post-fault restoration schemes in reconfigurable HVDC grids

Lucas MOLINA-BARROS<sup>1,2</sup>, Miguel ROMERO-RODRIGUEZ<sup>1</sup>, Laurent PIETRAC<sup>3</sup>  
Emil DUMITRESCU<sup>1,2</sup>,

1. Supergrid Institute SAS, 23 rue de Cyprian, 69100 Villeurbanne, France

2. Univ Lyon, INSA Lyon, Université Claude Bernard Lyon 1, Ecole Centrale de Lyon,  
CNRS, Ampère, UMR5505, 69621 Villeurbanne, France

3. Université Clermont Auvergne, CNRS, Clermont Auvergne INP, Institut Pascal,  
F-63000 Clermont-Ferrand, France

Email: lucas.molinabarro@supergrid-institute.com

## Keywords

«Control methods for electrical systems», «Design», «Fault handling strategy», «Modelling», «Multi-terminal HVDC», «Smart grids», «Systems engineering».

## Abstract

This paper studies the use of Supervisory Control Theory to design and implement post-fault restoration schemes in a HVDC grid. Our study focuses on the synthesis of discrete controllers and on the management of variable control rules during the execution of the protection strategy. The resulting supervisory control system can be proven “free of deadlocks” in the sense that designated tasks are always completed.

## Introduction

The industrial standard to design the control of process plants and machinery is to directly translate a set of informal specifications into a graphical or textual programming language, (e.g., Functional Block Diagrams, Ladder, C code for micro-controllers, etc.), which are then implemented in the target system. The term “informal” refers to everything that is not based on a strictly composed, syntactically and semantically well-defined form [1]. This work is usually assigned to system engineers and technicians, who must know how to properly use field inputs and outputs in the definition of control rules that make the system compliant with the expected behavior. In this process, they are also required to anticipate system dynamics that could create deadlocks, in which a set of conflicting instructions prevent the system under control to achieve its designated tasks. The control of power systems components, such as power plants and high voltage substations, involves similar practices in its design. Because of their logical complexity, size and value, this creates a significant level of difficulty in the definition of control rules and additional risks for a stable operation.

Future HVDC grids are expected to play a key role in power delivery [2]. They will comprise a large amount of sensors and actuators, distributed over large distances, and will involve the control of several components with fast dynamics. These circumstances call for advanced control methods to ensure the integrity and reliability of future HVDC systems and adjacent AC grids. This is particularly important under safety-critical situations, such as the implementation of automatic post-fault restoration schemes. Most of the available literature regarding the protection of HVDC grids focus on the descriptions of fault detection [3] and protection strategies [4], providing extensive data about the electrical behavior of grid components during the fault recovery. However, little information is given on the development of the logical rules used in these studies and their guarantees regarding correctness and deadlocks, with automatic control sequences usually presented through generic flowcharts. In this paper, we intend to

fill this gap by proposing the use of a formal approach to create a correct-by-construction supervisory control system for a multi-terminal HVDC grid. In our case study, the supervisory control implements a post-fault restoration scheme. Following the discrimination and isolation of a faulty transmission link, it monitors and coordinates the behavior of the grid components so as to safely restore the DC voltage and power flow under the new circumstances. The supervisory control also enables the possibility of reconnecting the faulty link to the grid (assuming it has been correctly fixed), automatically adjusting the power flow and bringing the system back to its nominal behavior.

Our method is based on the application of the Supervisory Control Theory (SCT) [5], which is a formal control approach based on mathematically provable properties. Therefore, it provides correction guarantees to meet the requirements expected from safety-critical control applications [1], such as the ones found in automated power systems. In recent years, examples of its use can be found in AC distribution systems, such as the implementation of power management strategies in microgrids [6] and the control of custom power parks [7]. In HVDC systems, the SCT has been used to control an on-load tap-changing transformer [8] and the start-up procedure of a point-to-point HVDC link [9]. The new application of SCT presented in this paper brings a reconfigurable supervisory control system [10], based on the commutation of separately computed discrete controllers that belong to what we call *elementary control modes*. In our case, the reconfiguration of the control system is provoked by the loss of an asset (e.g. an HVDC cable) after an electrical fault. To the best of our knowledge, this kind of situation has not yet been addressed by the research involving the application of SCT in power systems. The obtained supervisory control follows the evolution of the system's topology and its behavior, rapidly adjusting the control rules applied on the grid's equipment. A simulation with EMTP-RV is performed to demonstrate the action of the obtained supervisory control.

## Presentation of the case study

The HVDC network shown in Fig. 1 is used to illustrate the reconfiguration of supervisory control. It consists of a four-terminal grid in a bipolar configuration [11]. Therefore, each AC/DC station is equipped with a pair of Modular Multi-level Converter (MMC)[12], one for each pole of the DC grid, and breaking modules designed for an HVDC system. For the sake of simplicity, only the positive pole of the bipolar configuration is shown. Local control units (LCU), with fast monitoring and reaction capabilities, are situated at each AC/DC station. A central control unit (CCU), with slower monitoring and reaction capabilities, exchanges information with all local controllers across the grid to coordinate their control sequences. The CCU is assumed to be in a separate environment, such as the control room of the system operator that oversees this grid. The supervisory control system of the positive pole of this grid has an identical structure and is independent of the one designed for the negative pole.

We define for this grid four configurations (system topologies) according to the connection status of the DC cables that connect the AC/DC stations (see legend in Fig. 1). The network is assumed to be initially in nominal operation (configuration **T1**), with rated voltage values at all stations and power flowing through all DC cables. We consider that at any time a pole-to-ground fault can occur on one of the DC cables. The post-fault restoration studied in this paper is based on the primary sequence of the non-selective protection strategy presented in [4]. The first step is the suppression of the AC fault current contribution, accomplished by DC converter breakers (CBC) which are tripped right after the fault detection, immediately followed by the blocking of MMC converters. Next, a specialized fault identification algorithm is able to discriminate and isolate the faulty cable by opening the line DC breakers (CBL) at its ends. After the faulty line is isolated, the converter breakers re-close and the grid voltage restoration begins under coordination of the supervisory control system. During the time the MMC stations are isolated from the grid, they are deblocked and used as STATCOM devices to sustain the AC voltage by injecting reactive power into the AC grid. Assuming the faulty cable is repaired, the supervisory control monitors and coordinates the required actions for its reconnection.

Due to a space limitation, the development of our control models is limited to the reconfiguration caused by the loss of the cable **BD**, which corresponds to the commutation between configurations **T1** and **T2**. Among the existing possibilities, this one is the most complex from a control point of view as it generates

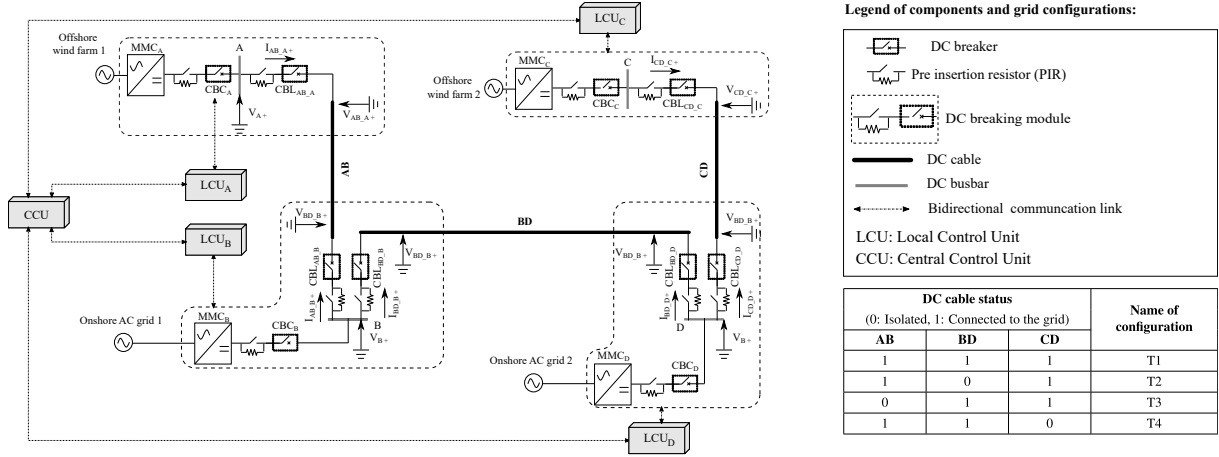


Fig. 1: Reconfigurable HVDC grid.

two point-to-point HVDC links that behave independently. Nevertheless, the method that is presented can be continued to take into account the isolation of the other DC cables in the network.

## Supervisory control of HVDC grids

### Introduction to the theoretical framework

In this paper, a design method relying on Supervisory Control Theory (SCT) [5] is presented. It studies the control of discrete event systems (DES) and relies on language theory to enforce behavioral requirements. Formally, the behavior of a DES is represented by a deterministic automaton  $G = (Q, \delta, \Sigma, q_0, Q_m)$ , as illustrated in Fig. 2a.  $Q$  is a finite set of states (circles), with  $q_0 \in Q$  as the initial state and  $Q_m \subseteq Q$  being the marked states (double circle). Marked states are used to designate desirable states or the completion of a task.  $\Sigma$  is a finite set of events ( $\sigma_i$ ), also known as alphabet, and is divided into two disjoint subsets:  $\Sigma_c$  and  $\Sigma_{uc}$ , respectively named controllable and uncontrollable event sets. Finally  $\delta: Q \times \Sigma \rightarrow Q$  is a transition mapping which gives the next state after occurrence of an event. In the figures of this paper, transitions associated with controllable events are represented as crossed arrows.

The SCT aims to determine an event-triggered control model called *supervisor* based on models of an uncontrolled process  $G$  and a user defined specification  $H$ . Following the feedback principle used in control theory, the action of a supervisor  $S$  on a given process  $G$  can be represented by the diagram in Fig. 2b. The supervisor  $S$  receives the stream of events  $s$  generated by  $G$  and reacts by disabling certain events so that its behavior complies with that determined by  $H$ . This implies that  $S(s) \subseteq \Sigma$ ,  $S(s)$  being the set of events enabled by the supervisor. The behavior of the controlled system, i.e.  $G$  under control of  $S$  (also designated  $S/G$ ), and the control rules of the supervisor can be graphically represented by an automaton  $S$ . It can be then translated as a conventional programming language and implemented into the target system as a control block containing inputs and outputs. This requires a physical interpretation of the events modelled in  $G$ . Typically, uncontrollable events are associated with the block inputs (e.g., crossing of a pre-defined threshold by a field measurement) whereas controllable events are associated with the modification of the block outputs (e.g., sending an order to close a circuit breaker), which can be connected to the actuators of the operative unit.

A supervisor can be computed by an algorithmic procedure that takes as input a couple of automata corresponding to  $G$  and  $H$ . Given the complexity of most systems, it is easier to define  $G$  and  $H$ , from the composition of sub-systems (or components) models ( $G_i$ ) and independent specifications ( $H_j$ ) respectively. The mathematical operation of parallel composition, represented by the operator “ $\parallel$ ” and formally defined in [13], allows to obtain all the independent behaviors when the automata do not share events and to synchronize them if they do. Therefore, a way to get a supervisor for a plant  $G$  so that it meets all specifications from  $H$  would be to calculate  $G \parallel H$ . However, the existence of a valid controller

is not guaranteed. When a solution is possible, the synthesis algorithm returns by default a supervisor that respects the properties of *controllability: the supervisor does not directly disable any occurrence of uncontrollable event*, and *nonblocking: marked states can always be reached from the system's initial state*. This last property means that designated tasks can always be terminated, which formally proves that the supervisory control is free of deadlocks.

One may also design multiple supervisors that enforce different specifications on the same plant. This is known as decentralized supervisory control and is represented in Fig. 2c. “p” is called “natural projection”, and represents the removal of events from the stream  $s$ , hence the partial observation of the plant behavior. The set of events enabled by individual supervisors ( $S_1(s), S_2(s)$ ) are based on their own observations. The intersection of these sets ( $S_{dec}(s)$ ) represents the final control rules applied on the system under control ( $G$ ). In SCT, there are different tools to verify if the interaction of decentralized controllers is nonblocking. In the following, the open source C++ library LibFAUDES (Friedrich-Alexander University Discrete Event Systems)[14] is used for the modeling of automata, synthesis of supervisors and verification of control properties.

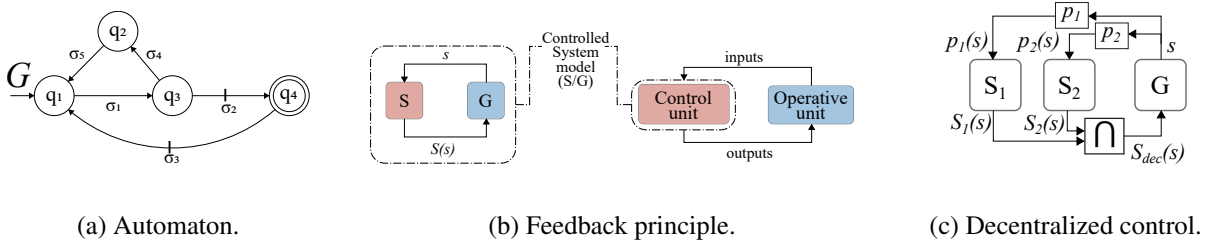


Fig. 2: Design assumptions in Supervisory Control Theory.

## Resolution method

The development of our control solution is based on different stages, shown in Fig. 3. The functional analysis of the system under control (Step 1 in Fig. 3) involves the textual description of the expected automated behavior, task done in the previous section, and requires the identification of grid components that are to be monitored or controlled by the supervisory control. In our case, the list of grid components contains the MMC converters ( $G_{MMC}$ ), line and converter DC circuit breakers ( $G_{CBC}$  and  $G_{CBL}$ ) and DC cables ( $G_{cable(V)}$  and  $G_{cable(I)}$ ), whose current and voltage levels are monitored at each extremity. These components are modelled as partial automata, shown in Fig. 4a. Initial and marked states are determined in step 2 (Fig. 3) as they may change according to the scenario being studied.

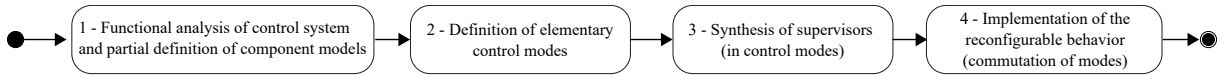


Fig. 3: Flowchart of the resolution method.

MMC converters can work in different control modes that change the way AC power and DC voltage values are locally regulated by it [15]. In our design, we use controllable events to represent the update of reference values and control parameters (droop constants, ramp limitation, time constants, etc.) in a MMC depending on its role and its position in the grid. The control settings associated with the events used in the reconfiguration between grid topologies **T1** and **T2** are shown in Table I. They are associated with self-loop transitions at state **Nominal** of  $G_{MMC}$  (Fig. 4a).

The models presented so far are related to physical components of the grid. In our design, we also represent logical components and interactions in the control system, as illustrated in Fig. 4b. A timer block is used to confirm when certain transient behaviors (e.g., fault clearing and charging of DC bus) are over. The timer is set (`Timer_start`) by the supervisory control when stable conditions are detected and reset (`Timer_reset`) otherwise (e.g., current and voltage oscillations). The timer output becomes true (`Timer_timeout`) if it is not reset during a pre-determined time (e.g., 10 ms), which is used to confirm the

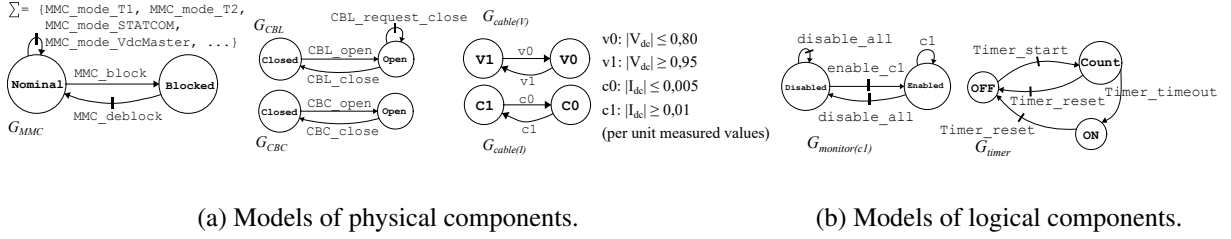


Fig. 4: Partial models of system components.

Table I: Reconfiguration events of MMC converters and associated parameters.

MMC Station	MMC_mode_T1	MMC_mode_T2	MMC_mode_VdcMaster	MMC_mode_STATCOM
A	Vdc droop: (320 kV, 250 MW, -50 MVAR)	Pdc constant: (250 MW, -50 MVAR)	Vdc constant: (320 kV, -50 MVAR)	Pdc constant: (0 MW, -50 MVAR)
B	Vdc droop: (320 kV, -325 MW, 50 MVAR)	Vdc constant: (320 kV, 50 MVAR)	Vdc constant: (320 kV, 50 MVAR)	Pdc constant: (0 MW, 50 MVAR)
C	Vdc droop: (320 kV, 400 MW, -70 MVAR)	Pdc constant: (250 MW, -70 MVAR)	Vdc constant: (320 kV, -70 MVAR)	Pdc constant: (0 MW, -70 MVAR)
D	Vdc droop: (320 kV, -325 MW, -100 MVAR)	Vdc constant: (320 kV, -100 MVAR)	Vdc constant: (320 kV, -100 MVAR)	Pdc constant: (0 MW, -100 MVAR)

end of a transient behavior. Additionally, we use a logical model to decide when a certain uncontrollable event should be monitored by the supervisory control system. For instance, the uncontrollable event  $c1$  can only be generated if  $G_{monitor(c1)}$  is at state **Enabled**. The execution of event `disable_all` stops the monitoring of all uncontrollable events of the same station.

Based on the results of the functional analysis and the list of automata that were defined for the system, the grid's behavior is decomposed into several elementary control modes (step 2 in Fig. 3). Each mode is associated with a set of physical and logical components that are controlled together. The commutation between grid configurations **T1** and **T2** involves a total of four elementary control modes as illustrated by the state-diagram in Fig. 5. The table in this figure indicates for these modes the role of the supervisory control and the components involved in the completion of designated tasks.

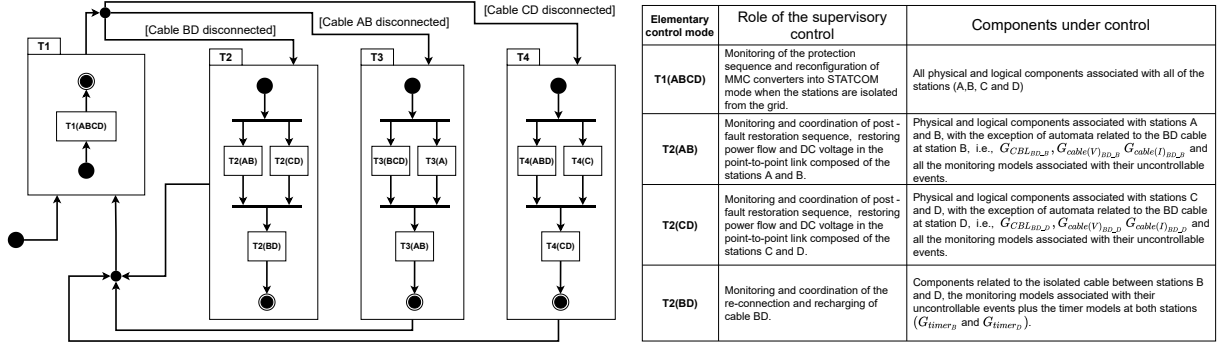


Fig. 5: Life cycle of elementary control modes

Initially, each control mode is studied as an independent supervisory control system composed of decentralized supervisors, as shown in Fig. 2c. The synthesis of controllers (step 3 of Fig. 3) follows a framework known as *hierarchical and decentralized control* [15], illustrated in Fig. 6. The supervisory control system is divided into two hierarchical levels. At the lowest level (LCUs), we have a direct control and monitoring of system components by local supervisors ( $S_1, S_2, \dots, S_n$ ). At the highest level (CCU), a "grid supervisor" ( $S^{hi}$ ) is used for the coordination of local ones. The synthesis starts at the lowest level using local models of components (both logical and physical) and control specifications. The behavior of each local controlled system ( $S_n/G_n$ ), graphically represented by automaton  $S_n$ , is projected to a smaller alphabet. This yields a new automaton with reduced size, which becomes a plant component at the higher level. Using the parallel composition of projected local supervisors (from different stations) and a new set of high-level specifications, a new high-level grid supervisor is computed. The coordination of local controllers is achieved through the intersection of the set of events enabled by local

( $S_n(s_n)$ ) and high-level ( $S^{hi}(s^{hi})$ ) supervisors. The supervisory control is nonblocking if all projections are *natural observers*. Because information is lost in the projection of an automaton, the property of natural observer (formally explained in [15]) indicates if the result contains sufficient data on the behavior of controlled low-level systems so that they can still be correctly coordinated by a high-level supervisor. These steps are illustrated in the computation of the high-level supervisor in mode **T2(AB)**, where a central coordination is required.

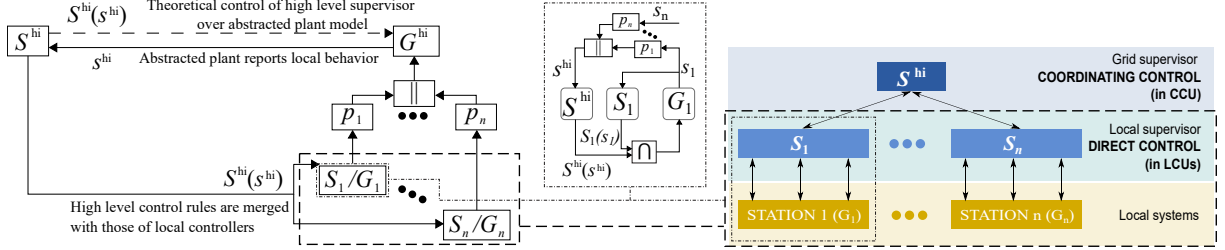


Fig. 6: Hierarchical and decentralized control architecture for one control mode.

We illustrate the synthesis of a low-level supervisor with  $S_A^{T1(ABCD)}$ , in the elementary control mode **T1(ABCD)**. Fig. 7 shows the initial state and set of marked states for the local components used in the synthesis algorithm as well as the individual control specifications of this mode ( $H_{A(i)}^{T1(ABCD)}$ ,  $i \in 1, \dots, 7$ ). Each state is associated with a certain control task determined by the color pattern. A set of logical events  $\Sigma_{logical_A} = \{\text{Fault\_Ack\_A}, \text{Fault\_Cleared\_A}, \text{Protected\_A}\}$  is used to simplify the design of models. They belong to a new local logical component  $G_{logical_A}^{T1(ABCD)}$ , added to station A for the study of elementary control mode **T1(ABCD)**. They are controllable because they can be directly disabled by the supervisory control. For instance, thanks to the synchronization of  $H_{A(1)}^{T1(ABCD)}$ ,  $H_{A(2)}^{T1(ABCD)}$  and  $H_{A(3)}^{T1(ABCD)}$  the acknowledgment of the fault by the local supervisor at station A (event  $\text{Fault\_Ack\_A}$ ) can only occur if the blocking of the MMC ( $\text{MMC\_Block\_A}$ ) and opening of the CDC ( $\text{CDC\_Open\_A}$ ) have been detected before.

In the specification models, self-loops transitions associated with uncontrollable events from physical components (e.g. states 1, 2, 5 and 6 of  $H_{A(1)}^{T1(ABCD)}$ ) indicate *when* and *which* sensor data should be monitored. Because the synthesis algorithm computes a maximally permissive solution, it is possible to obtain as result an automaton where two transitions associated with controllable events are possible. To enforce an *automatic* transition, a unique choice of controllable event must be possible. In our case, we choose to model a new local specification  $H_{A(7)}^{T1(ABCD)}$  that defines a unique order for the generation of controllable events in the final supervisor. Other alternative models are possible as long as a non-empty result is returned by the synthesis algorithm.

From  $G_A^{T1(ABCD)} = G_{MMC_A} \parallel G_{CBL_A} \parallel \dots$  (composition of local components) and  $H_A^{T1(ABCD)} = H_{A(1)}^{T1(ABCD)} \parallel \dots \parallel H_{A(7)}^{T1(ABCD)}$ , we compute  $S_A^{T1(ABCD)}$ , which contains 45 states and 88 transitions. The local supervisor  $S_C^{T1(ABCD)}$  (station C), has an identical structure whereas  $S_B^{T1(ABCD)}$  and  $S_D^{T1(ABCD)}$  have 127 states and 469 transitions. The higher size is justified by the additional cables and DC breakers (CDLs) that are monitored at these stations. The control tasks for the elementary control mode **T1(ABCD)** are all handled locally, without coordination from a central controller. Because local supervisors were computed through a synthesis algorithm and do not share any events, this elementary control mode is nonblocking.

Elementary control modes **T2(AB)**, **T2(CD)** and **T2(BD)** require a set of local supervisors, one at each station, and a higher level controller to coordinate them. Because of the grid's symmetry, the structure of controllers associated with **T2(AB)** is identical to those belonging to **T2(CD)**. Supervisors  $S_A^{T2(AB)}$  and  $S_C^{T2(CD)}$  both contain 33 states and 68 transitions whereas  $S_B^{T2(AB)}$  and  $S_D^{T2(CD)}$  both contain 34 states and 69 transitions. Their structure can be divided into two phases, one for the grid's charging and another for the power restoration. A simplified version of their graphical representation, where part of the transitions are omitted, is shown in Fig. 8.

The set of logical events  $\Sigma_{logical_A}^{T2(AB)} = \{\text{Charged\_A}, \text{Charged\_Grid\_AB}, \text{Power\_Restored\_A}, \text{Power\_Restored\_Gr-}$

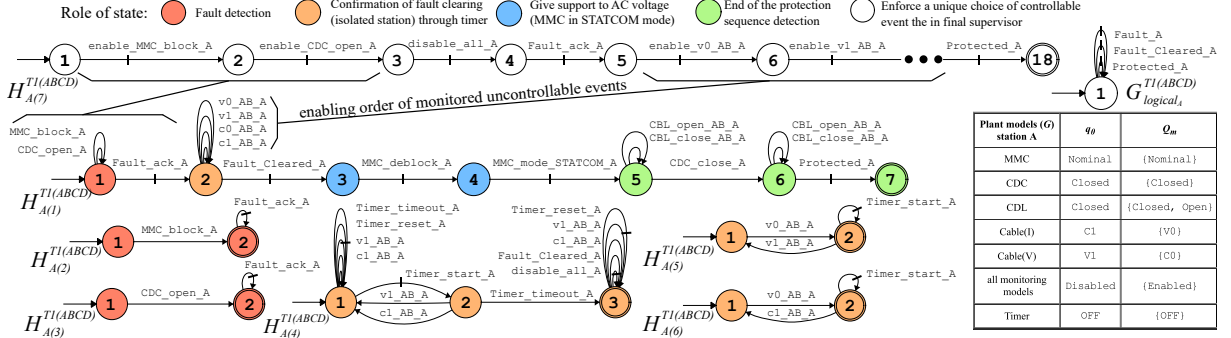


Fig. 7: Specification models for local supervisor  $S_A^{T1(ABCD)}$ .

id} and  $\Sigma_{logical_B}^{T2(AB)} = \{\text{Charged}_B, \text{Charged\_Grid\_AB}, \text{Power\_Restored}_B, \text{Power\_Restored\_Grid\_AB}\}$  are used in the models of control specifications (not shown for lack of space) of stations A and B respectively. We use the event set  $\Sigma_{T2(AB)}^{hi} = \Sigma_{logical_A} \cup \Sigma_{logical_B}$  as destination domain for the projection of local supervisors to a higher control level. The operation of *natural projection*, available in libFAUDES, is used for this purpose. It returns new smaller automata, corresponding to the projected version of local supervisors  $S_A^{T2(AB)}$  (station A) and  $S_B^{T2(AB)}$  (station B). The choice of event set used for this projection is not made randomly. At minimum, it must contain events shared with other stations and other local events that we want to use in a higher level specification. The grid supervisor  $S_{T2(AB)}^{hi}$  is computed through synthesis using the high-level plant model  $G^{hi} = p(S_A^{T2(AB)}) \parallel p(S_B^{T2(AB)})$  and the set of high-level specifications  $H^{hi} = H_{T2(AB)}^{hi(1)} \parallel \dots \parallel H_{T2(AB)}^{hi(4)}$ , all represented in Fig. 8. It can be seen that the high-level supervisor only authorizes the generation of events Charged\_Grid\_AB and Power\_Restored\_Grid\_AB after all stations locally confirm being charged (Charged\_A, Charged\_B) and power flow through them (Power\_Restored\_A, Power\_Restored\_B). Using libFAUDES, we verify that that the projection functions used to abstract local supervisors contain the property of *natural observer*, which proves that the interaction of  $S_A^{T2(AB)}$ ,  $S_B^{T2(AB)}$  and  $S_{T2(AB)}^{hi}$  is nonblocking.

The computation of supervisors for **T2(BD)** and other elementary control modes follows the same guidelines presented so far.  $S_B^{T2(BD)}$  and  $S_D^{T2(BD)}$  both contain 64 states and 137 transitions, whereas  $S_{T2(BD)}^{hi}$  only has 6 states and 7 transitions. These supervisors are used to control and monitor the reconnection and recharging of the BD cable. Grid configurations **T3** and **T4** follow the isolation of a faulty cable that creates an isolated station through which power cannot flow. Therefore, they contain each an elementary control mode for isolated stations, i.e., **T3(A)** and **T4(C)**, in which the supervisory control coordinates and monitors the recharging procedure of the local DC bus to its nominal value. Because only one station is involved in this process, the coordination of a high-level supervisor is not required.

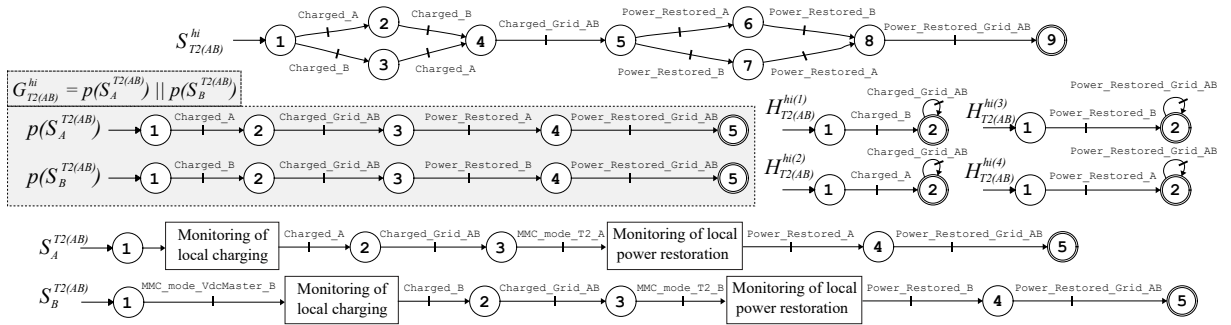


Fig. 8: High-level supervisor  $S_{T2(AB)}^{hi}$  and elements used for its synthesis.

The commutation of control modes is studied next (step 4 in Fig. 3). We wish to enforce the reconfiguration behavior and parallelism of modes shown in Fig. 5 with the additional specification that all



MMC converters be reconfigured to the ideal settings for the initial grid topology (`MMC_mode_T1`) once the grid configuration **T1** is re-activated. Achieving these tasks is possible thanks to a procedural modification and combination of the supervisors obtained previously. We start by adding new logical events that indicate the activation and deactivation of grid configurations:  $\Sigma_{reconfiguration} = \{T1\_start, T1\_end, T2\_start, T2\_end, T3\_start, T3\_end, T4\_start, T4\_end\}$ . Knowing that all supervisors can be represented as an automaton containing a unique initial state ( $q_0$ ) and only one marked state ( $Q_m = \{q_m\}$ ), local supervisors of the same station can be merged into a single low-level controller following the reconfiguration behavior indicated in Fig. 5. Fig. 9a shows the example for station B, where a single unique level supervisor  $S_B$  is obtained from the combination of other automata and the addition of new states (**S1**, **S2** and **S3**) and transitions. It can be seen that the control sequence from the mode **T2(BD)**, where the re-closing  $CDL_B$  is authorized, can only happen after the complete execution of **T2(AB)**. Also, as soon as the event `T1_start` is generated, the local MMC is reconfigured to its ideal settings for configuration **T1** (event `MMC_mode_T1_B`). A similar modification is carried out for all other stations.

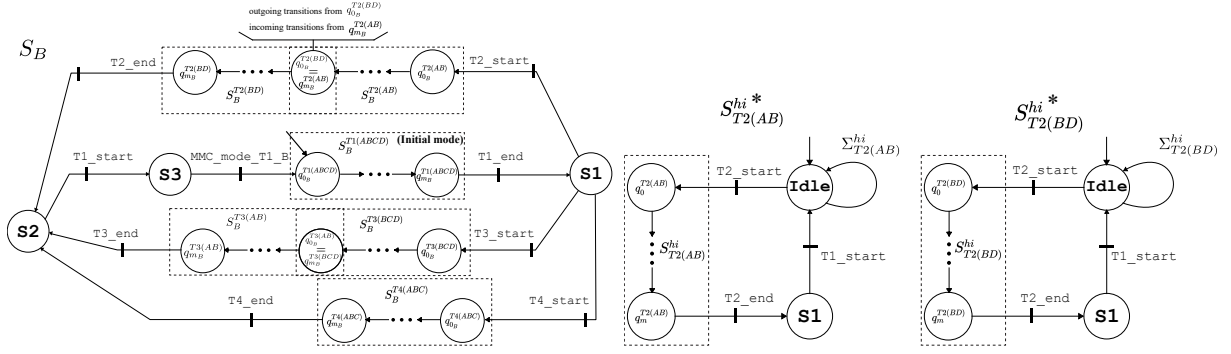
The modification of high-level supervisors is different because two controllers belonging to different elementary control modes may be working concurrently at the CCU, e.g.,  $S_{T2(AB)}^{hi}$ ,  $S_{T2(CD)}^{hi}$ . This is not the case for LCUs, as an HVDC station can only work under a single elementary control mode. Also, with respect to the activation of elementary control mode **T2(BD)**, there is no clear choice as if the control rules of  $S_{T2(BD)}^{hi}$  should be combined with those of  $S_{T2(AB)}^{hi}$  or  $S_{T2(CD)}^{hi}$ . Therefore, in the modification of high-level controllers, we choose to maintain the parallel behavior of different supervisors, while enforcing the idea that those belonging to inactive grid configurations should not disable any transition. We indicate this modification in Fig. 9b, where new versions of the grid supervisors  $S_{T2(AB)}^{hi}$  and  $S_{T2(BD)}^{hi}$  are shown. Two new states were added in each automaton, with state **idle** becoming the new initial and marked state for both. When in the **idle** state,  $S_{T2(AB)}^{hi*}$  and  $S_{T2(BD)}^{hi*}$  do not disable any events, hence the use of self-loops transitions that contain the alphabet of their original version, i.e.,  $\Sigma_{T2(AB)}^{hi}$  and  $\Sigma_{T2(BD)}^{hi}$ . The generation of these events are coordinated by other active controllers until the event `T2_start` is generated, which corresponds to the activation of grid configuration **T2**. Until then,  $S_{T2(AB)}^{hi*}$  and  $S_{T2(BD)}^{hi*}$  remain inactive by not restraining the behavior of low-level controllers.

Thanks to the modifications that were made, the reconfiguration of the supervisory control is simultaneous for all supervisors in the grid. However, once the configuration T1 is deactivated, three choices of reconfiguration are possible. To enforce a unique choice during the system reconfiguration and to make sure it is the correct one based on the isolated cable, we add to our supervisory control a new set of local supervisors.  $Obs_A^{dec}(T3)$ ,  $Obs_B^{dec}(T3)$ ,  $Obs_B^{dec}(T2)$ ,  $Obs_D^{dec}(T2)$ ,  $Obs_D^{dec}(T4)$  and  $Obs_C^{dec}(T4)$ , represented in Fig. 9c, have the role of decentralized observers of the grid configuration. They follow the evolution of local CBLs so as to decide together which reconfiguration event can be generated. The subscript indicates which station they belong to and which grid configuration they are associated with. They are implemented in the LCU and work in parallel with the low-level supervisor of the same station. However, their control rules can only interfere in the generation of reconfiguration events `T1_start`, `T2_start`, `T3_start` and `T4_start`. The final distribution of controllers is shown in Fig. 10.

## Validation of results

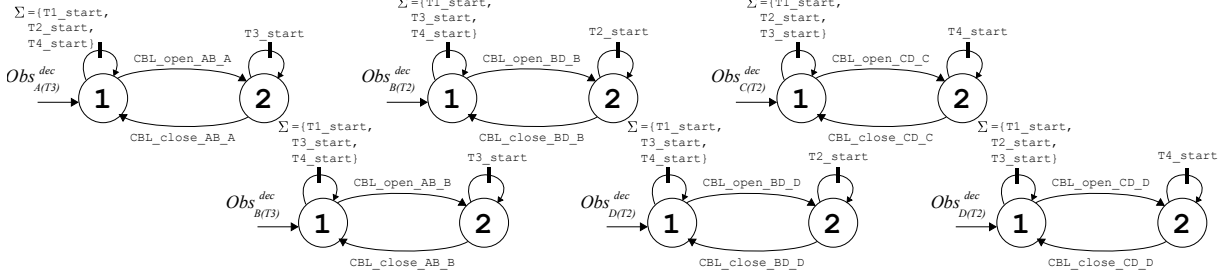
To demonstrate the performance of the obtained solution, the electrical circuit in Fig. 1 is modeled on EMTP-RV. The supervisors are converted into C code and implemented in this software according to the method described in [9]. We simulate a pole-to-ground fault on the BD cable at time  $t = 1s$ . Fig. 11 shows the evolution of the DC voltage and active power injected into grid by positive pole converters.

The supervisory control monitors the events of the grid and limits its behavior to that determined by the specifications used in the synthesis of its supervisors. It achieves that by timely intervening on the controls of relays and converters in the grid. Following the isolation of the cable **BD**, the DC voltage is restored to its nominal value ( $V_p = +320kV$ ) in all newly formed HVDC links. The power transfer is restored about 0.2s after the fault (assuming no delays in communication). If the two AC grids belong to the same synchronous network, restoration of the HVDC links also helps to restore the energy balance



(a) Low-level supervisors.

(b) High-level supervisors.



(c) Decentralized observers of the grid configuration.

Fig. 9: Modifications for the reconfiguration of the supervisory control system.

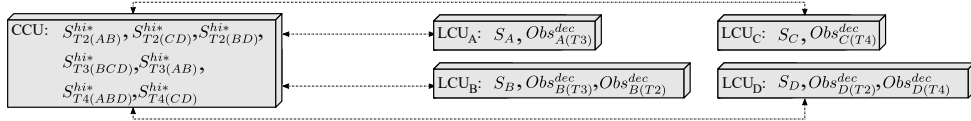


Fig. 10: Distribution of controllers.

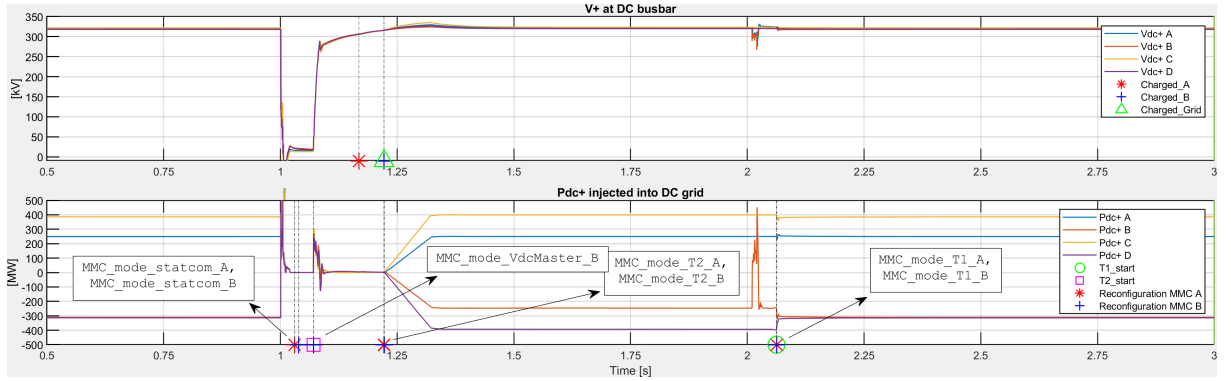


Fig. 11: Simulation results for the reconfiguration  $T1 \rightarrow T2 \rightarrow T1$ .

on a large scale. Indeed, the post-fault intervention of the grid operator is mitigated, which consists in re-routing the new power distribution and compensating for the new losses. The return to initial conditions (grid configuration  $T1$ ) is simulated at time  $t = 2s$ , with the reconnection of cable  $BD$ . First, the  $CBL_{BD,B}$  (line breaker at station B) is closed, assuming the fault has disappeared and the cable has been repaired. The supervisory control waits until the cable is fully charged before closing the line breaker  $CBL_{BD,D}$  and updating the control settings of all MMCs of the grid. In Fig. 11, it can be seen that the distribution of DC power and the voltage curves return to their original patterns once the event  $T1.start$  occurs.

## Conclusions

Using the Supervisory Control Theory (SCT), we were able to develop a reconfigurable supervisory control that safely implements a post-fault restoration scheme in a HVDC grid. The obtained solution enhances the deployment of meshed HVDC networks by automatically modifying control rules according to the current scenario, determined by the grid's partition in our design. The use of a reconfigurable approach simplified the development of the final supervisory control system, which was obtained from the combination of separate supervisors studied in different modes with their own components and specification models. Moreover, the use of a hierarchical and decentralized architecture in the synthesis of controllers prevents the need of direct communication between stations, thereby facilitating the implementation of the final supervisory control system.

The post-fault restoration implemented by the supervisory control presented in this paper was limited to the primary sequence of the protection strategy described in [4]. However, the reconfigurable design of the obtained solution can also be used to simplify the addition of a failure backup sequence (when the CBCs or CBLs of the grid fail to operate correctly), which would be studied as (a) new control mode(s). Future research will explore the update of an existing supervisory control for MTDC grids as well as the impact that delays of communication between local and high-level control units can bring on its performance. Indeed, the coordination of low-level controllers by a high-level one relies on the synchronization of automata. The latter is easy to enforce in a simulated environment in a single machine but this can be a source of errors when physical controllers, distributed over large distances, are used.

## References

- [1] B. Fernández Adiego et al., "Applying Model Checking to Industrial-Sized PLC Programs," in *IEEE Transactions on Industrial Informatics*, vol. 11, no. 6, pp. 1400–1410, 2015.
- [2] O. Despouys et al., "Twenties: Conclusions of a major R&D demonstration project on offshore DC grids," *CIGRE Sess. 45 - 45th Int. Conf. Large High Volt. Electr. Syst.*, 2014.
- [3] P. Verrax, A. Bertinato, M. Kieffer and B. Raison, "Applying Model Checking to Industrial-Sized PLC Programs," in *IEEE Transactions on Industrial Informatics*, vol. 11, Fast fault identification in bipolar HVDC grids: a fault parameter estimation approach, in *IEEE Transactions on Power Delivery*, 2021.
- [4] A. Bertinato, J. Gonzales, D. Loume, V. Creusot, B. Luscan and B. Raison, "Development of a protection strategy for future DC networks based on low-speed DC circuit breakers," *CIGRE Session Paris*, 2018.
- [5] P. J. Ramadge and W. M. Wonham. "Supervisory control of a class of discrete event processes" *SIAM Journal on Control and Optimization*, vol. 25, no. 1, pp. 206–230, 1987.
- [6] A. Ghasaei, Z. J. Zhang, W. M. Wonham and R. Iravani, "A Discrete-Event Supervisory Control for the AC Microgrid," in *IEEE Transactions on Power Delivery*, vol. 36, no. 2, pp. 663–675, 2021.
- [7] A. Kharrazi, Y. Mishra and V. Sreeram, "Discrete-Event Systems Supervisory Control for a Custom Power Park," in *IEEE Transactions on Smart Grid*, vol. 10, no. 1, pp. 483–492, 2019.
- [8] W. Chao, Z. Tang, J. Guo and G. Lin, "Supervisory Control of On-load Tap-Changing Transformers in Xiamen Bipole VSC-HVDC Transmission Demonstration Project," *IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, Chongqing, China, 2018, pp. 2269–2277.
- [9] M. Romero-Rodríguez, R. Delpoux, L. Piétrac, J. Dai, A. Benchaib and E. Niel, "An implementation method for the supervisory control of time-driven systems applied to high-voltage direct current transmission grids," *Control Engineering Practice*, vol. 82, pp. 97–107, 2019.
- [10] G. Faraut, L. Piétrac and E. Niel, "Formal Approach to Multimodal Control Design: Application to Mode Switching," *IEEE Transactions on Industrial Informatics*, vol. 5, no. 4, pp. 443–453, 2009
- [11] S. De Boeck, P. Tielens, W. Leterme and D. Van Hertem, "Configurations and earthing of HVDC grids," *IEEE Power Energy Soc. Gen. Meet.*, pp. 1–5, 2013.
- [12] S. Debnath, J. Qin, B. Bahrani, M. Saeedifard and P. Barbosa, "Operation, control and applications of the modular multilevel converter: A review," *IEEE Transactions on Power Electronics*, vol. 30, no. 1, pp. 37–53, 2015.
- [13] C. G. Cassandras and S. Lafortune, *Introduction to discrete event systems*, 2. ed. New York, NY: Springer, 2008.
- [14] T. Moor, K. Schmidt and S. Perk, "libFAUDES — An open source C++ library for discrete event systems," in *2008 9th International Workshop on Discrete Event Systems*, May 2008, pp. 125–130
- [15] J. Beerten and R. Belmans, "Modeling and control of Multi-terminal VSC HVDC systems," *Energy Procedia*, vol. 24, pp. 123–130, 2012.
- [16] K. Schmidt and C. Breindl, "Maximally Permissive Hierarchical Control of Decentralized Discrete Event Systems," *IEEE Transactions on Automatic Control*, vol. 56, no. 4, pp. 723–737, Apr. 2011.