# Supervisory control synthesis for MMC-based HVDC systems

Miguel Romero Rodríguez

N°d'ordre NNT : 2018LYSEI081

**THESE de DOCTORAT DE L'UNIVERSITE DE LYON**
opérée au sein de
**l'Institut National des Sciences Appliquées de Lyon**

**Ecole Doctorale** EDA160
**Electronique, Electrotechnique, Automatique (EEA)**

**Spécialité de doctorat** : Automatique

Soutenue publiquement le 09/11/2018, par :
**Miguel ROMERO RODRÍGUEZ**

---

# Synthèse de contrôle par supervision pour des systèmes HVDC à base de convertisseurs modulaires multiniveaux

# Supervisory control synthesis for MMC-based HVDC systems

---

Devant le jury composé de :

| | | |
|---|---|---|
| SECHILARIU, Manuela | Professeur des Universités, Univ. de Technologie de Compiègne | Présidente |
| FABIAN, Martin | Professeur des Universités, Univ. de Technologie de Chalmers | Rapporteur |
| ZAMAÏ, Éric | Maître de conférences HDR, Grenoble INP | Rapporteur |
| MOREL, Hervé | Directeur de Recherche CNRS, INSA Lyon | Examinateur |
| CARDOZO, Carmen | Docteur, Ingénieur de recherche, Réseau de Transport d'Electricité | Examinatrice |
| NIEL, Éric | Professeur des Universités, INSA Lyon | Directeur de thèse |
| PIÉTRAC, Laurent | Maître de conférences HDR, INSA Lyon | Co-directeur de thèse |
| DELPOUX, Romain | Maître de conférences, INSA Lyon | Co-encadrant de thèse |
| BENCHAÏB, Abdelkrim | Docteur HDR, Manager R&D, SuperGrid Institute | Invité, Co-encadrant de thèse |
| DAI, Jing | Professeur assistant, CentraleSupélec | Invité, Co-encadrant de thèse |

# Département FEDORA – INSA Lyon - Ecoles Doctorales – Quinquennal 2016-2020

| SIGLE | ECOLE DOCTORALE | NOM ET COORDONNEES DU RESPONSABLE |
|---|---|---|
| **CHIMIE** | **CHIMIE DE LYON**<br>**http://www.edchimie-lyon.fr**<br><br>Sec : Renée EL MELHEM<br>Bat Blaise Pascal 3e etage<br>secretariat@edchimie-lyon.fr<br>Insa : R. GOURDON | M. Stéphane DANIELE<br>Institut de Recherches sur la Catalyse et l'Environnement de Lyon<br>IRCELYON-UMR 5256<br>Équipe CDFA<br>2 avenue Albert Einstein<br>69626 Villeurbanne cedex<br>directeur@edchimie-lyon.fr |
| **E.E.A.** | **ELECTRONIQUE, ELECTROTECHNIQUE, AUTOMATIQUE**<br>**http://edeea.ec-lyon.fr**<br><br>Sec : M.C. HAVGOUDOUKIAN<br>Ecole-Doctorale.eea@ec-lyon.fr | M. Gérard SCORLETTI<br>Ecole Centrale de Lyon<br>36 avenue Guy de Collongue<br>69134 ECULLY<br>Tél : 04.72.18 60.97 Fax : 04 78 43 37 17<br>Gerard.scorletti@ec-lyon.fr |
| **E2M2** | **EVOLUTION, ECOSYSTEME, MICROBIOLOGIE, MODELISATION**<br>**http://e2m2.universite-lyon.fr**<br><br>Sec : Sylvie ROBERJOT<br>Bât Atrium - UCB Lyon 1<br>04.72.44.83.62<br>Insa : H. CHARLES<br>secretariat.e2m2@univ-lyon1.fr | M. Fabrice CORDEY<br>CNRS UMR 5276 Lab. de géologie de Lyon<br>Université Claude Bernard Lyon 1<br>Bât Géode<br>2 rue Raphaël Dubois<br>69622 VILLEURBANNE Cédex<br>Tél : 06.07.53.89.13<br>cordey@univ-lyon1.fr |
| **EDISS** | **INTERDISCIPLINAIRE SCIENCES-SANTE**<br>**http://www.ediss-lyon.fr**<br>Sec : Sylvie ROBERJOT<br>Bât Atrium - UCB Lyon 1<br>04.72.44.83.62<br>Insa : M. LAGARDE<br>secretariat.ediss@univ-lyon1.fr | Mme Emmanuelle CANET-SOULAS<br>INSERM U1060, CarMeN lab, Univ. Lyon 1<br>Bâtiment IMBL<br>11 avenue Jean Capelle INSA de Lyon<br>696621 Villeurbanne<br>Tél : 04.72.68.49.09 Fax :04 72 68 49 16<br>Emmanuelle.canet@univ-lyon1.fr |
| **INFOMATHS** | **INFORMATIQUE ET MATHEMATIQUES**<br>**http://edinfomaths.universite-lyon.fr**<br>Sec :Renée EL MELHEM<br>Bat Blaise Pascal, 3e<br>étage<br>Tél : 04.72. 43. 80. 46<br>Fax : 04.72.43.16.87<br>infomaths@univ-lyon1.fr | M. Luca ZAMBONI<br><br>Bâtiment Braconnier<br>43 Boulevard du 11<br>novembre 1918<br>69622 VILLEURBANNE Cedex<br>Tél :04 26 23 45 52<br>zamboni@maths.univ-lyon1.fr |
| **Matériaux** | **MATERIAUX DE LYON**<br>**http://ed34.universite-lyon.fr**<br><br>Sec : Marion COMBE<br>Tél:04-72-43-71-70 –Fax : 87.12<br>Bat. Direction<br>ed.materiaux@insa-lyon.fr | M. Jean-Yves BUFFIERE<br>INSA de Lyon<br>MATEIS<br>Bâtiment Saint Exupéry<br>7 avenue Jean Capelle<br>69621 VILLEURBANNE Cedex<br>Tél : 04.72.43 71.70 Fax 04 72 43 85 28<br>Ed.materiaux@insa-lyon.fr |
| **MEGA** | **MECANIQUE, ENERGETIQUE, GENIE CIVIL, ACOUSTIQUE**<br>**http://edmega.universite-lyon.fr/**<br><br>Sec : Marion COMBE<br>Tél:04-72-43-71-70 –Fax : 87.12<br>Bat. Direction<br>mega@insa-lyon.fr | M. Philippe BOISSE<br>INSA de Lyon<br>Laboratoire LAMCOS<br>Bâtiment Jacquard<br>25 bis avenue Jean Capelle<br>69621 VILLEURBANNE Cedex<br>Tél : 04.72 .43.71.70 Fax : 04 72 43 72 37<br>Philippe.boisse@insa-lyon.fr |
| **ScSo** | **ScSo***<br>**http://ed483.univ-lyon2.fr/**<br>Sec : Viviane POLSINELLI<br>Brigitte DUBOIS<br>Insa : J.Y. TOUSSAINT<br>Tél : 04 78 69 72 76<br>viviane.polsinelli@univ-lyon2.fr | M. Christian MONTES<br>Université Lyon 2<br>86 rue Pasteur<br>69365 LYON Cedex 07<br>Christian.montes@univ-lyon2.fr |

*ScSo : Histoire, Géographie, Aménagement, Urbanisme, Archéologie, Science politique, Sociologie, Anthropologie

*« (...) de conduire avec ordre mes pensées, en commençant par les objets les plus simples et les plus aisés à connaître, pour monter peu à peu comme par degrés jusques à la connaissance des plus composés (...) »*

René Descartes, *Discours de la méthode*

# Remerciements

L'obtention du grade de Docteur est la culmination d'un travail personnel de longue haleine, qui n'aurait jamais pu aboutir sans les différentes contributions dont j'ai pu en profiter au long de ces trois années. C'est à ceux qui ont bien voulu ramer à mes côtés que je dédie ces pages.

Je tiens tout d'abord à remercier Messieurs Martin FABIAN, Professeur à l'Université de Technologie de Chalmers (Göteborg, Suède), et Éric ZAMAÏ, Maître de conférences HDR à Grenoble INP, pour l'intérêt qu'ils ont accordé à mon travail en ayant accepté de rapporter sur mon manuscrit. Je remercie également, avec la même gratitude, Madame Manuela SECHILARIU, Professeur à l'Université de Technologie de Compiègne, pour avoir accepté de présider mon jury de thèse, et enfin Monsieur Hervé MOREL, Directeur de Recherche CNRS à l'INSA Lyon, et Madame Carmen CARDOZO, Docteur et ingénieur de recherche à RTE, pour m'avoir fait l'insigne honneur d'être de ce jury d'évaluation en tant qu'examinateurs et ce, malgré des emplois du temps très chargés. C'est pour moi un honneur et une fierté d'avoir pu rassembler ces cinq éminentes personnalités au sein de mon jury de thèse.

Je souhaite exprimer aussi mes profonds remerciements à toute l'équipe de supervision de thèse. Leurs efforts combinés ont assuré l'arrivée à bon terme de ces travaux, et sous leur guide j'ai pu transformer un projet de recherche en une riche expérience professionnelle et humaine. Je tiens à remercier Éric NIEL pour la confiance qu'il m'a accordé, puisqu'il m'a introduit, déjà avant la thèse, à ce sujet de recherche qui s'est avéré stimulant et sur lequel j'ai eu le plaisir de continuer à travailler sous sa direction. Pour cela je lui remercie et je lui souhaite de profiter d'une retraite bien méritée. Deuxièmement, je voudrais remercier Laurent PIÉTRAC, qui s'est joint au reste de l'équipe en début de thèse. Je lui suis très reconnaissant pour les échanges que l'on a pu avoir et pour sa constante disponibilité. Le plus important, ses précieux conseils et sa pédagogie ont réussi (enfin presque!) à développer en moi une démarche de travail et de réflexion rigoureuse et fondée. Ensuite, je ne peux qu'exprimer mes profonds remerciements à Romain DELPOUX, qui m'a encadré tout au long de ce projet et avant la thèse. Ses connaissances techniques ainsi que ses propres expériences professionnelles m'ont aidé à combiner avec succès les divers domaines traités dans cette thèse, ainsi que de mieux comprendre le monde de la recherche scientifique.

Ce travail n'aurait pas été assez complet sans la collaboration et l'expertise de mes deux encadrants à SuperGrid: Abdelkrim BENCHAIB et Jing DAI. Je remercie Abdelkrim pour toutes les fructueuses conversations que l'on a eue tout au long de cette thèse sur les plus divers sujets, ainsi que pour son engagement à faire converger les aspects industriels et scientifiques de cette thèse. Je lui suis reconnaissant et j'en tire des conclusions de grande valeur de ses conseils, tant humains que professionnels. Finalement, je ne peux que remercier Jing pour sa totale disponibilité et son aide. Infatigable correcteur, je lui remercie de m'avoir poussé à ne me conformer qu'avec le meilleur résultat possible. Je garde un très bon souvenir de tous les échanges que l'on a eu tout au long de ces années. Malgré le grand nombre de personnes qui ont composé cette équipe et la disparité entre leurs compétences, je suis fier de pouvoir affirmer que l'aboutissement de cette thèse est réalité grâce à leur encadrement exemplaire.

Je souhaite remercier aussi chaleureusement le directeur du programme P1 de SuperGrid Institute, Bruno LUSCAN, qui nous a fait confiance et qui nous a soutenu tout le long du déroulement de ce travail. Je n'oublierai pas non plus les autres responsables de sous-programme: Serge POULLAIN, Alberto BERTINATO et Sébastien SILVANT, pour leurs contributions d'une façon ou d'une autre à la bonne réussite de cette thèse. Mes remerciements vont aussi à la direction générale de SuperGrid Institute et à sa tête Hubert de la GRANDIÈRE pour m'avoir donné l'opportunité de travailler dans cet environnement unique de recherche et développement.

À mes collègues du P1, je leur remercie d'avoir créé un espace de travail où la bonne humeur et la collaboration sont à l'ordre du jour, et d'avoir arrosé cette bonne ambiance en dehors du bureau. Parmi eux je nomme ceux qui sont déjà docteurs (Swann, William, Kosei, Ahmed, Dieynaba et Janailson), ceux qui le seront bientôt (Juan Carlos, Amjad, Guilherme, Raga, Nicolás, Paul et Pascal) et je n'oublie pas Éric, Boussaad, Sellé, Manuel, Léo, Philippe et tous les stagiaires et alternants avec lesquels j'ai eu le plaisir de travailler. Je voudrais également étendre mes remerciements à Romain, Jérémie, David, Setareh, Thibault et Kévin, qui m'ont chaleureusement accueilli pendant mon séjour au Laboratoire Ampère.

Je voudrais laisser ici un mot à tous mes amis, puisque qu'ils ont contribué d'une façon ou d'une autre à ma réussite; soit pour leur soutient, soit pour leur habitude de squatter dans mon canapé, soit pour les nombreuses fêtes, conversations, voyages et rigolades. Donc Javi, Raúl, Toni, Enrique, Patrick, Natalia, Ana, Giovanna, Berni et tous ceux que je n'ai pas pu citer ici, merci!

Évidemment, rien de tout cela n'aurait été possible sans l'appui inconditionnel de mes parents, Enrique et Isabel, et de mon frère, Víctor. Je vous remercie pour tous les sacrifices humains et financiers que vous avez fait afin que je puisse arriver à ce stade. Enfin et surtout, je remercie Paola d'être à mon côté pour le bon et pour le mauvais et de m'accompagner dans ce long projet qui est la vie.

# Contents

# List of figures

# List of definitions and theorems

## Definitions

# Theorems

# List of tables

# Acronyms

**AC** . . . . . Alternating Current

**ACCB** . . . . . Alternating Current Circuit Breaker

**AGC** . . . . . Automatic Generation Control

**ASC** . . . . . Automated Supervisory Control

**BM** . . . . . Breaking Module

**CB** . . . . . Circuit Breaker

**CFD** . . . . . Control Flow Decomposition

**CS** . . . . . Control System

**DC** . . . . . Direct Current

**DCCB** . . . . . Direct Current Circuit Breaker

**DES** . . . . . Discrete Event System

**DFA** . . . . . Deterministic Finite Automaton

**DLL** . . . . . Dynamic-Link Library

**EMS** . . . . . Energy Management System

**EMTP** . . . . . Electromagnetic-Transients Program

**ENTSO-E** . . . European Network of Transmission System Operators for Electricity

**FB** . . . . . Function Block

**FSA** . . . . . Finite-State Automaton

**FSM** . . . . . Finite-State Machine

**HSS** . . . . . High-Speed Switch

**HV** . . . . . High-Voltage

**HVAC** . . . . . High-Voltage Alternating Current

**HVDC** . . . . . High-Voltage Direct Current

**IED** . . . . . Intelligent Electronic Device

**IGBT** . . . . . Insulated-Gate Bipolar Transistor

**IL** . . . . . Instruction List

**INELFE** . . . . Interconexión Eléctrica Francia-España

**I/O** . . . . . Input/Output

**IET** . . . . . Institute for Energy Transition

**LCC** . . . . . Line-Commutated Converter

**LD** . . . . . Ladder Diagram

**MMC** . . . . . Modular Multilevel Converter

**MTDC** . . . . Multi-Terminal Direct Current

**OHL** . . . . . Overhead Line

**OM** . . . . . Operational Mode

**OP** . . . . . Operational Procedure

**OS** . . . . . Operational State

**PIR** . . . . . Pre-Insertion Resistor

**PLC** . . . . . Programmable Logic Controller

**RES** . . . . . Renewable Energy Sources

**RMS** . . . . . Root Mean Square

**RO** . . . . . Region of Operation

**RTU** . . . . . Remote Terminal Unit

**SCADA** . . . . Supervisory Control and Data Acquisition

**SCT** . . . . . Supervisory Control Theory

**SF** . . . . . Service Function

**SFC** . . . . . Sequential Function Chart

**SFCL** . . . . . Superconducting Fault Current Limiter

**SGI** . . . . . SuperGrid Institute

**SM** . . . . . Submodule

**ST** . . . . . Structured Text

**TSO** . . . . . Transmission System Operator

**VSC** . . . . . Voltage-Source Converter

**XLPE** . . . . . Cross-Linked Polyethylene

# 1

# General introduction

## Contents

## 1.1 Trends in power transmission systems

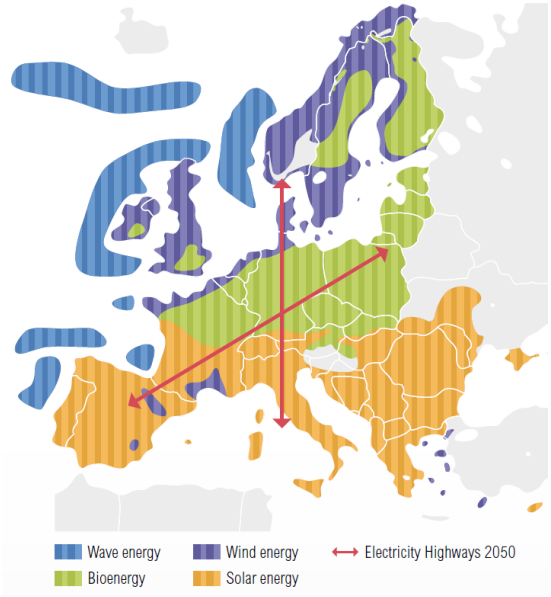
Technology is one of the main pillars on which modern society relies: communication, finance, logistics and transport are among a variety of sectors based on technological solutions. Despite the extraordinary versatility of today's technologies, the majority of them are powered by electricity: computers, lighting, trains...; electric power turns out to be essential for our life. In the future, this dependency is expected to increase as a result of the rising energy consumption and the decreasing use of fossil fuels as primary sources of energy. Indeed, concerns on sustainable development and environmental issues have pushed many countries to revisit their national policies in order to encourage the decarbonization of the energy sector. The European Union and its Member States, for instance, have agreed to produce 20% of their energy mix from Renewable Energy Sources (RES) by 2020, while China expects a 15% share of non-fossil energy for the same period [Nat16; Eur10]. Promoted by those initiatives, renewable energy has grown at an astonishing speed, as it accounted for 77% of the newly installed generation capacity in Europe from 2007 to 2015 [Eur16].

However, the integration of a massive amount of electricity produced from RES into the existing power transmission systems represents a challenge still to this day. The inherent volatility of the renewable energy sources disrupts the traditional way of power system operation which is based on a largely predictable consumption and fully controllable generators. Also, since renewable power stations, such as offshore wind farms, are most likely to be located far from the center of electricity consumption, RES drive the existing electric power transmission systems, based on Alternating Current (AC) technology, close to their physical limits. In consequence, the European Network of Transmission System Operators for Electricity (ENTSO-E) reported the need to reinforce the present grid if High-Voltage (HV) transmission corridors that connect the regions with high-potential renewable resources (hydro in Scandinavia, wind in the North Sea and the Atlantic Ocean, solar in the Mediterranean region, etc.) to the densely populated areas all across the European continent are to be developed. A significant increase of the interconnection capacity with neighboring countries is expected to improve the security and the reliability of the power supply by smoothing the effect of the volatility of these type of energy sources [Shi17].

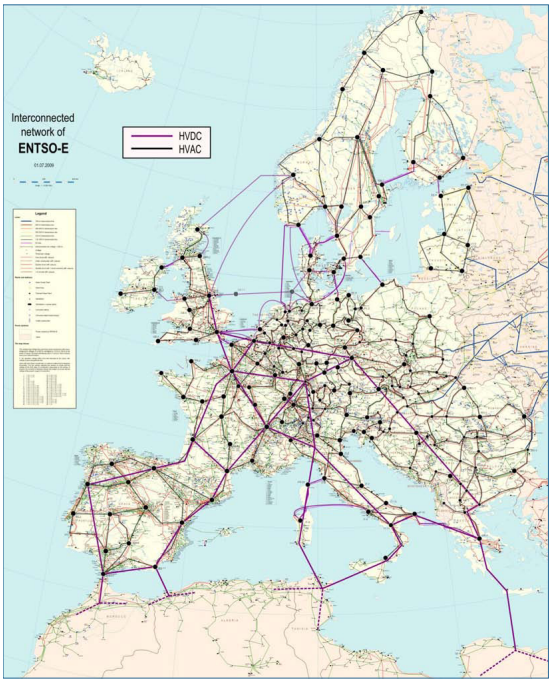
High-Voltage Direct Current (HVDC) transmission is considered as the most promising and technically feasible solution to the fundamental upgrade of the existing AC transmission systems [Mar09; Ahm12]. For long distances, power transmission becomes unfeasible in AC because of the capacitive effect in the high-voltage link, whether it is a submarine or subterranean cable or an Overhead Line (OHL). From a certain distance, which depends on the power exchanged, the capacitive losses become too great: the reactive power exchanged becomes too large in comparison with the useful active power. This makes offshore wind energy virtually inaccessible by the conventional High-Voltage AC (HVAC) technology beyond 100 km. In HVDC

transmission, however, because only the active power is exchanged, the capacitive losses are eliminated and less limitations on the length of the link exist. This makes HVDC transmission the most economical solution after a break-even distance [Her10]. Furthermore, the absence of frequency in DC systems allows the interconnection between two asynchronous AC transmission systems [Dor13]. Although this makes it possible to prevent frequency disturbances from being propagated between two distinct AC grids, an associated AC power disturbance would still impact the second AC system through the HVDC link, in addition to imposing certain time-response constraints on the operation of the HVDC systems [Abe17; Gon18].

Existing HVDC applications have been limited for the most part to ad hoc point-to-point links responding to a specific need [Arr07]. In the last years, however, the idea of working towards the large scale interconnection of HVDC links in order to form a Multi-Terminal DC (MTDC) grid [Bui17] has drawn interest from researchers of both the academia and the industry. If the number of interconnected links increases, MTDC grids may evolve into a bulk power grid, named supergrid, overlaid to the AC tranmission system [Her16]. The supergrid is considered to be an attractive solution to accommodating geographically spread RES (Figure 1.1a) and to mitigating the congestion of existing HVAC grids; while allowing a better integration of the different electricity markets and increasing the reliability and flexibility of the power supply [Her10]. The realization of such an ambitious project, however, is foreseen to be gradually developed from the interconnection of separate HVDC links, given the great amount of challenges that the construction of an international network of "electricity highways" implies (Figure 1.1b).




(a) RES development by 2050 [ENT13]




(b) Possible HV network by 2050 [Ene11]

Figure 1.1 – Perspectives on the development of a European Supergrid

## 1.2 Operation of HVDC systems

Electric power systems are typically composed of a large number of components arranged in separate geographical zones so as to ensure the generation, transmission and distribution of power from the power stations to the end users. As defined by the European Commission, the entity entrusted with the transmission of power from generation units over the electrical grid to regional or local electricity distribution operators is the Transmission System Operator (TSO).

Traditional AC power systems are mostly dominated by large synchronous rotating machines that are exploited by a reduced number of utilities. Transmission grids were developed so as to connect the production and consumption areas based on the requirements and long-term forecasts of a few utilities. This structure facilitates the monitoring and operation of the whole power supply chain from a centralized control center that adjusts the balance between the generation and demand by regulating the power produced by each one of the generators. Hence, AC power systems are a highly integrated network of large rotating masses that help compensate the power balance whenever a disturbance appears in the grid. However, because an HVDC system is largely composed of power electronics devices less capable of compensating a power unbalance in absence of control in comparison to AC generating units, disturbances are quickly propagated within the DC grid. If such disturbances are not properly counteracted by means of control and protection actions and a power balance condition is not restored within a short time range (hundreds of milliseconds), cascading component failures could eventually lead to a blackout of the grid and let the disturbance propagate to the adjacent AC networks, which is unacceptable. Furthermore, a certain coordination between the components of the HVDC system is needed so as to enhance system security whenever the multiple protection and control schemes are triggered by a disturbance.

With respect to AC systems, however, human intervention is further limited in HVDC systems during the real-time operation of the grid, because of the short-time range in which the different control and protection actions should be performed. Although the main monitoring, analysis and control functions are largely automated in AC power systems [Don17], whenever automated responses do not suffice, the component coordination tasks are mostly based on human expertise [Zha10]. In consequence, the development of an Automated Supervisory Control (ASC) that replaces the human operator for real-time monitoring and component coordination tasks is essential if an acceptable grid operation is to be expected. Such a supervisory control needs to be reactive to the grid's evolution so as to respect the required specifications in an automated manner, while the decision-making from the human operator is minimized as much as possible.

Given that on the one hand, the control of the components in an HVDC system is determined by the change in the digital signals that command them and that, on the other hand, their continuous-time dynamics can be differentiated during the distinct phases of operation,

the evolution of an HVDC system can be represented as an event-driven process [Ess99]. In consequence, it is possible to use Discrete Event Systems (DES) modeling to monitor the state of the components in the grid. Furthermore, given that each distinct phase of operation comprises a particular configuration of the system where a set of components must meet a set of specifications, the design method proposed in this thesis is based on formal approaches within the Supervisory Control Theory (SCT) framework based on automata, which constitute a powerful tool for the construction of safe models of the grid, which are obtained from the combination of low-level component models, and adapt the supervisory control architecture to the nature of power transmission systems, which are geographically dispersed, hierarchically operated and experiencing diverse situations during their operation.

To conclude, this thesis aims to develop, implement and validate a method for the design and practical implementation of an automated supervisory control system, based on DES modeling and formal methods within the SCT framework, which ensures a coordinated and secure operation of the network from the component level to the grid level. The motivation behind this thesis reflects the growing interest in MTDC grids and the concerns raised by the operational challenges that future grids will face.

## 1.3 Application of DES approaches to power systems

An hybrid system is a system where the continuous-state space is partitioned into several discrete states, with different continuous-time dynamics in each discrete state, and whose evolution depends on the occurrence of asynchronous discrete events [Kou00]; whether they occur following a disturbance (voltage collapse, overcurrent...) or an automated or manual control action (control activation signals, tripping orders...). Thus, it can be retained from the previous sentence that HVDC systems in particular, and power systems in general, are hybrid systems. If only the event-driven evolution of a power system is considered, the latter can be modeled as a DES. While control and protection commands are naturally modeled as events, the trajectory of the continuous-state variables of the system needs to be abstracted so that the discrete-event behavior is identified. Hence, an event $e_i$ occurs whenever the trajectory of a state variable $x$ crosses an hypersurface $h_i$, which corresponds to the boundary of two discrete states [Zha06], as shown in Figure 1.2. In this manner, the continuous-state space is separated into two adjacent sets representing the discrete-event dynamics of the system.

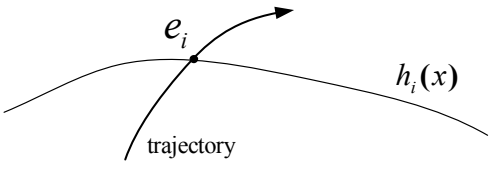



Figure 1.2 – Event generation from a continuous-state space [Zha06]

The interest of modeling the event evolution of a power system as a means to ease the understanding of the complex interactions within it has been pointed out for more than 20 years [Pro95; His00]. However, the large number of components and the hierarchical structure of power system's operation makes that difficult to be achieved. In consequence, most of the applications of DES approaches to power systems found in the literature are limited to a set of components or consider an overly simplified system. In [Pro95], for instance, the authors model a transmission line as a Finite-State Automaton (FSA) with two possible states (in service or out of service) that evolves following the occurrence of *trip* and *restoration* events. The global state of the system, obtained by composition of the $n$ line models, is used to realize security assessment. Whenever a maximum number $m$ of the $n$ lines are out of service, the authors calculate the power flow of each possible discrete state to verify that the thermal limits of any transmission line are not violated and that no network is isolated due to the trip of a certain line. If this is not the case, the considered state is defined as insecure. Hence, the transmission lines that were out of service are restored until all lines are back to service again, so that no insecure states are reached. Likewise, the authors of [Bis04] propose to monitor the state of the CBs and associated relays in a transmission line by means of a Petri net model, which is used to detect the possible faults associated with the operation of the CBs. The obtained model serves as a tool that helps the human operator understand the various causes behind a system failure.

The authors of [Wen07] synthesize a fault-tolerant supervisor within the SCT framework for an AC power system, which is represented by a simplified FSA whose states model the state of connection of the transmission lines or the state of activation of the system controllers. Then, the obtained supervisory control enables the activation of a certain controller or the reconnection of a certain line, depending on the operational state of the system, so as to avoid blackout and ensure the return to a non-faulty behavior whenever one or several transmission lines are faulted. At last, outside high-voltage applications, DES modeling in the interest of developing a supervisory control can be found for the mode-management of micro-grid systems [Kha07; Dra14]. In both references, the control of the different modules of a micro-grid system is predefined by a high-level controller, in the form of an automaton, which determines the control actions at each state so as to avoid voltage deviation and coordinate the charging of the storage elements in the system.

However, all of the referred methods lack the necessary conditions to be regarded for the control of a complex multi-terminal system, as the methods found in the literature are often proposed as a solution to an specific example and cannot be generalized. Furthermore, the lack of detail on the physical behavior of the system limits the efficiency of the proposed solution for real applications. Finally, most of the solutions in the literature are focused in the monitoring of the system and not so much on the development of an automated controller of the system.

Approaches based on DES modeling and supervisor synthesis have been applied in the literature to the operation of AC power systems. The proposed methods, however, are not appropriate

for their general use in real systems. The lack of a generic and formal method makes it difficult to apply the proposed solutions outside the considered case studies, especially since the physical behavior of the components forming the grid is often not thoroughly modeled. Furthermore, only centralized controllers were presented in the literature, whereas a distributed supervisory control that defines the envelope of operation at different levels of detail would be more adequate for power transmission systems, which are geographically dispersed and operated hierarchically.

## 1.4 Contributions and outline of the thesis

The main contribution of this thesis is to propose a systematic, self-sufficient and incremental method allowing to design and implement a supervisory control for HVDC systems.

The main contributions are summarized below:

- Proposal of an exhaustive methodology for the identification of the functional and monitoring requirements of an HVDC system, which results in an identification of the component's behavior and the modes of the grid.
- Modeling of the generic behavior of the components in an HVDC station: discrete-event modeling of the continuous-time behavior of the system components so as to capture their complete behavior, independently of the mode of operation.
- Proposal of a flexible supervisory control design method based on a vertical and modal decomposition of the system that eases the integration of new aspects (components, specifications...) with the minimum impact on the existing control structure and ensures that the obtained result is safe by construction, so that no verification is needed afterwards.
- Definition of the behaviors that can be potentially abstracted from the station models, so as to clearly distinguish the information essential to the coordination of the components in a station and that one essential to the coordination of the stations themselves, thus simplifying the models without compromising the properties of the supervisory control.
- Proposal of a multi-level implementation method, based on generic expressions that can be implemented in C code, which integrates the different models resulting from the design process at different levels of control and relates them to the continous-time physical components via an interface level. Furthermore, the proposed method is can be automated for the most part.

This thesis is organized as follows:

Chapter 2 compares current AC transmission systems to HVDC systems, so as to highlight their main differences and how they impact the security control of the grid at the control center. Then, a general overview of the different configurations of an HVDC grid are presented and the main components of the system are introduced in view of their modeling as DES.

Chapter 3 presents the formal methods used throughout the thesis. The concept of deterministic automaton is introduced and its manipulation through formal methods within the SCT for the synthesis of a supervisory control system is presented. Finally, previous works in the literature concerning the implementation of an automata-based supervisory control are presented.

Chapter 4 gives a thorough presentation of the proposed design and implementation method. First of all, a functional and monitoring analysis of each component is performed in view of their modeling as generic automata. A compositional approach is then used so as to obtain an appropriate control architecture within the hierarchical structure of the control system of HVDC grids. The method is illustrated through its application to the start-up procedure of a point-to-point link. Finally, a method for the implementation of the obtained control system is proposed.

Chapter 5 extends the design approach proposed in Chapter 4 so that different modes of operation can be integrated within the same supervisory control without interruption during commutations of mode. Hence, a first stage concerns the intramodal design of the models corresponding to the internal behavior of a mode, which are later extended in the intermodal study in order to ensure an uninterrupted and secure control of the grid.

Chapter 6 tests the design and implementation method in a large-scale MTDC grid. The proposed solution is applied to several modes of operation of a 3-terminal MTDC grid and validated through simulations in a dedicated software.

Chapter 7 draws the major conclusions of the thesis and proposes the direction of future work.

## 1.5 Research environment

In 2010, the French government launched a call for proposals for the creation of *Institutes for Energy Transition* (IETs), which are expected to be collaborative research platforms in the field

| SuperGrid Institute | Industrial | Academic and institutions |
|---|---|---|
| **Shareholders** | Alstom<br>EDF<br>GE<br>Nexans<br>Vettiner | CentraleSupélec<br>Ecole Centrale de Lyon<br>Grenoble INP<br>INSA Lyon<br>Université Claude Bernard Lyon 1<br>Université Joseph Fourier<br>Université Paris Sud |
| **Associated laboratories** | | CREMHYG<br>G2Elab<br>Laboratoire Ampère<br>Laboratoire des Signaux et Systèmes (L2S) |

Table 1.1 – SuperGrid Institute partners

of low-carbon energy, bringing together the expertise of industry and public research in the logic of public-private co-investment and close cooperation between all stakeholders of the sector. The SuperGrid Institute (SGI) project was successfully selected to develop the technologies of the future HVDC transmission networks. The institute counts several industrial and academic partners among its shareholders, summarized in Table 1.1.

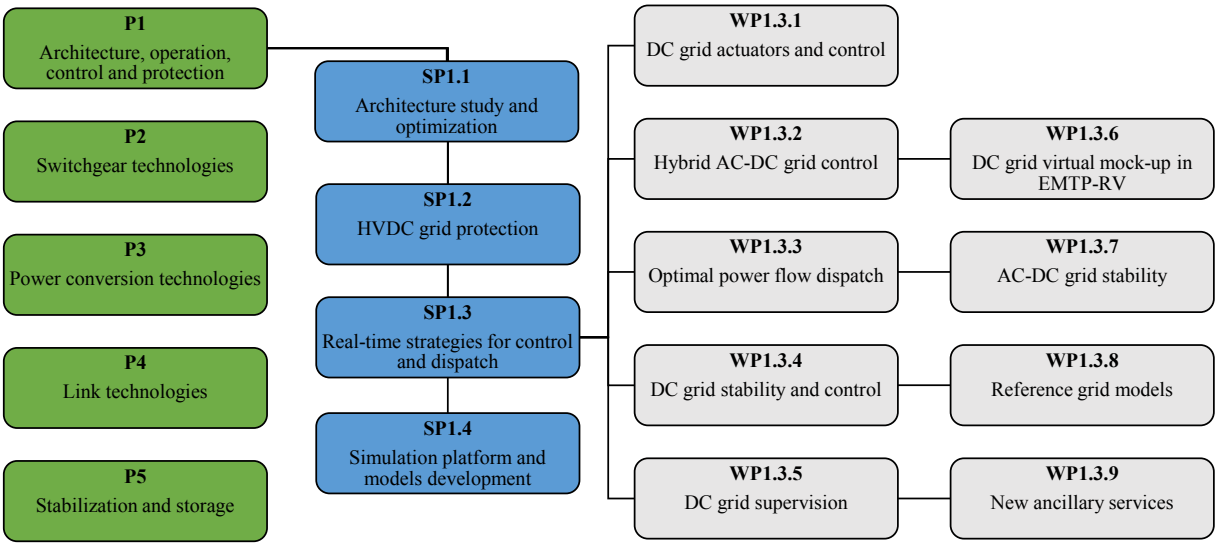



Figure 1.3 – SuperGrid Institute research programs

Research work is carried out in collaboration with public laboratories and divided into five programs, given in Figure 1.3. Program 1 is dedicated to the study of the grid's architecture along with the associated operation, control and protection strategies; and to the development of simulation tools for the rest of SGI. One of the main challenges related to the development of MTDC grids concerns the operational management of increasingly large and complex systems. Thereby, the subprogram 1.3 is in charge of the research on control and operation solutions for HVDC networks. In collaboration with the Laboratory Ampère, this PhD thesis was carried out within the work package (WP) 1.3.5 "DC grid supervision".

## 1.6 List of publications

### Journal

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, no. 1, pp. 97-107, 2019.

**Conference**

M. Romero Rodríguez, R. Delpoux, L. Piétrac, J. Dai, A. Benchaib and E. Niel, “Supervisory control for high-voltage direct current transmission systems”, *IFAC Proceedings Volumes (IFAC-PapersOnline)*, vol. 50, no. 1, pp. 12326-12332, 2017.

# 2

# Supervision requirements in HVDC systems

## Contents

## 2.1 Introduction

One of the roles of the TSO is to maintain a continuous balance between electricity supply from power stations and demand from consumers, in a manner that fluctuations in frequency are compensated for and interruptions of supply are avoided by means of automatic control, while integrating in the system operation the ancillary services and the maintenance planning provided by the electric utilities via computer-assisted functions. In addition, the TSO commonly manages the power dispatch in overload situations, which could lead to a fault in the system, by means of human decision-making. Traditionally, the real-time dispatch of the optimal combination of power generation and ancillary services in an AC transmission system is computer-assisted whereas the security management of any contingent events that might cause the balance between supply and demand to be disrupted, e.g. a frequency disturbance or a congested line, are automated or based on human decisions.

However, the fast-dynamics of HVDC systems along with the additional constraints due to its interconnection with surrounding AC systems makes it difficult to utilize in this type of grids the same operational structure of AC transmission systems. Thus, in this chapter, an overview of the current supervision methods for AC systems is first given and its limits are discussed. Then, HVDC systems are presented and the focus is set on the configurations and functional structure of an HVDC system so as to introduce the possible interactions between the control and protection systems and a supervisory control managing the different events that appear in the HVDC grid. Finally, HVDC systems are compared to their AC counterpart, so that their fundamental differences and how they affect system operation are highlighted.

## 2.2 Supervisory control of HVAC transmission systems

A High-Voltage AC (HVAC) transmission system is composed of high-voltage three-phase AC transmission lines and substations including protection devices and a transformer that steps up or down the power, depending on whether the transformer is connected to a power station or to the distribution grid, as illustrated in Figure 2.1. The whole power system consists of the generation system, the transmission system and the distribution system. Currently, an AC power system is monitored and operated from a centralized facility, known as the control center. The TSO is in charge of the operation of the transmission portion of the power system. Depending on the number of substations forming the grid, however, several control centers may be needed to monitor a regional portion of the power system.

As illustrated in Figure 2.2, a set of computers at the control center communicate with the apparatus in each power station via the Remote Terminal Units (RTUs), which are the entire group of devices, functional modules, and assemblies that are electrically interconnected to affect the remote station supervisory functions [ANS87]. In particular, the RTUs interface with

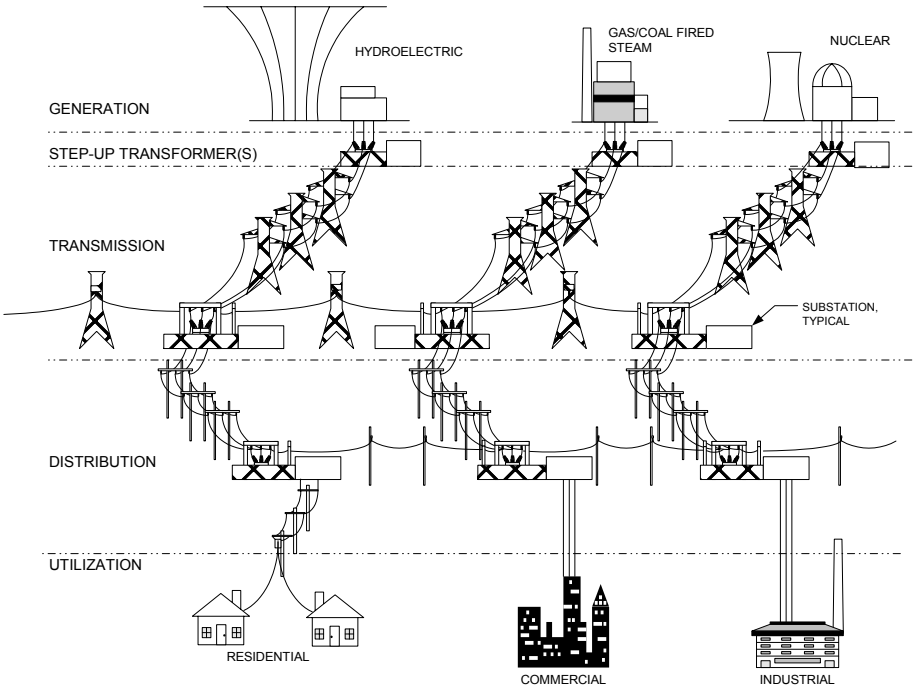



Figure 2.1 – AC power system [Bar04]

the apparatus in the power stations by means of associated Input/Output (I/O) modules that gather status and meter data, such as equipment status or voltage and current measurements, and transmit the control signals and reference points to the corresponding devices. While meter data is sampled periodically, typically every second or a few seconds, status data is sent whenever there is an actual change of status [Bar04]. The system including the data collection computers at the control center and the RTUs in the field is referred to as the Supervisory Control and Data

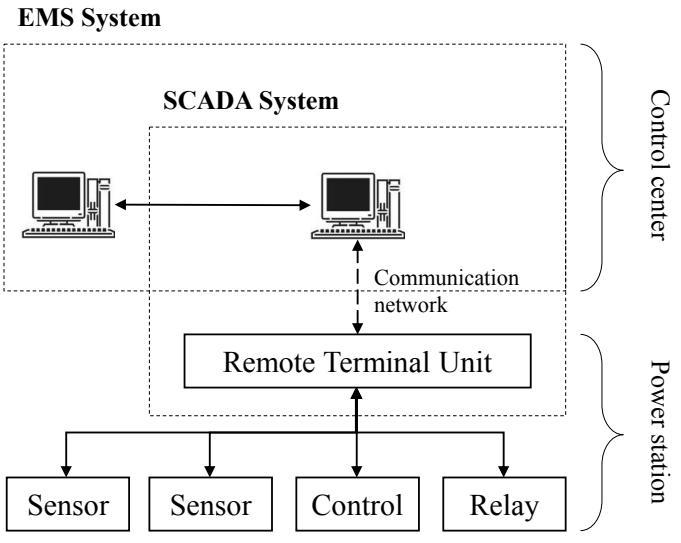



Figure 2.2 – Control center block diagram

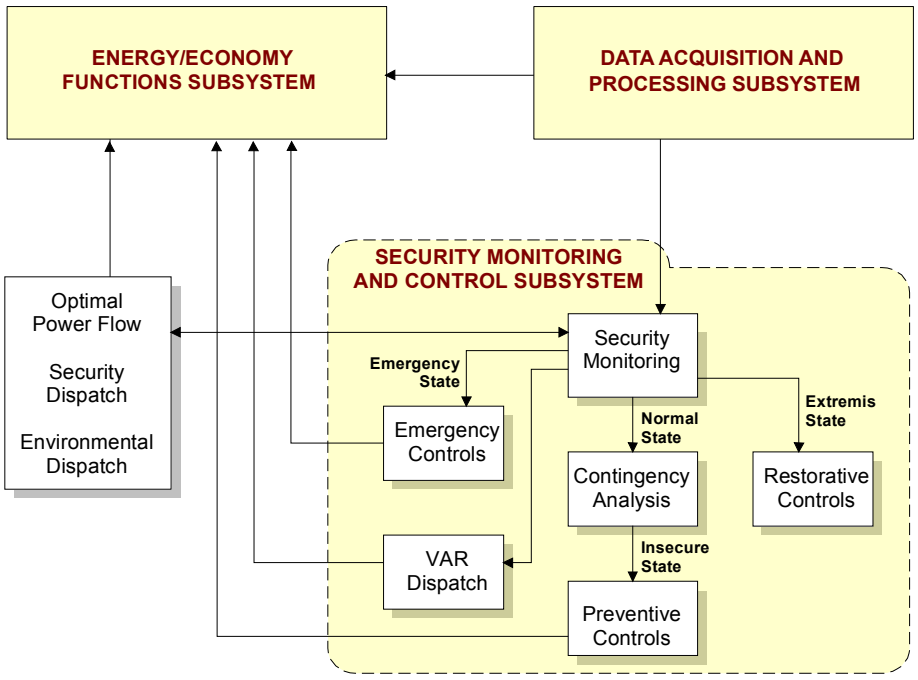



Figure 2.3 – Functional diagram of an EMS (adapted from [Mel04])

Acquisition (SCADA) system. The gathered data is then processed at the control center by a set of computer or human assisted functions. The ensemble of operational functions carried out by a TSO forms the Energy Management System (EMS). From a functional viewpoint, the EMS can be split into the three distinct subsystems shown in Figure 2.3 [Mel04].

### 2.2.1 Data acquisition and processing subsystem

The objective of this subsystem is to obtain an accurate estimate of the state of the grid from the data collected by the SCADA system. Once the data is received, status data is utilized to estimate and display the topology of the network. Meter data, on the other hand, is utilized to compute the best possible estimate of the power flow in the system. As a result, the electrical circuit of the grid and the power flowing in each line are reconstructed.

### 2.2.2 Energy/economy functions subsystem

Because in traditional power systems, electric power is produced by large rotating turbo-generators, the balance between generation and load can be judged by measuring the system frequency [Rau14]. Thus, an important function of the energy and economy functions subsystem is to manage the power reference values to be transmitted to each generation unit, with economic and security constraints, so that the frequency is maintained around its nominal value for each time instant. For this reason, the load forecasting function predicts the requested load hours or days in advance by means of probabilistic approaches, supported by historical data and

real-time data like outside temperature. The resulting forecast is used to select the appropriate reference points for the generators by taking into account the requested load, the resources costs, the amount of power that can be transacted with neighboring transmission grids and the environmental objectives.

### 2.2.3 Security monitoring and control subsystem

The security of an electric power system refers to the degree of risk in its ability to survive imminent disturbances without interruption of customer service [Kun04]. In AC transmission systems, this is equivalent to ensuring that, in the long-term, the balance between power generated and load will be maintained for any possible contingency. Thus, the security of the system is maintained by continuously monitoring and performing control actions when this balance is compromised. However, a power system can have an infinite number of operating points, which results in an enormous amount of situations to be analyzed in terms of security.

T. Dy-Liacco proposed in [DyL67] a classification, later extended by L.H. Fink and K. Carlsen in [Fin78], which allows to categorize the operating points of the system into five operational states based on the requirements to be respected by the different parameters collected by the SCADA system (voltage, frequency, current, temperature, etc.). In turn, such constraints are of two types:

- *Inequality constraints* such as: limits on system frequency, limits on voltage magnitude, thermal limits, voltage stability limits, transient stability limits, etc.
- *Equality constraints* expressing the fact that the balance between the generated power and the load should be respected, i.e. the power flow equations.

Then, depending on whether the above constraints are satisfied or not, the operating points of a power system are classified into the different operational states in Figure 2.4 [DyL67; Fin78]:

- *Normal*: all equality and inequality constraints are satisfied for the present system and for any foreseeable and probable contingency;
- *Alert*: all equality and inequality constraints are satisfied for the present system, but not for one or more foreseeable (and probable) contingencies;
- *Emergency*: all equality constraints are satisfied, but one or more inequality constraints are violated;
- *In Extremis*: one or more equality constraints are violated, and one or more inequality constraints are violated;
- *Restorative*: all inequality constraints are satisfied, but one or more equality constraints are violated.

Although good practice recommends the respect of equality and inequality constraints at all times, it is economically expensive and technically complicated to achieve this. Furthermore, recent trends in the energy market (deregulation, independent operation) usually encourage the

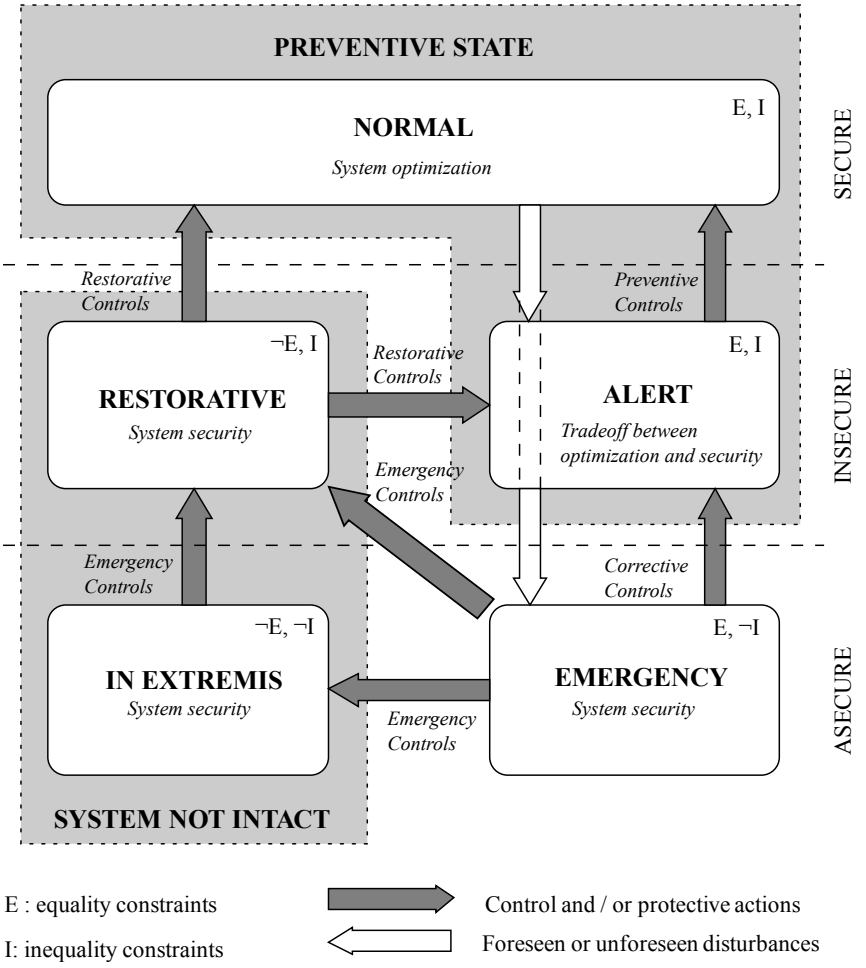



Figure 2.4 – Power system operational states (adapted from [Fin78; Pav00])

increase of the power exchange, which results in overloaded lines and operating conditions close to the emergency state. In consequence, when the system is in normal operating state, it is necessary to perform a security assessment at predefined time intervals in order to asses the most critical (or imminent) disturbances that could occur in the system [Sha05]. Their impact on the grid is then estimated [Weh89] and ranked by means of *contingency ranking* methods. A *contingency analysis* is then performed in order to determine how the power should be allocated by the Automatic Generation Control (AGC) to each power station or how the reactive power should be managed by the VAR dispatch for each disturbance in the contingency ranking.

Whenever the system deviates from the normal state, however, control actions are exercised so as to bring the system back to the normal state. Depending on the violated constraints, these actions are characterized as preventive, corrective, emergency or restorative. Typically, security control is subordinated to a human operator and comprises a number of automated and manual actions of different time scales. For instance, preventive and corrective actions typically correspond to an automatic response of the frequency control, schematized in Figure 2.5. The primary frequency control applies a predefined droop so as to counteract a frequency perturba-

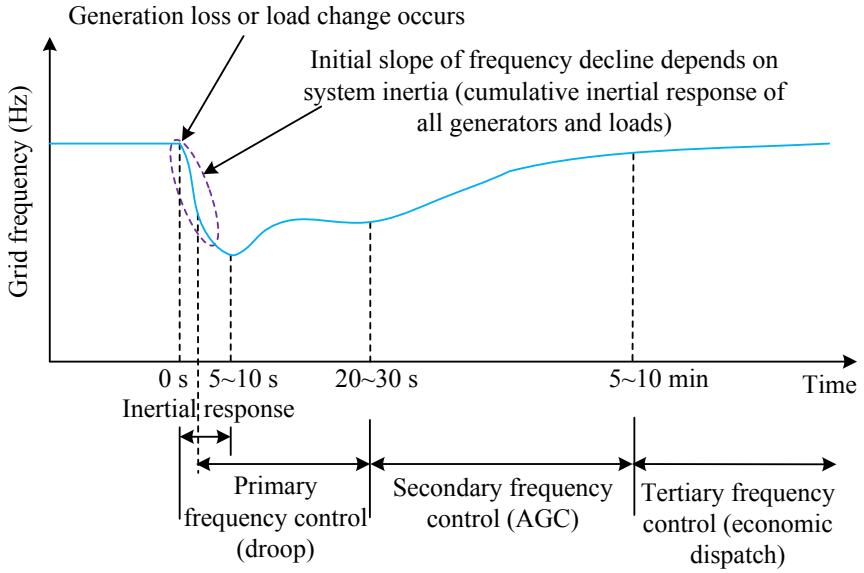



Figure 2.5 – Schematic diagram of frequency control in an AC system [Wu18]

tion 5 to 10 seconds after its appearance. The secondary frequency control corresponds to the AGC, which adjusts the reference value of the power to be generated 20 to 30 seconds after the disturbance. Finally, new reference points might be given by the tertiary frequency control, after an economic dispatch for the new operating conditions is carried out minutes later.

If the automatic frequency control fails to restore the power balance, then, security controls such as load curtailment or load shedding come into play. Similarly, when a blackout occurs due to a total collapse of the system, restorative actions need to be realized. Because emergency or restorative actions imply the disconnection from (respectively, connection to) the grid of one or several generators and/or loads, they can take several hours. In addition, the system enters an emergency situation whenever one or several lines are congested. This situation compromises the long-term power balance as a future disturbance could provoke cascading failures. Thus, the power needs to be re-dispatched through the non-congested lines as determined by the human operator. The latter needs also to manage the power exchanged with other transmission grids.

In consequence, the operation of an AC transmission system is built upon automatic control and the expertise and interpretation of the data by a human operator. Indeed, the modification of the power reference points, load-shedding and line tripping are mostly realized by automatic functions whereas dispatch management in order to ensure long-term security is mostly a human-centered operation [Zha10].

## 2.3 Control and protection of HVDC transmission systems

An HVDC transmission system connects two or more AC systems by means of HVDC links. The AC and DC systems are interfaced via a set of converter stations that transform the power

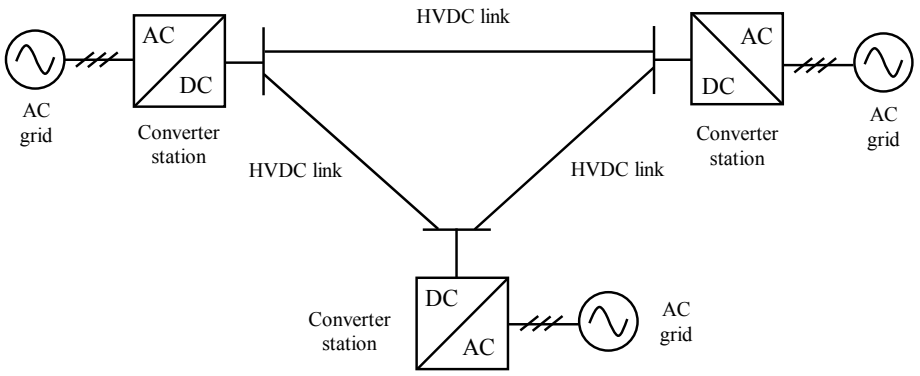



Figure 2.6 – Meshed MTDC architecture

from AC to DC and conversely by means of a power converter. Traditionally, HVDC projects were used to connect two AC grids that were separated by a large distance or worked at different frequencies. Hence, a point-to-point architecture with two converter stations connected to the concerned AC grids sufficed [Arr07]. In recent years, however, the meshed MTDC architecture (Figure 2.6) has emerged as the best solution to the massive integration of RES [Rao15; Bui17]. Despite the high cost of construction, the redundancy introduced by the existence of more than one electrical path between two converter stations enables to transfer power between them even if a link is out of service, which increases the reliability and security of the grid.

### 2.3.1 System topology

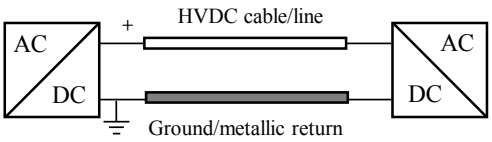
The basic topology of an HVDC link is distinguished by the converter station arrangements and earthing schemes [DeB13; Let14], which ultimately influence how the grid is operated and protected [Kon15]. A schematic view of the common topologies is shown in Figure 2.7.

The symmetric monopolar scheme consists of two HVDC conductors with the opposite voltage polarity (Figure 2.7b). The earth reference can be provided by a high-impedance reactor on the AC side and by mid-point capacitors on the DC side. Thus, there is no current flowing through earth during normal operation.

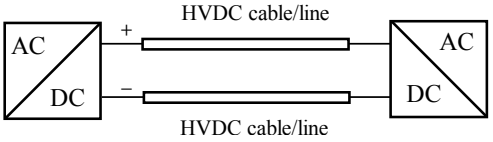
The asymmetric monopolar scheme consists of two converters connected by one HVDC conductor, with the other pole of each station connected to the earth (Figure 2.7a). If the connection is made directly to an earth electrode, the current flows freely through the conductor and the earth return path, although this type of earthing configuration is prohibited in many cases by environmental and safety constrains. In such cases, a dedicated low voltage metallic conductor connecting a common earth electrode to the pole of each station is installed.

Finally, the bipolar topology is a combination of two asymmetric monopole schemes with an opposite voltage polarity and linked to an earth reference through the mid-point of the two cascaded converter stations (Figure 2.7c). As each pole is controlled independently, if one of them is suffering from a fault or is shut down for maintenance, the remaining pole can continue

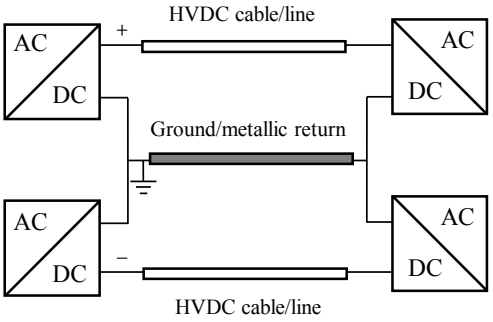
to transfer power at half of the overall transmission capacity, or more by using overloading capabilities.




(a) Asymmetric monopolar scheme




(b) Symmetric monopolar scheme




(c) Bipolar scheme

Figure 2.7 – HVDC link topologies

### 2.3.2 Subsystems in a converter station

In Figure 2.8, a functional diagram for the EMS of an HVDC system is proposed. Presently, several works in the literature have treated the functions in the Data Acquisition and Processing Subsystem [Hwa12; Cho17]. With respect to the economic considerations of an HVDC grid operation, it is to be expected that the methods already known for AC systems are adapted without major difficulties [PRO17]. In consequence, the research community is currently focused on two main aspects: the exploration of the technical capabilities of the physical components to be used, and the control of the power balance within the grid, as well as the necessary control and protection actions to be deployed whenever this balance is not maintained.

Although several works explore the capabilities of VSCs for the automatic control of the power balance in the grid [Rau14; Shi17] or develop novel protection schemes for the system components [Des13; Leó16; Lou17], there does not exist any previous work regarding the development of an Automated Supervisory Control (ASC) that monitors and coordinates the security actions instead of a human operator, given the time constraints described in Section 2.4. In consequence, the security monitoring and control subsystem of Figure 2.3 is renamed to operational monitoring and control subsystem in Figure 2.8, as the ASC is expected to assume some of the supervision functions realized by the human operator in AC systems (coordination, reference point selection capability...), in addition to the security-related actions. Hence, the human operator in HVDC systems is in charge of the tuning of the ASC and relegated to a back-up role in order to cope with the situations that cannot be managed through automatic control.

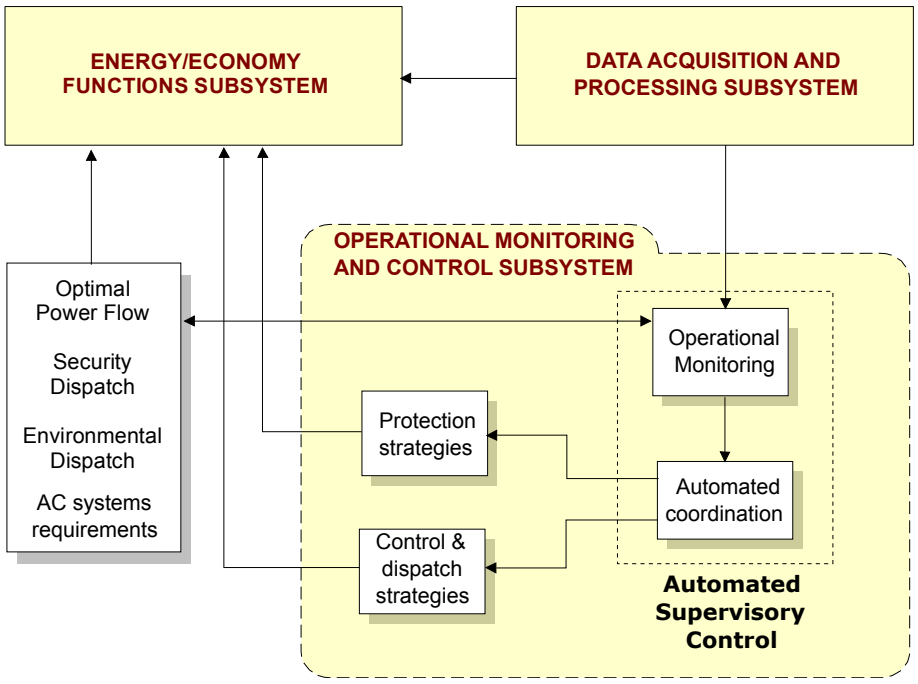



Figure 2.8 – Proposed functional diagram of an HVDC EMS

The power components and their respective controllers in the converter station that are considered to be essential for the proper functioning of the ASC can be classified into three main subsystems: the HVDC cable or Overhead Line (OHL), the power conversion subsystem and the protection subsystem. While the AC filter and the transformer at the AC side of the converter station form a subsystem that adjusts the voltage coming from the adjacent AC grid (the former by filtering the harmonics and the latter by modifying the amplitude of the AC voltage), their study is beyond the scope of this thesis. A schematic configuration of the considered subsystems is given in Figure 2.9 for the case of a symmetric monopolar link. For asymmetric monopolar or bipolar topologies, one of the DC poles is directly connected to the ground (cf. Figure 2.7).

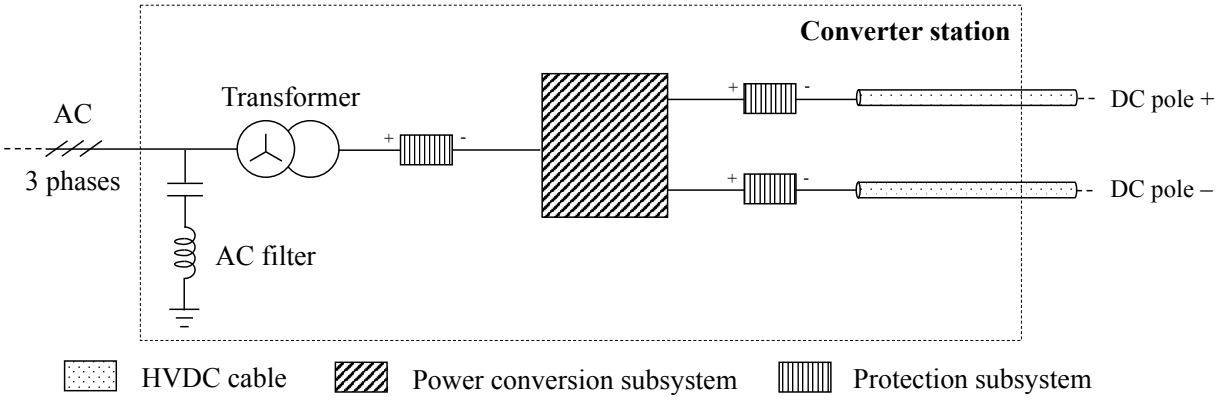
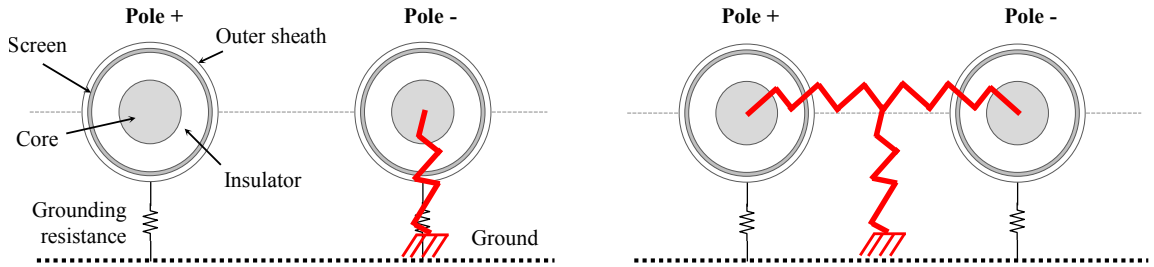



Figure 2.9 – Converter station configuration for a symmetric monopolar scheme




(a) Pole-to-ground (b) Pole-to-pole

Figure 2.10 – Short-circuit fault types

### HVDC cable

The role of the HVDC cables or OHLs is to effectively transfer the power from one converter station to another. In consequence, it is crucial for the security control system to measure their current and voltage in order to estimate the state of the grid in terms of power balance. Although transmission lines are usually laid over hundreds of kilometers, the measurement devices are local to each converter station. For future MTDC grids, underground or submarine cables based on Cross-Linked Polyethylene (XLPE) insulation technology have been retained, as they have higher mechanical strength, a lower environmental risk compared to the classic cables and the ability to support higher temperatures [Ame15]. This type of cables consist of two main conductors: the core that carries the DC current and is surrounded by an insulation layer, and a screen surrounded by an outer sheath. The screens of the cables on the positive and the negative poles of each converter station are connected to the ground via the material in which the cables are buried. This material has a grounding resistance of low impedance.

However, short-circuit faults occur whenever the insulation layer between both conductors in the cable breaks. Because the screen of the cable is grounded through a very low impedance, the potential in the core is forced to be almost zero, leading to a collapse of the grid's DC voltage. As a short-circuit fault in an HVDC system is characterized by a rapid propagation of the fault along the grid, a high current of several tens of kiloamps flows into the grid from the surrounding AC systems. In consequence, the power balance is disrupted and eventually the rest of the components in the converter stations will also be damaged if the faulty cable is not isolated, as they are not able to tolerate the overheat caused by the high current. It is thus necessary that the designed supervisory control coordinates both the protection actions so as to prevent further damage on the healthy components of the grid and the control and dispatch actions so as to restore the power balance once the short-circuit fault is isolated from the rest of the system. Two types of short-circuit faults can be distinguished: the pole-to-ground faults when the insulation break takes place in a pole (Figure 2.10a), and the pole-to-pole (and to ground) faults whenever the insulation layer is fissured in both cables (Figure 2.10b), although this last case is very rare and is thus not considered in the following [DeB13].

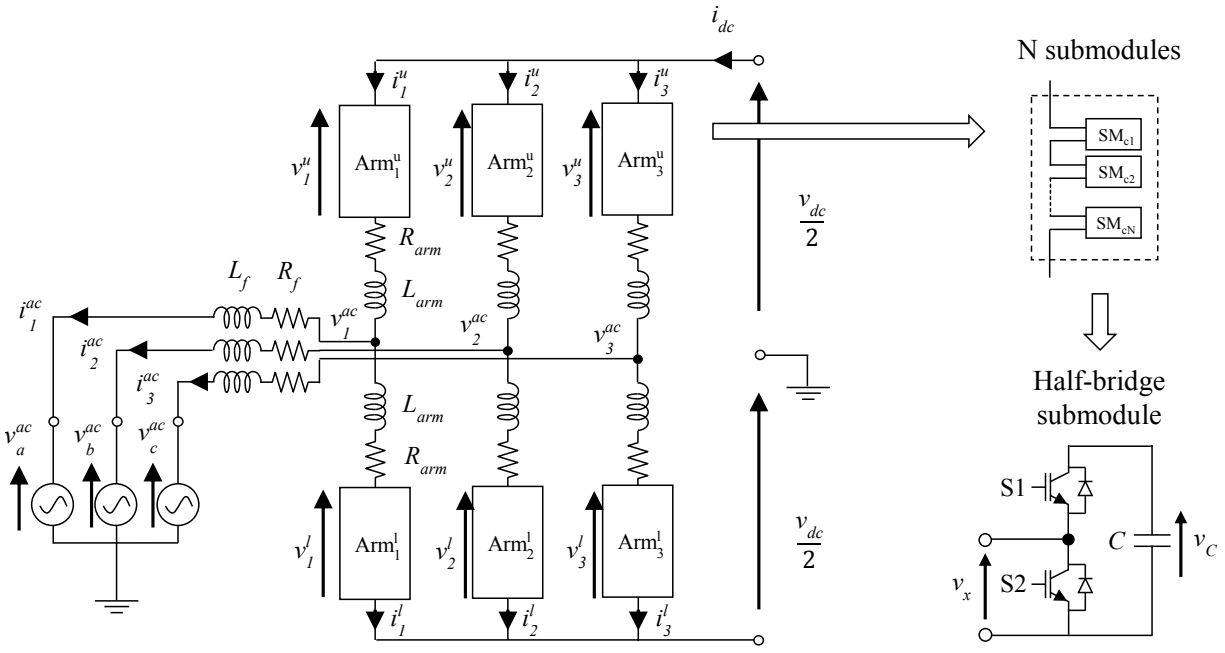



Figure 2.11 – Configuration of an MMC

### Power conversion subsystem

The power conversion subsystem comprises the power converter and the associated continuous-time controllers, both necessary for the transformation of power and for the regulation of the power balance within the grid. In recent years, the development of a new generation of VSCs, based on the Modular Multilevel Converter (MMC) topology [Les03], has made for an important push towards the creation of large MTDC grids. The MMC inherits the capability of power flow reversal and independent control of active and reactive power from earlier VSCs while offering better power rating scalability and an excellent harmonic performance in comparison.

The structure of the MMC is shown in Figure 2.11. An MMC consists of three phases, also called legs. The middle of each leg is connected to the AC side of the converter station to one of the three phases of the adjacent AC grid. Furthermore, each phase is formed by an upper and a lower arm, which are composed of a set of identically designed Submodules (SMs). The arms in each leg are connected to the HVDC system through the positive and negative poles of the converter station, depending on the system configuration (see Figure 2.7). Finally, the inductors $L_{arm}$ and $L_f$, respectively found in each arm and each AC phase, are used to smoothen the variations of the current.

Although several SM topologies are proposed in the literature [Deb15], all of them are designed such that there is a floating DC capacitor acting as a switchable voltage source and some switching devices. The most common SM topology, called half-bridge, consists of two Insulated-Gate Bipolar Transistors (IGBTs) with anti-parallel diodes and a DC capacitor previously charged to their rated voltage $v_c$. Then, by acting on the control signals $S1, S2$ of the

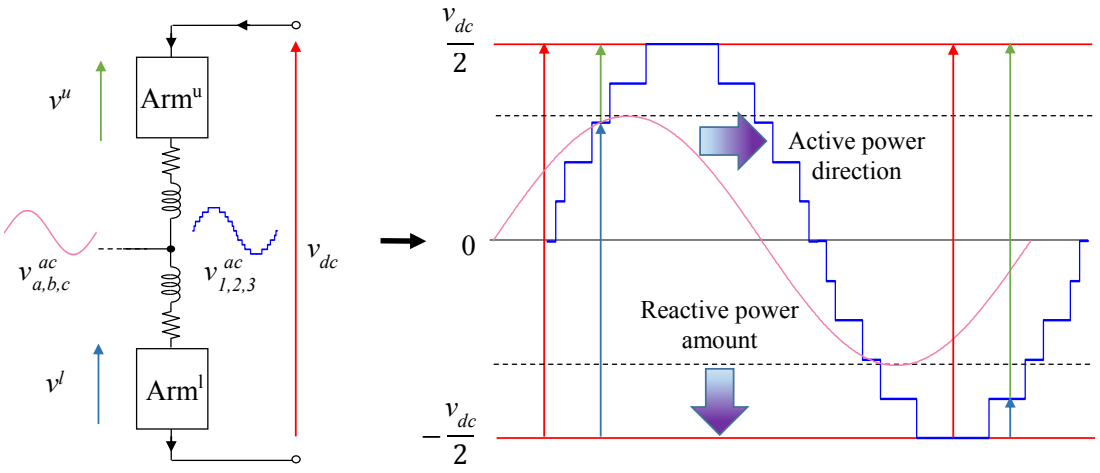



Figure 2.12 – Operation principle of the MMC (illustrated for one phase)

IGBTs, the voltage of the capacitor can be either inserted or bypassed, determining the voltage $v_x$ at the output of the submodule to be either zero or $v_c$ in consequence. However, because the typical rated voltage supported by commercial IGBTs is of 3.2 kV, it is necessary to stack a large number of SMs in series in each arm in order to operate at high voltage. In consequence, although the economical cost of the MMCs is increased with respect to traditional VSCs, its modular architecture provides great reliability and maintainability as the converter can continue its operation during the failure of a few SMs [Les03].

Then, the MMC is operated by using the capacitors $C_j$ $(j = 1,\ldots,N)$ in the arms' SMs as voltage sources to synthesize both a DC and an AC voltage in each leg. The DC voltage is given by the sum of all the SM voltages inserted in the leg between the positive and negative DC poles while the AC voltage is synthesized by varying the proportion of the number of inserted SMs in the two arms, as illustrated in Figure 2.12. The sine wave created by the MMC in each leg is then compared to one of the corresponding phase of the adjacent AC grid. While the phase difference between both waveforms determines the direction of the active power (i.e. from the MMC to the AC grid or vice versa); the amplitude difference determines the amount of reactive power transmitted by the MMC to the AC grid. On the DC side of the converter, the sum of the voltages of the inserted capacitors in each leg creates the DC voltage, which it needs to be around the nominal voltage $v_{dcn}$ in all MMCs (cf. Section 2.4), while the sum of the currents in the three legs determines the amplitude and direction of the DC current. In consequence, the amplitude and direction of the DC power depends on the DC current, as the DC voltage remains nearly constant.

Yet, for the MMC to be able to modify the power transmitted through the station, the arms should be previously charged to a voltage level at least equal to the peak value of the AC grid's sinewave so that active power regulation can be accomplished. In order to be able to regulate the reactive power, however, the arms are typically charged until the available voltage reaches

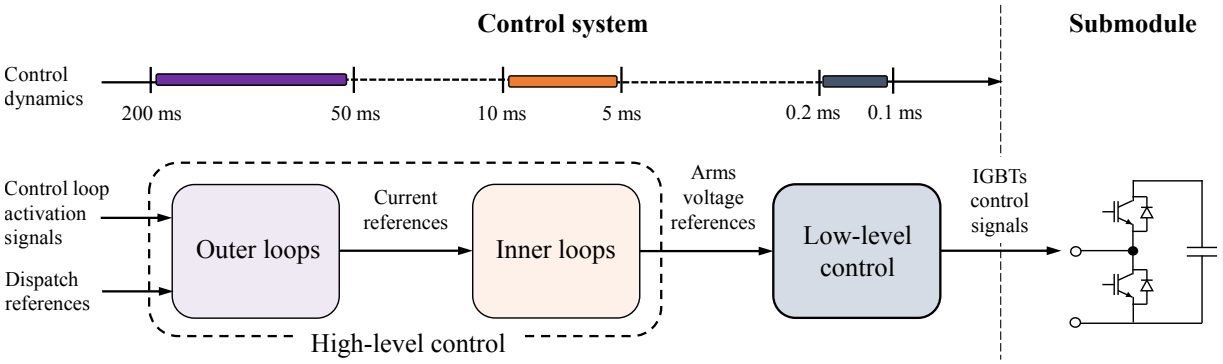



Figure 2.13 – Global control structure of the MMC with corresponding response times

the rated level so as to create an AC-side waveform of an amplitude bigger than the peak value of the AC grid's voltage. Thus, the instantaneous total voltage available in the MMC, denoted by $v_{mmc}$, is calculated as the average of the available voltage in the six arms and is considered to be such that $v_{mmc} \approx v_{dcn}$. Therefore, the MMC can be regarded as an energy buffer that stores an electrostatic energy $E_c$ of the form:

$$E_c = \frac{1}{2} C v_{dcn}^2, \tag{2.1}$$

with

$$C = 6C_{arm} = 6\frac{1}{\sum_{j=1}^{N} \frac{1}{C_j}}. \tag{2.2}$$

Hence, the continuous-time control of the MMC, schematized in Figure 2.13, plays a fundamental role in the operation of the system. According to [Shi17], the control of the MMC is divided into two main levels: the high-level control and the low-level control. In turn, the high level control is decomposed into the inner control loops and the outer control loops. The outer loops deal with the regulation of variables that affect the HVDC system as a whole: active power, reactive power and DC voltage, mainly, but also the energy stored in the MMC. They provide the inner loops with the references so that the latter can achieve fast control of the AC, DC and arm currents in each leg. The low-level control receives then the voltage references of each arm from the inner loops in order to insert the appropriate SMs in each arm, so as to maintain the voltage in all the capacitors around $v_c$ while creating the AC and DC voltages dispatched by the control center to the outer loops (see Figure 2.8).

However, depending on the situation of the system, different control methods can be appropriate for the same control loop in the outer loops. Also, in case of a short-circuit fault, the continuous-time control of the MMC should be deactivated until the fault is isolated and the healthy cables reconnected. Otherwise, the inability to restore the power balance in the grid

could lead to a failure of the MMC due to large variations of the stored energy. In consequence, for a secure operation of the system, the supervisory control needs to: either activate the appropriate outer loops so as to maintain or restore the power balance in the grid, or deactivate the controllers in order to protect the MMC when required by the situation.

### Protection subsystem

As stated above, the cable suffering a short-circuit fault needs to be isolated from the rest of the grid in order to prevent additional damage on the rest of the components. On that account, the protection subsystem identifies the faulty cable in order to isolate it. The configuration of the protection subsystem is determined by the protection strategy to be implemented, which in turn depends on the identification algorithm and the power device used for fault current interruption [Let15]. According to [Lou17], the protection strategies follow two general approaches, namely the selective approach and the non-selective approach.

In the selective approach, the faulty cable is selectively isolated by means of fast detection algorithms and the fault current is limited through cutting-edge technology that withstands temporarily the overcurrent [Leó16]. In consequence, the rest of the cables in the grid are not affected by the fault, thus preventing the DC voltage from collapsing and a new power balance is reached through automatic power control. However, the fault current limiting devices necessary for such strategies to be effective have a high cost and are not yet fully mature. Non-selective strategies, on the other hand, use well-proven apparatus for the protection of AC systems. Unlike the selective approach, non-selective strategies do not prevent the fault current from spreading throughout the grid. Hence, all the HVDC cables need to be disconnected from the converter stations until no more current is injected into the grid from the adjacent AC systems. In the meantime, the faulty cable is identified and then isolated, while the healthy cables are reconnected [Lou17]. However, because the entire HVDC grid is impacted, the DC voltage collapses and the power balance is lost. In consequence, it is necessary to quickly restore the power flow in the grid, as the loss of an HVDC system can compromise the stability of the adjacent AC grids after a critical time [Gon18].

In the following, the configuration of the protection subsystem used in [Lou17] is presented. The protection devices necessary to achieve the protection strategy are assembled into modules, such as the Pre-Insertion Resistor (PIR) module and the Breaking Module (BM) shown in Figure 2.14, which are located at each point connecting the station to the AC and DC grids (see Figure 2.9). A BM is composed of a set of devices aiming to isolate the grid components they are associated to: voltage and current sensors, a Circuit Breaker (CB) and a High-Speed Switch (HSS) for current suppression and short-circuit isolation, and finally a protective relay that commands the previous devices. A modern protective relay is a microprocessor-based Intelligent Electronic Device (IED) that acts as the controller of the BM equipment (CB, HSS) and can receive external requests and communicate, for instance, with the ASC. The relay mon-

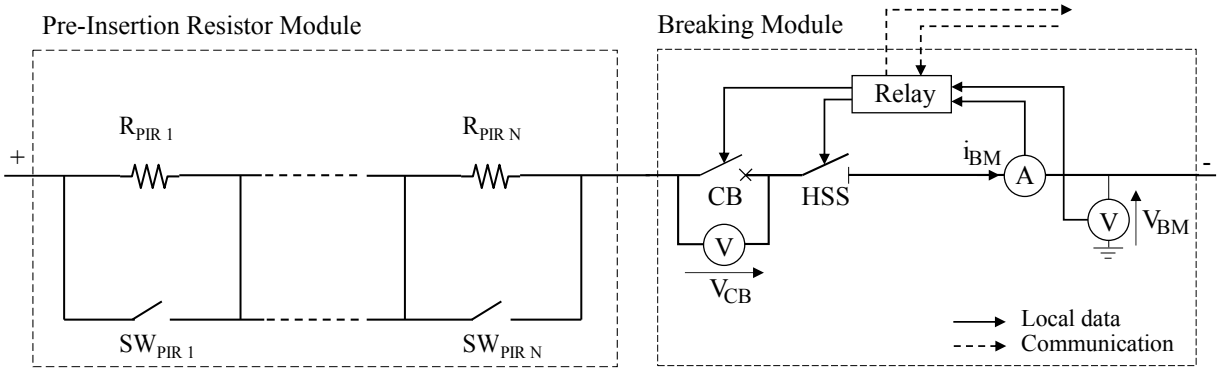



Figure 2.14 – Configuration of a BM and a PIR module

itors the measurements of the cable going through the associated module to detect abnormal situations. Each protective relay includes an identification algorithm that detects the location of the fault and generates the tripping signals sent to the CB and the HSS, which separate the electrical circuit of the converter station from that of the AC or HVDC grid upon reception of the tripping order. Typically, a CB is capable of breaking the circuit up to a fault current of around 20 kA, whereas the HSS is not capable of interrupting a non-zero current. Thus, HSS are commonly used in case the associated CB fails to open. When the BM is located on the DC side of the station, we refer to the CBs as DCCBs, while their AC counterpart are referred as ACCBs. While the objective of an AC and a DC side BM are identical, the CB technology might differ and the HSS could be absent from the AC side BM.

Finally, since the voltage in the cables collapses to zero during the short-circuit fault, it is necessary to restore it to its rated value once the damaged cable is isolated. In order to limit the subsequent inrush current drawn by the HVDC cables when reconnected, a PIR module is used [Des13]. This ancillary device is composed of one or several series resistors, each in parallel with a bypass switch, which are commonly commanded by the relay of the associated BM. Then, all the resistors are inserted at first to be progressively shunted later on, until the rated DC voltage level is reached. In addition, although the PIR module limits the peak value reached by the inrush current, it prolongs the charging time of the cables as it multiplies the resistance of the grid. Hence, the impedance of the module can be optimized so that time and security requirements are met.

## 2.4 Comparison between AC and DC dynamics

### AC systems

In AC systems, electric power is mainly generated by synchronous rotating machines connected to the transmission grid. The electromechanical dynamics of a turbo-generator are de-

pendent on its stored rotational energy, denoted by $E_r$, which is defined as a function of the moment of inertia $J$ [kg · m$^2$] and the angular speed $\omega = 2\pi f$ [rad/s][1]:

$$E_r = \frac{1}{2} J\omega^2. \tag{2.3}$$

As a rule, the motion of the generator's rotor determines the variation of rotational energy and is governed by the difference between the mechanical input power $P_m$ [W] and the electrical output power $P_e$ [W]. Hence, the swing equation governing the motion of the rotor is derived from $E_r$ as follows:

$$\frac{dE_r}{dt} = \frac{d\left(\frac{1}{2} J\omega^2\right)}{dt} = P_m - P_e, \tag{2.4}$$

thus

$$\frac{d\omega}{dt} = \frac{P_m - P_e}{J\omega}. \tag{2.5}$$

Then, supposing a linearization around the operating point where $\omega = \omega_n$, the swing equation can be rewritten as follows:

$$\frac{d\omega}{dt} = \frac{P_m - P_e}{J\omega_n}. \tag{2.6}$$

As the generators rotors are large rotating masses, their inherent inertia offers a resistance to a change in the state of motion, i.e. to a variation of the angular frequency. In order to compare the inertial effect of synchronous generating units with different ratings, the inertia constant $H$ [s] of a generator is defined as the ratio between its stored rotational energy $E_r$ and its rated apparent power $S_b$ [MVA] [Kun94]. Thus, the inertia constant can be interpreted as the time during which a generator can keep supplying its rated power with its stored energy alone [Ulb14], e.g. in case of a disconnection from its driving turbine. It typically ranges from 2 to 10 seconds, depending on the type of the generating unit [Kun94].

The swing equation can then be rewritten in the per unit system ($\bar{x}$):

$$\frac{d\bar{\omega}}{dt} = \frac{\bar{P}_m - \bar{P}_e}{2H}, \tag{2.7}$$

with $\bar{\omega} = \frac{\omega}{\omega_n}$, $\bar{P}_m = \frac{P_m}{S_b}$ and $\bar{P}_e = \frac{P_e}{S_b}$.

Because the kinetic energy stored in the whole AC system is equivalent to the sum of all the kinetic energies stored in each generator composing the grid, the inertia $H_{sys}$ of the whole AC

[1] The relationship between the voltage frequency $\omega$ and the mechanical rotating speed of the generator $\Omega$ is given by $\omega = p * \Omega$, where $p$ is the number of pole pairs of the generator. To simplify the study, we assume the number of pole pairs to be one ($p = 1$) so that $\omega = \Omega$.

system is expressed as follows:

$$H_{sys} = \frac{\sum_{i=1}^{n} H_i S_{b_i}}{\sum_{i=1}^{n} S_{b_i}}. \tag{2.8}$$

Considering that all the generators in the grid need to be synchronized around the nominal frequency $\omega_n$, the swing equation of a generator can be extended to an aggregated expression of the whole system [Tie16]:

$$\frac{d\bar{\omega}}{dt} = \frac{\bar{P}_g - \bar{P}_l}{2H_{sys}}, \tag{2.9}$$

with $\bar{P}_g = \frac{\sum_{i=1}^{n} P_{m_i}}{\sum_{i=1}^{n} S_{b_i}}$ and $\bar{P}_l = \frac{\sum_{i=1}^{n} P_{e_i}}{\sum_{i=1}^{n} S_{b_i}}$.

As a result, the frequency of the system acts as a global indicator of the instantaneous balance between the total generated power $P_g$ and the total load power $P_l$ of the system. When this balance is not respected, the system frequency fluctuates and deviates from its nominal value. However, because of the global inertia $H_{sys}$ of the system, the variation of the frequency can be arrested in a first instance, as the grid is capable of maintaining the rated power for a certain time until the speed of each generator is regulated in order to compensate for the frequency variation (and eventually reestablish the frequency to its nominal value). The typical value of the system inertia constant of the continental European grid is around 5 to 6 seconds [Uni04].

### DC systems

The power converters used in HVDC systems are classified into two main types: Line-Commutated Converters (LCCs) and Voltage Source Converters (VSCs). Because VSCs allow to easily change the current direction while maintaining the DC voltage polarity, they are more appropriate for MTDC grids with constantly changing power flows [Her10]. Due to the capacitive behavior of the VSCs and the cables, the energy in the HVDC grid is stored in the form of electrostatic energy, denoted by $E_c$, which is defined as a function of the grid's capacitance $C$ [F] and the grid's DC voltage $v_{dc}$ [V]:

$$E_c = \frac{1}{2} C v_{dc}^2. \tag{2.10}$$

Analogously to AC systems, the electrostatic inertia constant, denoted by $H_c$, is then defined as the ratio between the stored electrostatic energy $E_c$ and the rated apparent power $S_b$ of the grid. According to [Jac10], the typical value of $H_c$ for new generation VSCs is around 40 milliseconds. Because the electrostatic energy stored in the high-voltage links is negligible [Rau14], the energy stored in the whole DC system is equivalent to the sum of all the energies stored in each converter, and so the inertia $H_{c\,sys}$ of the whole DC system is expressed similarly to its AC counterpart. As stated in [Shi17], a reasonable approximation of $H_{c\,sys}$ can be deduced

to be close to the individual inertia constant of each converter.

Also, although the DC voltage might not be equal at two different converter stations because of the voltage drop along the resistance in the high-voltage conductors, $v_{dc}$ is considered to be common to all stations in practice, as the voltage drop represents less than 1% of the total value [Hau08]. Then, considering a linearization around the nominal voltage $v_{dcn}$, the swing equation of the DC system can be obtained in the per unit system as [Ben15]:

$$\frac{d\bar{v}_{dc}}{dt} = \frac{\bar{P}_{rec} - \bar{P}_{inv}}{2H_{csys}}, \tag{2.11}$$

with $\bar{v}_{dc} = \frac{v_{dc}}{v_{dcn}}$, $\bar{P}_{rec} = \frac{P_{rec}}{S_b}$ and $\bar{P}_{inv} = \frac{P_{inv}}{S_b}$.

As a result, the DC voltage $v_{dc}$ plays a role analogue to that of the frequency in AC systems, as it serves as a global indicator of the balance between the total power injected into the DC system by the rectifier converters, denoted by $P_{rec}$, and the total power withdrawn from the DC system by the inverter VSCs, denoted by $P_{inv}$. Thus, a power imbalance leads to a fluctuation of the DC voltage level [Ren16]. The power balance of the DC grid should then be reestablished by means of DC voltage control. However, because of the lower inertia of DC systems and the lower number of stations, the DC voltage is far more volatile than the angular frequency.

### Comparison

The different analogies made between AC and DC systems are summarized in Table 2.1. From the given data it can be implied that, for the same power disturbance (in the per unit system), the rate of change of the DC voltage would be around 100 times more important than that of the frequency in an AC grid, increasing in consequence the constraint on the response time of the control system. Whenever the voltage control fails to reestablish the power balance and the DC voltage collapses, security controls must be initiated so that the converters are still capable of restoring the power flow in the grid. While in AC systems, such control actions could last for several hours or days, in HVDC systems a power balance condition must be restored within a short time range (hundreds of milliseconds), otherwise the collapse of the DC grid would impact negatively the neighboring AC systems, which is unacceptable [Abe17; Gon18]. In particular, given the low inertia of the grid, a collapse of the DC voltage is rapidly propagated throughout the network, triggering an almost simultaneous response of several control and protection schemes all over the system, which may cause additional failures because of the possible negative interactions. In consequence, because of the fast response that is required, a human operator cannot intervene, as it is the case of AC transmission systems. Furthermore, local control alone does not suffice to bring back the system to a secure operation and the security control actions applied by the different converter stations should be coordinated.

However, although the differences between the dynamics of both systems justifies the need for a more automated operation of HVDC system, the operational structure of AC systems, largely

| System | AC | DC |
|---|---|---|
| Common variable | Angular frequency | DC voltage |
| | $\omega$ | $v_{dc}$ |
| Stored energy | Kinetic energy in rotating mass | Electrostatic energy in capacitor |
| | $E_r = \frac{1}{2} J\omega^2$ | $E_c = \frac{1}{2} C v_{dc}^2$ |
| Inertia or electrostatic constant | Generator | Converter |
| | $H = 2 \sim 10$ s | $H_c = 40$ ms |
| System inertia or electrostatic constant | Continental Europe | Reasonable value |
| | $H_{sys} = 5 \sim 6$ s | $H_{c\,sys} \approx 40$ ms |
| Swing equation (per unit) | $\frac{d\bar{\omega}}{dt} = \frac{\bar{P}_g - \bar{P}_l}{2H_{sys}}$ | $\frac{d\bar{v}_{dc}}{dt} = \frac{\bar{P}_{rec} - \bar{P}_{inv}}{2H_{c,sys}}$ |

Table 2.1 – Analogy between AC and DC systems [Shi17]

based on the Purdue Enterprise Reference Architecture [Ber96], is certainly adequate for HVDC systems given the similarities between the hierarchical control and protection structure of both systems and their distributed physical nature.

## 2.5 Conclusion

To conclude, while the coordination of the security control actions might rely in certain cases on the decision making of human operators in AC systems, this is no longer possible in HVDC systems, which are required to be restored after a fault within a time range that prohibits human intervention. It is thus necessary to develop an ASC that substitutes the human operator for real-time monitoring and component coordination. The operational requirements of an HVDC transmission system, like for traditional AC power systems, are met through the management of the interactions between discrete-event and continuous-time dynamics for each of the control and protection subsystems integrated in the hierarchical operational structure.

In consequence, it is the objective of this PhD thesis to propose a systematic method for the synthesis and implementation of a supervisory control for HVDC systems, based on the terminology of DES and the use of the formal methods within the SCT framework, which allow to obtain a set of coordinated actions that respect some predefined security specifications.

# 3

# Formal framework

## Contents

## 3.1 Introduction

The goal of this chapter is to present and define the different concepts and methods that are to be used in the following for the synthesis of an automated supervisory control in view of its implementation for HVDC applications. Thus, the concepts of formal language and automaton are first presented. The main notions and methods of the SCT are defined then, so as to manipulate the modeled automata in order to synthesize a supervisor automaton. Different control architectures in which the obtained supervisor and the associated automata can be arranged are detailed and a quick overview of the existing implementation methods is given. Finally, the most appropriate methods for our work are retained.

## 3.2 Language theory and automata

Language theory originally emerged from linguistics as a way to understand the syntactic regularities found in natural languages and characterize them according to a set of formal properties. Thus, formal grammars provide logic rules for the construction of languages which can be mathematically manipulated. Depending on the restrictions imposed on formal grammars, several types of languages can be generated [Cho59]. In this work, only those languages that can be represented by an FSA (also called Finite-State Machine (FSM)), are considered.

### 3.2.1 Basic concepts

A language is a collection of sentences, all constructed from a finite alphabet of symbols [Cho59]. Thus, a *symbol*, depicted by $\sigma$, is the element from which languages are built. Given that the smallest unit of a writing system of any language is the grapheme, the characters referred to by this term are generally considered to be symbols: letters, numbers, punctuation marks, logograms, graphical signs, etc. Symbols are grouped in a finite set called *alphabet*, which is commonly denoted by $\Sigma$. A sentence or *string*, denoted as $s$, is then formed by the *concatenation* (cf. Definition A.1) of symbols included in the alphabet, i.e. $s = \sigma_1\sigma_2\ldots\sigma_n$. All the possible strings formed by the concatenation of symbols $\sigma_i \in \Sigma$ $(i = 1,\ldots,n)$ are included in the infinite set $\Sigma^*$, which is derived by the operation called Kleene-closure (*): $\Sigma^* = \{\varepsilon, \sigma_1, \sigma_2, \sigma_3, \sigma_1\sigma_2, \sigma_1\sigma_3, \ldots\}$, where $\varepsilon$ is the empty string.

The concatenation of two strings $u$ and $v$ such that $u = u_1u_2\ldots u_n$ and $v = v_1v_2\ldots v_m$ is the string denoted $uv$ equal to $u_1u_2\ldots u_nv_1v_2\ldots v_m$, which is obtained by simple juxtaposition. This operation is associative and non-commutative. When a string $s$ is constituted by the concatenation of three other strings $t,u,v$, such that $s = tuv$, the following terminology is used: $t$ is called a *prefix* of $s$, $u$ is called a *substring* of $s$ and $v$ is called a *suffix* of $s$. It should be noted that both $\varepsilon$ and $s$ are prefixes, substrings and suffixes of $s$.

A language $L$ is a set of finite-length strings formed from symbols in $\Sigma$. Therefore, $L$ is a subset of $\Sigma^*$ ($L \subseteq \Sigma^*$), where $\Sigma^*$ consists of all the feasible strings over an alphabet $\Sigma$. Hence, the different possible concatenations of events (i.e. the symbols) of a DES describe all the possible event-driven behaviors (i.e. the language) of the system. Since languages are mathematically considered as sets, common set operations, such as union, intersection, complement and Cartesian product, are applicable to languages. In addition, it is also possible to apply the operations defined in Appendix A, namely: concatenation, prefix-closure, Kleene-closure, projection and inverse projection. These operations constitute a significant part of the fundamental functions necessary for the manipulation and construction of languages and can be found in various works such as [Cas08; Won17].

### 3.2.2 Finite-state automata

Regular languages are defined to be the languages that are generated by Type-3 grammars in the Chomsky hierarchy. Their particularity is that they are representable by an FSA, one of the simplest models that exist to perform calculations with a finite memory [Sip06]. If a unique outcome is produced for each string of symbols that is accepted or rejected by a finite-state automaton, we talk about Deterministic Finite Automaton (DFA). In the following, we shall use the term automaton when referring to DFA. Formally, an automaton is defined as follows:

***Definition 3.1 (Automaton)***
*An automaton $G$ is defined by a quintuple $G = (Q, \Sigma, \delta, q_0, Q_m)$ such that:*

- *$Q$ is a finite set containing all the states of $G$;*
- *$\Sigma$ is the finite set of events, also called alphabet, associated with $G$;*
- *$\delta$ is the partial transition function, defined by $\delta : Q \times \Sigma \to Q$. $\delta(x, \sigma) = y$ represents a unique transition labeled by event $\sigma$ from state $x$ to state $y$;*
- *$q_0$ is the initial state of automaton $G$, such that $q_0 \in Q$;*
- *$Q_m$ is the set of marked states of $G$, such that $Q_m \subseteq Q$.* ♦

Considering as an example a regular language $L$ over an alphabet $\Sigma = \{\alpha, \beta, \mu, \gamma\}$ denoted by the regular expression $L = (\alpha\beta + \alpha\gamma\mu)^*(\alpha\gamma + \alpha + \varepsilon)$, such a language is generated by an automaton that can be graphically represented by the state-transition diagram illustrated in Figure 3.1. This automaton describes the behavior of a device and has three states: ($I$)dle, ($W$)orking and ($F$)ailure. The state $I$ is the initial state of our automaton. It is symbolized graphically by an incoming arrow into the state that does not come out of any other state. State $I$ is also considered to be the only marked state of the device ($Q_m = \{I\}$). Graphically, a marked state is represented by a second circle surrounding the state. Typically, a marked state symbolizes an end of task or a goal to be reached: the device stops working at the end of a task.

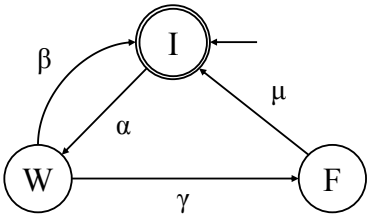



Figure 3.1 – State-transition diagram of an automaton

Hence, each symbol in $\Sigma$ is the label associated to a discrete event describing the evolution of the device. From state $I$, an outgoing arrow labeled by the event $\alpha$ enters the state $W$. This arrow represents the transition $\delta(I,\alpha) = W$, indicating the start of a task. The set of arrows therefore represents the transition function $\{((I,\alpha),W),((W,\beta),I),((W,\gamma),F),((F,\mu),I)\}$, of which the elements mean respectively: start of task, end of task, occurrence of a failure and repair of the device. In this example, the automata of the different devices of a converter station can be combined and manipulated through the operations defined in Appendix B, which can be also found in [Cas08; Won17].

Thus, the particularity of an automaton is to recognize, or generate, a language. The marked language which contains all the strings leading to a marked state is included in this language. Formally, this conduces to the following definitions:

***Definition 3.2 (Languages generated and marked)***
*The language generated by* $G = (Q,\Sigma,\delta,q_0,Q_m)$ *is defined by:*

$$L(G) = \{s \in \Sigma^* | \delta(q_0,s) \text{ is defined}\}$$

*The language marked by G is defined by:*

$$L_m(G) = \{s \in L(G) | \delta(q_0,s) \in Q_m\}$$ ♦

The above definitions assume that the transition function is extended such that $\delta : Q \times \Sigma^* \to Q$.

An algorithm transforming a given deterministic finite automaton into a regular expression was proposed by S.C. Kleene in [Kle56, p. 37-40]. Therefore, the Kleene's theorem, formulated in [Hop79], establishes the equivalence between regular languages and finite-state automata.

***Theorem 3.1 (Kleene's theorem)***
*A language L is said to be regular if and only if there exists a finite-state automaton G such that* $L = L(G)$. ♦

Furthermore, because finite-state automata have finite memory, the language generated by an automaton is prefix-closed by definition: $L(G) = \overline{L(G)}$ (cf. Definition A.2). Then, from definitions 3.1 and 3.2, we have:

$$L_m(G) \subseteq \overline{L_m(G)} \subseteq L(G)$$

The first set inclusion is due to the fact that $Q_m$ may be a proper subset of $Q$, while the second set inclusion is a consequence of the definition of $L_m(G)$ and the fact that $L(G)$ is prefix-closed by definition. This second set inclusion implies that an automaton $G$ can reach a state $q \notin Q_m$ with no transition going out of the state. Such a situation is called a *deadlock* because no further event can be executed. Thus, if deadlock happens, $L_m(G)$ is a proper subset of $L(G)$, since any string in $L(G)$ that ends at state $q$ cannot be a prefix of a string in $L_m(G)$. This issue can be extended to a set of unmarked states in $G$, which are reachable from one another, but with no transition going out of the set. This situation is called a *livelock*. In both the deadlock and the livelock, the system can never complete the task started since no state in the set is marked and the system cannot leave this set of states. Therefore, the automaton modeling the system is blocked. Formally, a blocking automaton is defined as follows:

***Definition 3.3 (Blocking automaton)***
*Consider an automaton $G$ that generates the language $L(G)$ and marks the language $L_m(G)$. Such an automaton is said to be blocking if:*

$$\overline{L_m(G)} \subset L(G)$$

*inversely, it is non-blocking if:*

$$\overline{L_m(G)} = L(G) \qquad \blacklozenge$$

Hence, an automaton is blocking if there exists a string of $L(G)$ which is not the prefix of a string allowing to reach a marked state.

## 3.3 Supervisory Control Theory

Because DES are finite systems, their physical behavior can be modeled by a regular language that is generated by an automaton (cf. Theorem 3.1). It is often necessary, however, to modify the behavior of the system by feedback control in order to achieve a given set of specifications or to avoid an undesired blocking state. Thus, the SCT, initiated by Ramadge and Wonham [Ram87; Ram89], offers a formal framework for the synthesis of a supervisor ensuring that the system under control respects a set of behavioral specifications, imposed by the designer, within its physical limitations.

### 3.3.1 Plant and supervisor

Two main models are distinguished within the SCT: the plant model and the supervisor model.

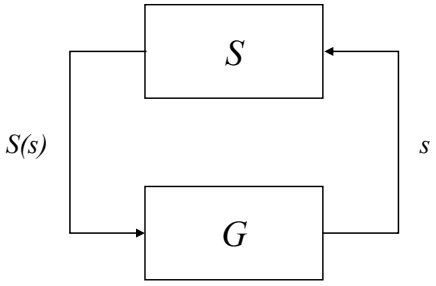



Figure 3.2 – Feedback loop of supervisory control

The plant is modeled by an automaton $G$ that generates the regular language $L(G)$ and marks $L_m(G)$. This model represents all the possible behaviors of the considered system. Therefore, the objective is to control this plant by prohibiting certain evolutions that are assumed to be inadequate. This amounts to forbidding the occurrence of certain events. However, unpreventable events, such as events related to a fault, cannot be forbidden. In consequence, the set of events $\Sigma$ is partitioned into two disjoint sets: the set of controllable events $\Sigma_c$ and the set of uncontrollable events $\Sigma_{uc}$. Hence, $\Sigma = \Sigma_c \cup \Sigma_{uc}$ and $\Sigma_c \cap \Sigma_{uc} = \emptyset$. A DES is often decomposable into $n$ elementary plants. Each of these plants has sufficiently simple behavior to be modeled by an automaton $G_i$ $(i = 1, \ldots, n)$ with only a few states, transitions and events. The global plant is then constructed from the combination of these elementary models using the parallel composition (see Definition B.5). Thus, the behavior of the different components is synchronized when they share common events while allowing occurrences of unshared events. The alphabet $\Sigma$ of the obtained automaton is the union of the alphabets $\Sigma_i$ of the plants $G_i$.

The supervisor $S$ is a function associated to the plant to prohibit undesired behaviors by disabling the occurrence of certain (controllable) events. This function is defined by:

$$S : L(G) \rightarrow 2^{\Sigma}.$$

Thus, for each string $s$ generated by $G$, the supervisor $S$ returns the set of all permitted events $S(s)$. Then, the plant $G$ can only evolve by generating an event included in this set (Figure 3.2). Hence, the pair $(S, G)$ represents the behavior of the controlled system. This closed-loop system, denoted $S/G$ (read as "$S$ controlling $G$"), generates the language $L(S/G)$ and marks the language $L_m(S/G)$. These two languages are determined by the Definition 3.4 and have the following inclusion property: $\emptyset \subseteq L_m(S/G) \subseteq \overline{L_m(S/G)} \subseteq L(S/G) \subseteq L(G)$.

***Definition 3.4 (Languages generated and marked by $S/G$)***
*The language $L(S/G)$ is defined recursively as follows:*

1. $\varepsilon \in L(S/G)$
2. $[(s \in L(S/G))$ and $(s\sigma \in L(G))$ and $(\sigma \in S(s))] \Leftrightarrow [s\sigma \in L(S/G)]$

*The marked language $L_m(S/G)$ is defined by: $L_m(S/G) = L(S/G) \cap L_m(G)$.* ♦

### 3.3.2 Nonblocking supervisory control

The supervisor $S$ is introduced in order to restrict the evolution of the uncontrolled plant $G$ to an admissible subset $L_a \subseteq L_m(G)$. In order to synthesize such a supervisor, the requirements on $G$ need to be represented as an automaton that marks $L_a$. The admissible language $L_a$ is typically preceded by the modeling of a set of specifications by means of individual automata, denoted by $E_i$ in the following, with $i = 1, \ldots, n$. The global specification automaton $E$ is obtained from the parallel composition of all $E_i$. Then, using either a product or a parallel composition between $E$ and $G$, for $\Sigma_E \subseteq \Sigma_G$, an automaton $H_a$ is obtained such that $L_m(H_a) = L_a$.

In consequence, in order for the supervisor to restrict $G$ to the admissible behavior $L_a$, it is essential that $L_a$ is defined over the event set $\Sigma$ of $G$ and that it is included in the language marked by $G$: $L_a \subseteq L_m(G)$. Such a language respects the $L_m(G)$-closure condition, which is defined as follows:

***Definition 3.5 ($L_m(G)$-closure)***
*Let $L_a$ and $L(G)$, such that $L_a \subseteq L_m(G)$ with $L_a \neq \emptyset$, be two languages defined over an event set $\Sigma$. $L_a$ is $L_m(G)$-closed with respect to G if and only if:*

$$L_a = \overline{L_a} \cap L_m(G) \qquad \blacklozenge$$

A second necessary condition for the supervisor to enforce the strings in $L_a$ is that the admissible language needs to be controllable. To determine if a language is controllable, it must be verified that the restriction of $L(G)$ to $L_a$ does not lead to prohibiting the occurrences of uncontrollable events. So for any string $s \in L_a$ and for any event $\sigma \in \Sigma_{uc}$, if $s\sigma$ is a string of $L(G)$, then it must also be a string of $L_a$. This is formalized by the following definition:

***Definition 3.6 (Controllability)***
*Let $L_a$ and $L(G)$ be two languages defined over $\Sigma = \Sigma_{uc} \cup \Sigma_c$, where $\Sigma_c$ is the set of controllable events and $\Sigma_{uc}$ is the set of uncontrollable events. $L_a$ is controllable with respect to G and $\Sigma_{uc}$ if and only if:*

$$\overline{L_a}\Sigma_{uc} \cap L(G) \subseteq \overline{L_a} \qquad \blacklozenge$$

Then, the conditions for the existence of a non-blocking supervisor allowing to reach a marked state of $G$ are given by the following theorem:

***Theorem 3.2 (Nonblocking controllability theorem)***
*Consider the plant $G = (Q, \Sigma, \delta, q_0, Q_m)$ where $\Sigma_{uc} \subseteq \Sigma$ is the set of uncontrollable events. Consider the admissible language $L_a \subseteq L_m(G)$, where $L_a \neq \emptyset$. There exists a nonblocking supervisor S for G such that*

$$L_m(S/G) = L_a \;\; and \;\; L(S/G) = \overline{L_a}$$

*if and only if the two following conditions hold:*

1. *Controllability:* $\overline{L_a}\Sigma_{uc} \cap L(G) \subseteq \overline{L_a}$
2. $L_m(G)$*-closure:* $L_a = \overline{L_a} \cap L_m(G)$ ♦

This implies that if $L(G)$ is the maximally permissive behavior of the system, the admissible closed-loop behavior $L_m(S/G)$ can only be a subset of $L(G)$. The proof of this theorem is given in [Won17, p. 120] and [Cas08, p. 163].

### 3.3.3 Supremal controllable sublanguage

Because the $L_m(G)$-closure condition is inherent to the construction of $L_a$, if the former is not respected, the specification automata $E_i$ would need to be modified until the obtained $L_a$ is $L_m(G)$-closed. The controllability condition, on the other hand, depends on the properties of the DES under consideration, and so the admissible behavior can be uncontrollable. Therefore, if a given language $L_a$ is not controllable, the sublanguages $K$ of $L_a$ that are controllable with respect to $G$ and $\Sigma_{uc}$ should be found. The class of such sublanguages is defined as follows:

***Definition 3.7 (Class of controllable sublanguages of** $L_a$**)***
*The class of controllable sublanguages of* $L_a$*, denoted by* $\mathcal{C}(L_a)$*, is defined by:*

$$\mathcal{C}(L_a) := \{K \subseteq L_a \mid \overline{K}\Sigma_{uc} \cap L(G) \subseteq \overline{K}\}$$ ♦

Since $\emptyset \in \mathcal{C}(L_a)$, the class is not empty. Furthermore, because the union of two controllable languages is also a controllable language, $\mathcal{C}(L_a)$ is closed under union. Hence, the largest controllable subset of $L_a$, that is, the supremal controllable sublanguage of $L_a$ (denoted by $L_a^{\uparrow c}$ or $\sup\mathcal{C}(L_a)$), corresponds to the union of all the elements in $\mathcal{C}(L_a)$ and is defined as follows:

***Definition 3.8 (Supremal controllable sublanguage of** $L_a$**)***
*The supremal controllable sublanguage of* $L_a$*, denoted by* $L_a^{\uparrow c}$*, is defined by:*

$$L_a^{\uparrow c} := \bigcup\{K \mid K \in \mathcal{C}(L_a)\}$$ ♦

If no sublanguage $K \neq \emptyset$ is found, then $L_a^{\uparrow c} = \emptyset$, given that $\emptyset \in \mathcal{C}(L_a)$. On the other hand, if $L_a$ is controllable, then $L_a^{\uparrow c} = L_a$. The standard algorithm for calculating the supremal controllable sublanguage is presented in [Ram87; Cas08].

## 3.4 Decomposition approaches

As mentioned in the previous section, the plant $H_a$ representing the admissible behavior of a system is constructed from the combination of $G$ and $E$. This type of approach poses two problems, both related to the size of the obtained models. The first is the combinatorial explosion, as the automaton $H_a$ resulting from the combination of $G$ and $E$, respectively of $n$ states and $m$ states, can have in the worst case a number of $n \times m$ states. Therefore, $n$ and $m$ will grow exponentially with the number of $G_i$ and $E_i$ involved in the operation. This can pose a problem of computation time and memory size required, mitigated by the fact that existing software is capable of computing automata with several hundreds of thousands of states. Long before reaching this critical size, however, the designer is confronted with the problem of interpreting the models[1]; as the difficulty to interpret a model is not only related to the number of states, but also to the number of transitions and the structure of the automaton (self-loops, livelocks, etc.).

The following approaches seek to decompose the centralized (or "monolithic") control shown in Figure 3.2 in order to solve these two problems. They can be classified into three different types: horizontal decomposition, vertical decomposition and modal decomposition. These three types are not mutually exclusive and so approaches of different types can be combined.

### 3.4.1 Horizontal decomposition

Because a DES is often decomposable into $n$ elementary plants, the horizontal decomposition seeks to split the supervisory control into $m$ elementary supervisors. The global supervision is then ensured by the concurrent action of the elementary supervisors. Two main approaches can be distinguished: modular and decentralized.

#### Modular approach

The modular approach, illustrated in Figure 3.3, is based on a decomposition of the constraints in order to create several supervisors instead of one. Thus, each supervisor $S_i$ ($i = \{1,\dots,n\}$) enforces a single specification $L_{a_i}$ and all these local supervisors act simultaneously to restrict the behavior of the plant to the language $L_a = L_{a_1} \cap \cdots \cap L_{a_n}$ [Gou04; Kom08]. For instance, the supervisor $S$ of Figure 3.2 is decomposed into two supervisors, denoted by $S_1$ and $S_2$, in Figure 3.3. These two supervisors are built with respect to the plant $G$ and their intersection forms the $S_{mod}$ supervisor which restricts the behavior of the plant. Each supervisor allows or prohibits a set of events and only those that are not prohibited by any supervisor can be generated. Formally, this amounts to saying that if $S_{mod}$ is defined such that $S_{mod} = S_1 \cap S_2$, the modular control is such that:

- $L(S_{mod}/G) = L(S_1/G) \cap L(S_2/G)$

[1] By interpretation, we refer to the designer's ability to understand the behaviors modeled by the automaton.

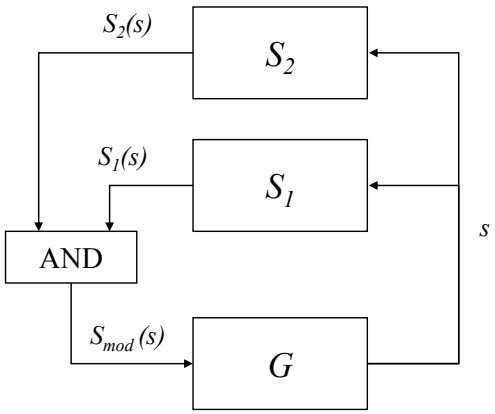



Figure 3.3 – Modular control architecture [Cas08]

- $L_m(S_{mod}/G) = L_m(S_1/G) \cap L_m(S_2/G)$

For each specification, it is possible to obtain a maximally permissive nonblocking supervisor. However, to obtain a nonblocking supervisor from the conjunction of two nonblocking supervisors, the following property must be verified:

***Definition 3.9 (Nonblocking modular supervisors)***
*Let $S_1$ and $S_2$ be two individual nonblocking supervisors for $G$. $S_{mod} = S_1 \cap S_2$ is non-blocking if and only if $L_m(S_1/G)$ and $L_m(S_2/G)$ are nonconflicting, that is, if and only if :*

$$\overline{L_m(S_1/G) \cap L_m(S_2/G)} = \overline{L_m(S_1/G)} \cap \overline{L_m(S_2/G)} \quad \blacklozenge$$

The modular control is said to be minimally restrictive if its closed-loop language is equivalent to that of a monolithic supervisor, that is $L(S/G) = L(S_{mod}/G)$ and $L_m(S/G) = L_m(S_{mod}/G)$. Furthermore, in the case where the admissible language $L(S_{mod}/G)$ cannot be enforced, the supremal controllable sublanguage $L_a^{\uparrow c}$ corresponds to the intersection of the supremal controllable sublanguage of each modular supervisor [Cas08], such that $L_a^{\uparrow c} = (L_{a_1} \cap L_{a_2})^{\uparrow c} = L_{a_1}^{\uparrow c} \cap L_{a_2}^{\uparrow c}$.

The advantages of this approach are an easier interpretation by reducing the size of the $S_i$ and $S_i/G$ models and the saving of time and computing resources. The two major drawbacks are the need to calculate the global plant $G$, which may be of consistent size, and the verification of the nonconflicting conditions of the two supervisors.

### Decentralized approach

The decentralized approach, illustrated in Figure 3.4 for two supervisors, is based on the decomposition of the centralized control into several supervisors alike the modular approach. In addition, it proposes to reduce the size of the plant on which each supervisor is built [Lin88; Rud92]. Indeed, on many real systems, the control can be divided into several controllers, each

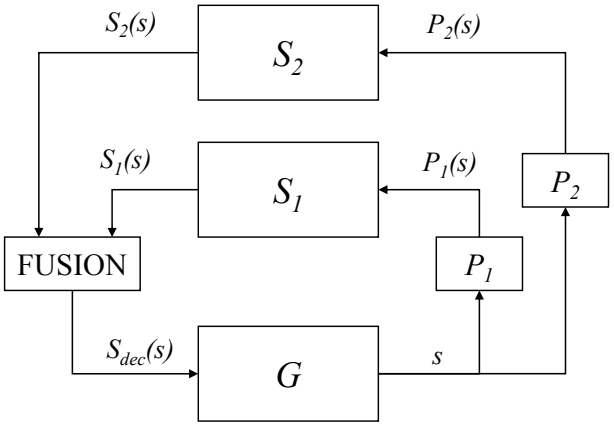



Figure 3.4 – Decentralized control architecture [Cas08]

observing only part of the system. Therefore, each local supervisor $S_i$ $(i = 1,\ldots,n)$ is based on a partial observation of the complete system, which is obtained by means of the projection function $P_i : \Sigma^* \to \Sigma_{o,i}^*$ (see Definition A.4) of the global plant alphabet $\Sigma$ onto the set of local observable events $\Sigma_{o,i}$, such that $\Sigma_{o,i}$ is a subset of the alphabet $\Sigma_o$ of observable events of the plant. The events in $\Sigma_o$ that are not included in $\Sigma_{o,i}$ are locally unobservable and form the set $\Sigma_{uo,i} = \Sigma_o \setminus \Sigma_{o,i}$. Thus, the control actions $S_i(s)$ of the individual supervisors for a string $s \in L(G)$ correspond to the control actions of the supervisors under partial observation $S_{P_i}$ for the strings projected by $P_i$, such that $S_i(s) := S_{P_i}[P_i(s)]$.

Similarly to the modular approach, the net control action on the global plant G is built on the joint action of all the individual $S_i(s)$. In most works, an event can only occur in the plant if it is authorized by all supervisors [Tak98; Jia00]. The authors of [Yoo02], however, have generalized this fusion rule. In consequence, the control actions of the individual supervisors can be merged either by *conjunction* or by *disjunction*. In a conjunctive fusion, an event is allowed in the plant $G$ only if it is not prohibited by any of the supervisors. In a disjunctive fusion, on the other hand, an event is prohibited on the plant $G$ only if it is not authorized by any of the supervisors. We shall detail in the following the decentralized control architecture associated to the conjunctive fusion. For more information on the disjunctive architecture, see [Yoo02; Cas08].

Because the conjunctive fusion rule is based on the intersection of enabled events (or complementarily, the disjunction of disabled events), the global control map $S_{dec}(s) : L(G) \longrightarrow 2^{\Sigma}$ is defined by:

$$S_{dec}(s) := \bigcap_{i=1}^{n} S_i(s)$$

The set of controllable events $\Sigma_c$ of the system is then divided into several local subsets $\Sigma_{c,i}$, such that $\Sigma_c = \cup_{i=1}^{n} \Sigma_{c,i}$. The set of locally uncontrollable events is then obtained such that $\Sigma_{uc,i} = \Sigma_c \setminus \Sigma_{c,i}$. Therefore, because the local supervisor $S_i$ can only disable the local controllable

events $\Sigma_{c,i}$, all controllable events in $\Sigma_c \setminus \Sigma_{c,i}$ are enabled by default and so the local supervisor follows a "permissive" decision rule. In consequence, the coobservability property must be respected for the set of local supervisors to control the admissible language [Cie88; Rud92]. The CP-coobservability, which is a relaxed version of this property, was introduced in [Yoo02] for a conjunctive (C) and permissive (P) architecture. The definition is given below:

***Definition 3.10 (CP-coobservability)***
*Let $L_a$ and $L(G) = \overline{L(G)}$ be two languages over an event set $\Sigma$. Let $\Sigma_{c,i}$ and $\Sigma_{o,i}$ be sets of controllable and observable events, respectively, for $i = 1,\ldots,n$. Let $P_i$ be the natural projection corresponding to $\Sigma_{o,i}$, with $P_i : \Sigma^* \to \Sigma_{o,i}^*$. $L_a$ is said to be CP-coobservable with respect to $L(G)$, $\Sigma_{o,i}$ and $\Sigma_{c,i}$, $i = 1,\ldots,n$, if for all $s \in \overline{L_a}$ and for all $\sigma \in \Sigma_c = \bigcup_{i=1}^{n} \Sigma_{c,i}$, $(s\sigma \notin \overline{L_a})$ and $(s\sigma \in L(G)) \Rightarrow$ there exists $i \in 1,\ldots,n$, such that $P_i^{-1}[P_i(s)]\sigma \cap \overline{L_a} = \emptyset$ and $\sigma \in \Sigma_{c,i}$.* ♦

In other words, if event $\sigma$ needs to be disabled, then at least one of the supervisors that can control $\sigma$ must unambiguously know that disabling $\sigma$ does not prevent any string in $L_a$; consequently, each supervisor can still follow the permissive policy when it is uncertain about whether it should disable an event or not. Then, the conditions for the existence of a decentralized and nonblocking supervisory control are given by the following theorem:

***Theorem 3.3 (Controllability and coobservability theorem - conjunctive case)***
*Consider the plant $G = (Q, \Sigma, \delta, q_0, Q_m)$ where $\Sigma_{uc} \subseteq \Sigma$ is the set of uncontrollable events, $\Sigma_c = \Sigma \setminus \Sigma_{uc}$ is the set of controllable events, and $\Sigma_o \subseteq \Sigma$ is the set of observable events. For each site $i$, $i = 1,\ldots,n$, consider the set of controllable events $\Sigma_{c,i}$ and the set of observable events $\Sigma_{o,i}$; overall, $\bigcup_{i=1}^{n} \Sigma_{c,i} = \Sigma_c$ and $\bigcup_{i=1}^{n} \Sigma_{o,i} = \Sigma_o$. Let $P_i$ be the natural projection from $\Sigma^*$ to $\Sigma_{o,i}^*$, $i = 1,\ldots,n$. Consider also the language $L_a \subseteq L_m(G)$, where $L_a \neq \emptyset$. There exists a nonblocking decentralized supervisor $S_{dec}$ for $G$ such that*

$$L_m(S_{dec}/G) = L_a \quad \text{and} \quad L(S_{dec}/G) = \overline{L_a}$$

*if and only if the three following conditions hold:*

1. *$L_a$ is controllable with respect to $L(G)$ and $\Sigma_{uc}$;*
2. *$L_a$ is $L_m(G)$-closed;*
3. *$L_a$ is CP-coobservable with respect to $L(G)$, $\Sigma_{o,i}$ and $\Sigma_{c,i}$, with $i = 1,\ldots,n$.* ♦

The proof of this theorem can be found in [Yoo02; Cas08]. Typically, some kind of communication between the decentralized supervisors exists so that the CP-coobservability condition is fulfilled [Bar00; Sch04]. Then, the decentralized control is minimally restrictive when $L(S/G) = \bigcap_{i=1}^{n} L(S_i/G)$ and $L_m(S/G) = \bigcap_{i=1}^{n} L_m(S_i/G)$. In the case where the admissible language $L_a$ cannot be enforced, [Lin88; Ove00] explain how to calculate the supremal controllable sublanguage $L_a^{\uparrow c}$ for a decentralized architecture.

Therefore, this approach reduces the size of the supervisor models, similarly to the modular architecture. However, when the individual specifications of the system are not totally independent (as it might be the case in HVDC systems), decentralized control is a good compromise between modular and monolithic control, as it allows the local supervisors to communicate just the necessary information.

### 3.4.2 Vertical decomposition

The principle of the vertical decomposition is based on the concept of multilevel models. Two approaches can be distinguished: hierarchical and compositional.

#### Hierarchical approach

The hierarchical approach is based on the decomposition of the DES as a multilevel structure, with a supervisor controlling the plant model at each level.

The authors of [Zho90; Won96] propose a system decomposed in two levels (Figure 3.5). The models indexed $lo$ are those of the low level and those of index $hi$ those of the high level. The low-level plant $G^{lo}$ is the model of the global plant, whereas $G^{hi}$ is an abstraction of this model that evolves on a high-level alphabet generated by $G^{lo}$ and is updated via the information channel $inf_{lohi}$. This high-level alphabet corresponds to the output alphabet of $G^{lo}$, which is a Moore automaton (introduced in Section 3.5.1). Thus, a silent event is associated to all states of $G^{lo}$ that must not cause an evolution in $G^{hi}$. The $S^{lo}$ model represents the supervisor controlling $G^{lo}$, while $S^{hi}$ is the supervisor controlling $G^{hi}$, via the control channels $con_{hi}$ and $con_{lo}$, respectively. The command channel $com_{hilo}$, on the other hand, conveys the high-level directives as commands to the low-level supervisor $S^{lo}$. The hierarchical approach, however, requires the introduction of the *hierarchical consistency* property, which makes it possible to check the consistency between the restrictions of low and high levels, so that the behavior obtained is as permissive as the one that would have been obtained by a centralized control [Zho90].

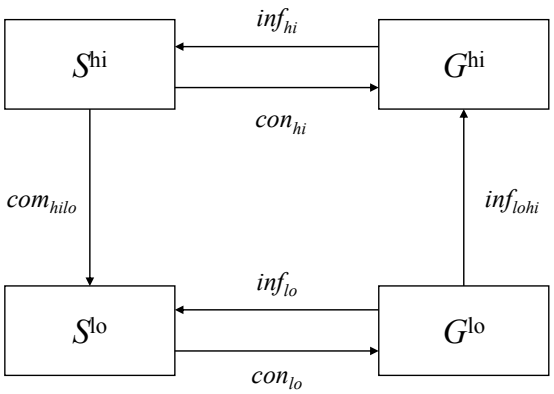



Figure 3.5 – Hierarchical control architecture [Zho90]

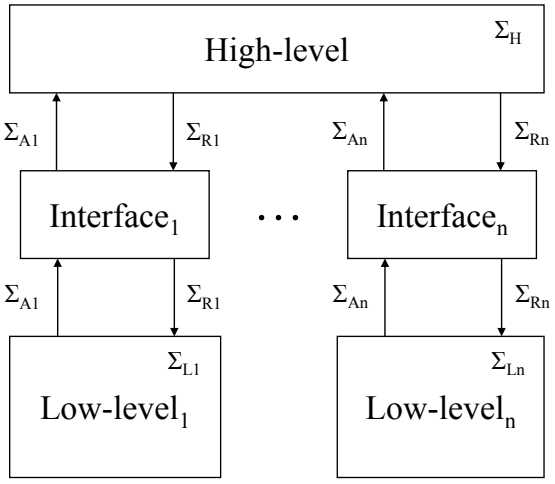



Figure 3.6 – Hierarchical interface-based control architecture [Led09]

A variant of the above architecture, shown in Figure 3.6, is presented in [Led09]. The plant is similarly decomposed in a high-level and several low-levels. The high-level plant $G^{hi}$, however, is defined over two types of event sets. The alphabet $\Sigma_H$ is formed by events specific to the high-level model. The events in the alphabets $\Sigma_R = \bigcup_{j=1}^{n} \Sigma_{Rj}$ and $\Sigma_A = \bigcup_{j=1}^{n} \Sigma_{Aj}$, on the other hand, are shared with the low-level automata. Likewise, the low-level plants $G^{lo,j}$ are defined over a set $\Sigma_{Lj}$ of events of their own, as well as over $\Sigma_{Rj}$ and $\Sigma_{Aj}$. Thus, the events in $\Sigma_R$ correspond to request commands, such as the beginning of a task; while those in $\Sigma_A$ correspond to answer responses, such as the end of a given task. Hence, $\Sigma_{hi} = \Sigma_H \cup \Sigma_R \cup \Sigma_A$ and $\Sigma_{lo,j} = \Sigma_{Lj} \cup \Sigma_{Rj} \cup \Sigma_{Aj}$. An interface level relates each request to its corresponding answer, such that $\Sigma_{Ij} = \Sigma_{Rj} \cup \Sigma_{Aj}$. Hence, the event set $\Sigma_I = \bigcup_{j=1}^{n} (\Sigma_{Rj} \cup \Sigma_{Aj})$ includes all the shared events. Then, the low-level nonblocking supervisors $S^{lo,j}$ are synthesized over $\Sigma_{Lj}$ while the high-level nonblocking supervisor $S^{hi}$ is synthesized over $\Sigma_{hi}$. Finally, such a control exists and is the minimum restrictive solution, if and only if the *interface consistency* property, which is an extension of the hierarchical consistency to the case of an interface-based architecture, is respected.

The hierarchical approach allows to reduce the size of the models as a pair of a supervisor and a plant is obtained for each considered level. The control structure, however, needs to be defined *a priori* and the consistency of the obtained models needs to be verified *a posteriori*. This approach can also be used in combination with the modular or decentralized approach, as shown in [Sch08; Fen08].

### Compositional approach

Compositional synthesis methods, as shown in Figure 3.7, abstract the elementary plant models in several levels of detail to remove the states and transitions that are superfluous for the purpose of synthesis [Flo07; Moh14]. The compositional approach exploits the modular structure of the plant to incrementally obtain a set of modular supervisors. However, the obtained

modular supervisors can either be arranged vertically or horizontally. Thus, the compositional approach is classified in this thesis as a vertical decomposition approach given the use we make of it in the next chapters. Unlike the hierarchical approach, a supervisor is not necessarily synthesized in each level.

In [Moh14], the authors propose a framework for compositional synthesis based on the notions of *synthesis abstraction*, *renaming* and *synthesis triples*. A plant composed of many elementary plants is represented by the following set of automata:

$$\mathcal{G} = \{G_1, G_2, \ldots, G_n\}$$

Then, compositional synthesis works by repeated abstraction of the elementary plants $G_i$, $i = 1, \ldots, n$. For instance, those events that appear in $G_i$ and in no other automata $G_j$, with $i \neq j$, can be abstracted, as they are local to $G_i$. The set of *local events* of the global plant is denoted by $\Upsilon$, while $\Omega = \Sigma \setminus \Upsilon$ denotes the set of *shared events* between the elementary plant's alphabets $\Sigma_i$, such that $\Sigma = \bigcup_{i=1}^{n} \Sigma_i$. The removal of local events from the elementary plants leads to simpler automata $\tilde{G}_i$. If further abstractions are possible, a simplified model $G_i'$ is obtained. Thus, both $\tilde{}$ and $'$ denote an abstraction equivalence. Once no further abstraction is possible, several plants in $\mathcal{G}$ are composed, so that new local events can be abstracted from the resulting automaton. Additionally, if one of the plants in $\mathcal{G}$ presents a behavior that should be clearly disabled by a supervisor (e.g. a controllable transition leading to a blocking state), a supervisor automaton $S_i$ is obtained along with the simplified automaton $G_i'$. These supervisors are included in the set of *collected supervisors*, denoted by $\mathcal{S}$.

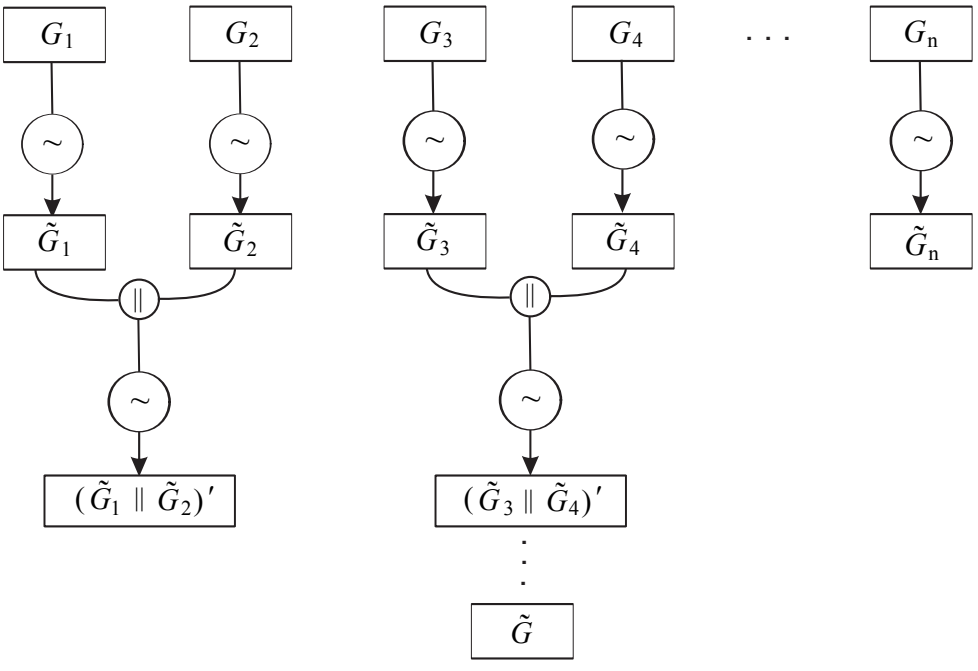



Figure 3.7 – General compositional approach [Moh12a]

Furthermore, the abstraction of some events could lead to non-determinism, that is, the transition function of $G_i$ could be not uniquely defined for a given event in a state. Such a problem is avoided by *renaming* the label of the corresponding event [Won04]. Therefore, the renaming is a controllability-preserving map that relates the events in the alphabet $\Sigma_\rho$ of the abstracted plants to those in the original event set $\Sigma$, such that $\rho : \Sigma_\rho \to \Sigma$. This operation is further extended to strings and languages by applying it to each event. If $\rho(\sigma) = \sigma$ for all $\sigma \in \Sigma$, then $\rho$ is the identity map id: $\Sigma \to \Sigma$.

To communicate correctly with the original plant, the renamed events need to be included in all the automata $G_j$, with $j \neq i$, and the collected supervisors in $\mathcal{S}$. A *$\rho$-distinguisher* (see Definition 3.11) automaton $D$, which differentiates between the renamed events such that only one of them is enabled at each state, is added as well to the set $\mathcal{S}$.

***Definition 3.11 ($\rho$-distinguisher)***
*Let $\rho : \Sigma_2 \to \Sigma_1$ be a renaming. An automaton $G_2$ with alphabet $\Sigma_2$ is a $\rho$-distinguisher if, for all traces $s,t \in L(G_2)$ such that $\rho(s) = \rho(t)$, it holds that $s = t$.* ♦

The status of the compositional synthesis at each step of the approach is therefore given by the set $\mathcal{G}$ of uncontrolled plant automata, the set $\mathcal{S}$ of collected supervisor automata and the renaming map $\rho$. This information can be combined in a *synthesis triple*, which is defined as follows [Moh14]:

***Definition 3.12 (Synthesis triple)***
*A synthesis triple is a triple $(\mathcal{G};\mathcal{S};\rho)$, where $\mathcal{G}$ is a set of deterministic plants, $\mathcal{S}$ is a set of deterministic supervisors and $\rho$ is a renaming, such that:*

1. *$L(\mathcal{S}) \subseteq L(\mathcal{G})$;*
2. *$\mathcal{S}$ is a $\rho$-distinguisher;*
3. *for all events $\gamma_1, \gamma_2$ such that $\rho(\gamma_1) = \rho(\gamma_2)$, there exists at most one automaton $G_j \in \mathcal{G}$ that differentiates $\gamma_1$ from $\gamma_2$.* ♦

Once no further abstraction of the plant models is possible, the final synthesis result is then obtained by composing the monolithic supervisor representing the supremal controllable sublanguage for the remaining plants, sup$\mathcal{C}(\mathcal{G})$, with the intermediate supervisors collected in $\mathcal{S}$ during the different steps of the synthesis, and renaming if needed. In [Moh14], for algorithmic reasons, the specification automata are transformed into plants by adding, for every uncontrollable event that is not enabled in a state, a transition to a new blocking state. As the concerns on algorithm optimization are out of the scope of this thesis, however, the supervisors are calculated on specification and plant automata in our work.

The final synthesis result can then be defined by the following synthesis triple:

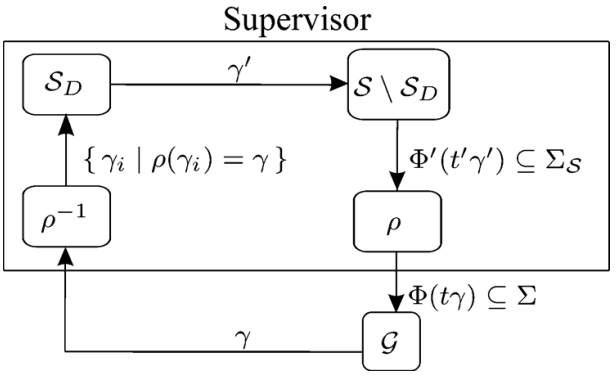



Figure 3.8 – Control architecture for the compositional approach [Moh14]

***Definition 3.13 (Synthesis result for a synthesis triple)***
*Let* $(\mathcal{G};\mathcal{S};\rho)$ *be a synthesis triple. Then* $\sup\mathcal{C}(\mathcal{G};\mathcal{S};\rho) := \rho(\sup\mathcal{C}(\mathcal{G}) \parallel \mathcal{S})$. ♦

The resulting control architecture is illustrated in Figure 3.8, where $\mathcal{G}$ is the original plant, $\mathcal{S}$ are the computed modular supervisors and $\mathcal{S}_D \subseteq \mathcal{S}$ are $\rho$-distinguishers. Thus, the inverse renaming function $\rho^{-1}$ relates the events of the original plants $\mathcal{G} = \{G_1, G_2, \ldots, G_n\}$ to the renamed events, e.g. $\rho^{-1}(\gamma) = \{\gamma_1, \gamma_2\}$. The distinguisher automata are then able to recognize the event that took place and the transition labeled by it occurs. The modular supervisors, based on the distinguishers alphabets, output a dedicated control map. The control decision is then renamed by the $\rho$ function, e.g. $\rho(\gamma_2) = \gamma$, such that the original plants recognize the event. The compositional approach is illustrated in Appendix C through the example presented in [Moh14].

For the compositional control architecture to be equivalent to the monolithic control of the original plant, every abstraction step must ensure that the synthesis result is identical as it would have been for the non-abstracted components. This property is called *synthesis equivalence*:

***Definition 3.14 (Synthesis equivalence)***
*Two triples* $(\mathcal{G}_1;\mathcal{S}_1;\rho_1)$ *and* $(\mathcal{G}_2;\mathcal{S}_2;\rho_2)$ *are synthesis equivalent* $(\mathcal{G}_1;\mathcal{S}_1;\rho_1) \simeq_{synth} (\mathcal{G}_2;\mathcal{S}_2;\rho_2)$, *if* $L(\sup\mathcal{C}(\mathcal{G}_1;\mathcal{S}_1;\rho_1)) = L(\sup\mathcal{C}(\mathcal{G}_2;\mathcal{S}_2;\rho_2))$. ♦

Then, from the original triple $(\mathcal{G};\mathcal{S};\mathrm{id})$, the intermediate synthesis triples $(\mathcal{G}_k;\mathcal{S}_k;\rho_k)$ are continuously abstracted through $k$ iterations such that synthesis equivalence is preserved until $\mathcal{G}_k = \emptyset$. Whenever an abstracted model needs to be restricted, the corresponding supervisor is included in $\mathcal{S}$. Hence, a minimally restrictive compositional control, whose control loop is equivalent to the one of a monolithic control, exists as stated in the following theorem:

***Theorem 3.4 (Compositional and monolithic control equivalence theorem)***
*Let* $\mathcal{G} = \{G_1, G_2, \ldots, G_n\}$ *be a set of automata, and let* $(\mathcal{G};\mathcal{G};\mathrm{id}) \simeq_{synth} (\emptyset;\mathcal{S};\rho)$. *Then* $L(\rho(\mathcal{S})) =$

$L(\sup\mathcal{C}(\mathcal{G}))$. ♦

The *projection* operation constitutes the most common abstraction operation on synthesis triples. *Parallel composition* and *monolithic synthesis* are also methods that allow to rewrite a triple such that synthesis equivalence is preserved, that is respectively:

$$(\{G_1, G_2, \ldots, G_n\}; \mathcal{S}; \rho) \simeq_{synth} (\{G_1 \parallel G_2, \ldots, G_n\}; \mathcal{S}; \rho)$$

and

$$(\mathcal{G}; \mathcal{S}; \rho) \simeq_{synth} (\emptyset; \mathcal{S} \cup \{\sup\mathcal{C}(\mathcal{G})\}; \rho).$$

Other methods include *halfway synthesis* [Flo07] and *local supervisors* [Su10], which are different strategies to remove from the plants the uncontrollable transitions leading to blocking states; *bisimulation* [Mil89] and methods based on *observation equivalence* [Moh11; Moh12b], which merge those states that reach equivalent states by the same strings.

To conclude, the compositional approach reduces the size of the obtained models by calculating intermediate supervisors for the different abstracted versions of the plants. Although compositional synthesis results in a set of modular supervisors, the latter can be vertically arranged, depending on the plant structure. Unlike the hierarchical approach, however, the control levels need not be defined *a priori*, as they emerge naturally through the different model abstractions. Furthermore, because the synthesis equivalence is verified at each abstraction step, the obtained models are consistent by construction and so the consistency of the whole architecture need not be validated *a posteriori*.

### 3.4.3 Modal decomposition

The modal approach relies on the decomposition of the system operation using the concept of mode. A mode is seen as a non-permanent operation of the system that has a beginning, an end, and carries out a function in the broad sense. Then, the succession of modes corresponds to a succession of non-permanent operations of the system such that, considered together, correspond to a continuous and permanent operation of the latter [Far10].

The first step when using the modal approach lies in identifying the modes and deriving models from them when designing a system. It is then necessary to discriminate data from the specifications to focus only on those useful for each non-permanent operation of the system. All of this information, for a given mode, characterizes a particular configuration of the system. The second stage concerns the succession of modes and more specifically the transitions between modes. Indeed, if the non-permanent operations of the system are known, by the specifications, it is necessary to determine the functioning of the system between its modes and which information is needed to completely characterize a configuration change.

The modal decomposition has been treated in the literature for a long time, with in particular the work on the GEMMA (in French, *Guide d'Etude des Modes de Marches et d'Arrêts*) [ADE81]. All the proposals since then have tried to define rigorously the stages of construction of the models. As the system's complexity increases, these building steps become also more complex and it is necessary to respect properties on these models in order to define them completely and mathematically [Jah94; Koo01]. However, few methods are completely defined, automatable and consider all possible modes and evolutions of the system. Indeed, a difficult compromise between exhaustivity and genericity must be found, as a too general approach makes the proposal inapplicable while an overly detailed approach is no longer generic and applies only to a specific type of system. Furthermore, experiencing a loss of information during mode commutation [Mar98] or forgetting to represent a mode is equivalent to no longer having a state that corresponds to the physical state of the system, which can potentially lead to severe malfunction of the system.

We shall focus in the following on the works that preceded our research in the Ampère Laboratory, based on the SCT framework and a systematic approach that defines a generic commutation pattern, which allows to identify the most obvious modes but also those unscheduled (either because they do not fit the pattern or because of the complex commutative behavior necessary to reach them). This approach allows not to be limited to a predefined canvas while limiting the number of modes (and therefore the combinatorial explosion), as only those necessary for the current operation of the system are considered. Early works focused on the representation of a multi-mode system, which behaves differently depending on the specifications to be respected in each distinct situation through its various configurations, as well as on the dependability, i.e. the expected specifications should be met even in case of failures [Nie95; Nou96]. In the continuity of the previous works, the thesis of Kamach [Kam04] and Faraut [Far10] aim to propose a multi-model approach, illustrated in Figure 3.9 as a UML activity diagram, to control a system with $n$ modes of operation and where several failures can occur.

As stated in [Far09], the *set of components* $C_i$ of the system is defined as $\mathcal{C} = \{C_1, C_2, \ldots, C_i\}$, where $i \in \mathbb{N}$ and $i \geq 1$. Each component is then modeled according to the following definition (Activity 1 of Figure 3.9):

***Definition 3.15 (Model of a component)***
*A component $C_i$ is modeled by an automaton $G^{C_i} = (Q^{C_i}, \Sigma^{C_i}, \delta^{C_i}, q_0^{C_i}, Q_m^{C_i})$, where:*

- *$Q^{C_i}$, $Q_m^{C_i}$ and $q_0^{C_i}$ are respectively the set of states, the set of marked states and the initial state of the component $C_i$;*
- *$\Sigma^{C_i}$ is the event set of the component $C_i$, including two partitions:*
  - *$\Sigma^{C_i} = \Sigma_c^{C_i} \cup \Sigma_{uc}^{C_i}$ with $\Sigma_c^{C_i} \cap \Sigma_{uc}^{C_i} = \emptyset$. $\Sigma_c$ and $\Sigma_{uc}$ are respectively the disjoint subsets of the controllable and uncontrollable events of the component $C_i$;*
  - *$\Sigma^{C_i} = \Sigma_{\rightleftarrows}^{C_i} \cup \Sigma_{\circlearrowleft}^{C_i}$ with $\Sigma_{\rightleftarrows}^{C_i} \cap \Sigma_{\circlearrowleft}^{C_i} = \emptyset$. $\Sigma_{\rightleftarrows}^{C_i}$ is the set of switch events of $C_i$. $\Sigma_{\circlearrowleft}^{C_i}$ are the*

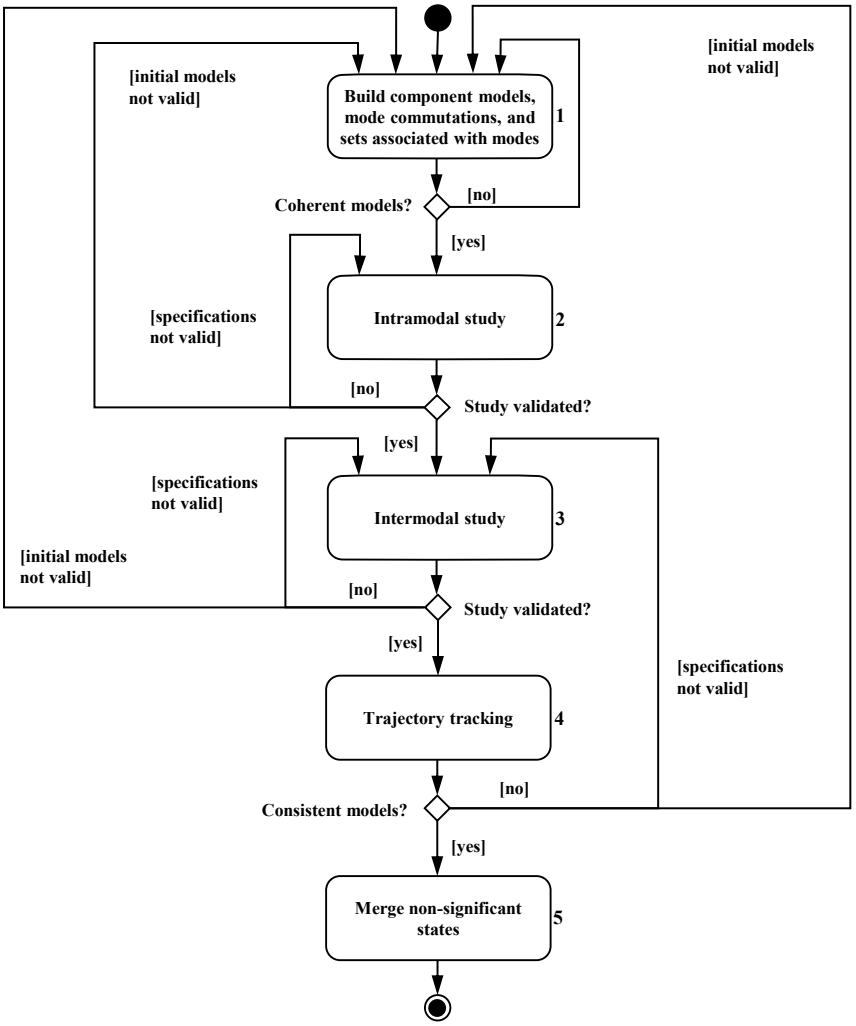



Figure 3.9 – Modal decomposition approach [Far10]

*other events of the component.*

- $\delta^{C_i}$ *is the transition function. It includes* $\delta^{C_i}_{\rightleftarrows}$*, which represents the set of switch transitions where the event labeling the transition is in* $\Sigma^{C_i}_{\rightleftarrows}$*.* ♦

From the above definitions, it is possible to define the set of all modes of the system $\mathcal{M} = \{M_0, M_1, \ldots, M_j\}$, where $M_j$ is the name of the represented mode. At every moment the system is in one and only one mode. Because a mode is a particular physical configuration of the system where it must meet a set of specifications, the particular configuration that is characterized by a set of components used in the considered mode is defined as follows:

***Definition 3.16 (Set $\mathcal{C}^{M_j}$ of components in a mode $M_j$)***
*We define $\mathcal{C}^{M_j}$ as the set of components used in the mode $M_j$, where $\mathcal{C}^{M_j} = \mathcal{C}^{M_j}_{\circlearrowleft} \cup \mathcal{C}^{M_j}_{\leftarrow} \cup \mathcal{C}^{M_j}_{\rightarrow}$ such that:*

- $\mathcal{C}_{\circlearrowleft}^{M_j}$ *is the set of components representing the* intramodal *behavior of the plant in* $M_j$*;*
- $\mathcal{C}_{\leftarrow}^{M_j}$ *is the set of components generating a switch event involving an* activation *of* $M_j$*;*
- $\mathcal{C}_{\rightarrow}^{M_j}$ *is the set of components generating a switch event involving a* deactivation *of* $M_j$*;*
- $\mathcal{C}_{\rightleftarrows}^{M_j} = \mathcal{C}_{\leftarrow}^{M_j} \cup \mathcal{C}_{\rightarrow}^{M_j}$ *is the set of components that can generate a switch event.*

*All the components contained in the modes are included in the set of components:* $\mathcal{C} = \bigcup_{M_j \in \mathcal{M}} \mathcal{C}^{M_j}$ ♦

Then, it is up to the designer to model a modes automaton representing the commutative behavior between the modes of the system:

***Definition 3.17 (Modes automaton)***
*Let a modes automaton* $G^{\mathcal{M}}$ *represent the switch behavior of the system. Formally, this automaton is denoted by* $G^{\mathcal{M}} = (Q^{\mathcal{M}}, \Sigma^{\mathcal{M}}, \delta^{\mathcal{M}}, q_0^{\mathcal{M}}, Q_m^{\mathcal{M}})$ *such that:*

- $Q^{\mathcal{M}} = \mathcal{M}$
- $\Sigma^{\mathcal{M}}$ *is the set of switch events of the studied system,*
- $\delta^{\mathcal{M}} : \mathcal{M} \times \Sigma^{\mathcal{M}} \rightarrow \mathcal{M}$ *is the transition function of the modes automaton,*
- $q_0^{\mathcal{M}} \in \mathcal{M}$,
- $Q_m^{\mathcal{M}} \subseteq \mathcal{M}$. ♦

Once the conditions given in [Far10, p. 57] to verify the coherence between models are respected, i.e. all modes have both a switch and an internal behavior such that $\forall M_j \in \mathcal{M}$, $\mathcal{C}_{\circlearrowleft}^{M_j} \neq \emptyset \wedge \mathcal{C}_{\rightleftarrows}^{M_j} \neq \emptyset$, the plant $G_{in}^{M_j} = ||_{C_i \in \mathcal{C}_{\circlearrowleft}^{M_j}} G^{C_i}$ representing the internal behavior in the mode $M_j$ is constructed for each mode (Activity 2 of Figure 3.9). Similarly, each specification $E_{in}^{l,M_j}$ (with $l \in \mathbb{N}$) is modeled. $E_{in}^{M_j}$ is then defined as the model of the specifications to be respected in the mode $M_j$. The controlled plant $H_{in}^{M_j}$ in the mode $M_j$ is then obtained by the following definition:

***Definition 3.18 (Internal controlled plant*** $H_{in}^{M_j}$***)***
$H_{in}^{M_j} = (Y_{in}^{M_j}, \Sigma_{in}^{M_j}, \tau_{in}^{M_j}, y_{in,0}^{M_j}, Y_{in,m}^{M_j})$ *where* $L_m(H_{in}^{M_j}) = [L_m(G_{in}^{M_j} \times E_{in}^{M_j})]^{\uparrow c}$ ♦

Once the controlled plant has the expected behavior, the intramodal study is concluded. Nevertheless, these models do not represent all the possible commutations with the other modes. Hence, the plant $G_{in}^{M_j}$ is extended so that the components in $\mathcal{C}_{\circlearrowleft}^{M_j}$ are taken into account (Activity 3 of Figure 3.9). Formally, the extended plant $G^{M_j}$ is constructed such that $G^{M_j} = (Q^{M_j}, \Sigma^{M_j}, \delta^{M_j}, q_0^{M_j}, Q_m^{M_j})$, where $G^{M_j} = [||_{C_k \in \mathcal{C}^{M_j}} G^{C_k}]||G^{\mathcal{M}}$. Likewise, the intramodal specifications, named $E_{\circlearrowleft}^{M_j}$, are manually extended so that the added events are considered, although it is not necessary to completely redesign the specification models. In this step, new models $E_{\rightleftarrows}^{M_j}$ representing the specifications related to mode-switching are also added. Hence, the spec-

ification $E^{M_j}$ is constructed from the product (cf. Definition B.4) of the specifications $E_{\circlearrowleft}^{M_j}$ and $H_{\rightleftarrows}^{M_j}$, themselves constructed from basic specifications $E_{\circlearrowleft}^{l,M_j}$ and $E_{\rightleftarrows}^{l,M_j}$.

From the plant automata $G^{M_j}$ in the considered modes and the specification automata $E^{M_j}$, the designer can obtain the controlled plants $H^{M_j}$ representing the admissible behaviors of the extended plants in each mode. Formally, the synthesis of $H^{M_j}$ is done as shown in Section 3.3:

***Definition 3.19 (Extended controlled plant $H^{M_j}$)***
$H^{M_j} = (Y^{M_j}, \Sigma^{M_j}, \tau^{M_j}, y_0^{M_j}, Y_m^{M_j})$ *where* $L_m(H^{M_j}) = [L_m(G^{M_j} \times E^{M_j})]^{\uparrow c}$ ♦

Then, because a string that activates a mode must always put the system in the same physical state (otherwise a determinism problem appears during the commutation), it is necessary, for each transition of $G^{\mathcal{M}}$, to track the trajectories in order to search for the trajectories of $H^{M_j}$ leading to a state where an event provoking a commutation to $H^{M_k}$ is generated, to find the states reached in the mode of arrival by these trajectories and to check that for each commutation a unique state is reached (Activity 4 of Figure 3.9).

Because the modes automaton $G^{\mathcal{M}}$ allows to include in each state of the models the information of the active mode, it is possible to isolate the states corresponding to the internal and commutative behavior of the system and to merge the other states (Activity 5 of Figure 3.9). The plant obtained is denoted by $H_{merge}^{M_j}$. Thus, the set of states not included in the active mode, called *non-significant states*, form a new state called *inactive state*, named $y_{id}^{M_j}$, which represents the inactivity of the mode. This state is used to represent the *uniqueness of active mode*, i.e. the assumption that only one supervisor is active at a time [ADE81; Jah94; Koo01; Kam05]. Its purpose is to avoid the inconsistency and indeterminism caused by the existence of several controllers globally controlling the same system. Thus, the concurrent behavior of all $H_{merge}^{M_j}$ represents the overall behavior of the system. This is equivalent to performing an implicit parallel composition between them, as expressed by the following definition:

***Definition 3.20 (Equivalent global behavior)***
*Considering the plant $H_{merge}$ representing the concurrent behavior of all modes of the system, with $\mathcal{M}$ being the set of all modes, we have:* $H_{merge} = ||_{M_j \in \mathcal{M}} H_{merge}^{M_j}$. ♦

Furthermore, because parallel composition synchronizes common events, when the $N-1$ inactive modes are in their idle state, it must be ensured that they do not restrict the behavior of the active mode. For this reason, it must be guaranteed that the inactive state can generate all possible events, either in a self-loop transition or out of this state. Formally, if $Y_{mer}^{M_j}$ is the set of non-significant states, the automaton $H_{merge}^{M_j}$ is defined such that:

***Definition 3.21 (Merged controlled plant in mode $M_j$)***
$H^{M_j}_{merge} = (Y^{M_j}_{merge}, \Sigma^{M_j}_{merge}, \tau^{M_j}_{merge}, y^{M_j}_{merge,0}, Y^{M_j}_{merge,m})$ *such that:*

- $Y^{M_j}_{merge} = (Y^{M_j} \setminus Y^{M_j}_{mer}) \cup \{y^{M_j}_{id}\}$
- $\Sigma^{M_j}_{merge} = \Sigma^{M_j} \setminus \bigcup_{C_i \in (\mathcal{C}^{M_j}_{\rightleftarrows} \setminus \mathcal{C}^{M_j}_{\circlearrowleft})} \Sigma^{C_i}_{\circlearrowleft}$
- $\tau^{M_j}_{merge} = \begin{array}{l} \tau^{M_j} \quad \setminus \{((y_a, s), y_b) \mid s \in \Sigma^{M_j},\ y_a, y_b \in Y^{M_j}_{mer},\ \tau^{M_j}(y_a, s) = y_b \text{ is defined}\} \\ \qquad \cup \{((y^{M_j}_{id}, s'), y^{M_j}_{id}) \mid s' \in (\Sigma^{M_j} \setminus \Sigma^{M_j}_{\leftarrow})\} \end{array}$
- $y^{M_j}_{merge,0} = \begin{cases} y^{M_j}_0 & \text{if} \quad y^{M_j}_0 \notin Y^{M_j}_{mer} \\ y^{M_j}_{id} & \text{otherwise} \end{cases}$
- $Y^{M_j}_{merge,m} = \begin{cases} Y^{M_j}_m & \text{if} \quad Y^{M_j}_m \cap Y^{M_j}_{mer} = \emptyset \\ Y^{M_j}_m \setminus Y^{M_j}_{mer} \cup \{y^{M_j}_{id}\} & \text{otherwise} \end{cases}$

♦

As a result of the modal approach, the size of the models is reduced given that a controlled plant is obtained for each mode, with each mode representing a reduced and particular behavior of the system. The conditions for the modal control to be equivalent to its centralized counterpart are given in [Far10, p. 94].

## 3.5 Implementation of automata

Nowadays, the control of industrial discrete-event systems mainly relies on the direct implementation of control tasks based on the interpretation of informal specifications and text documents by a control engineer. Nevertheless, this practice often leads to a deficient control logic that is difficult to reconfigurate and maintain. Therefore, formal approaches, such as the SCT, aim to ease the development of such control [Zay17]. Hence, the goal of the supervisors obtained by synthesis in the form of automata, is to control a real plant, i.e. to be ultimately implemented as a closed-loop control with I/O signals. The asynchronous event-driven framework proposed in the SCT, however, is not respected in the synchronous signal-based devices, such as Programmable Logic Controllers (PLCs), in which they are implemented. Furthermore, the plant is considered to be a generator of events in the SCT, whereas it is often the case where the evolution of the real plant must be forced by an external agent, such as a controller. Thus, the set of supervisors cannot be directly implemented in computer-based devices.

### 3.5.1 Automata with inputs and outputs

Two variants to the definition of automaton given in Section 3.2.2 can be derived: the Moore automaton and the Mealy automaton (equivalently referred to as Moore and Mealy machines, respectively). Unlike ordinary automata, which have a single alphabet, Moore and Mealy automata possess both an input and an output alphabet [Cas08]. These types of automata are named after G. H. Mealy and E. F. Moore, who defined them in 1955 and 1956, respectively.

A Mealy automaton is a finite-state automaton that relates an input alphabet to an output alphabet depending on its current state and its current input symbol. Thus, along the transition function that is common to all automata, an output function needs to be declared. A Mealy automaton can be formally defined as follows:

***Definition 3.22 (Mealy automaton)***
*A Mealy automaton $G_{Mealy}$ is defined by a sextuple $G_{Mealy} = (Q, q_0, \Sigma, \Lambda, \delta, \omega)$ such that:*

- *$Q$ is a finite set containing all the states of $G_{Mealy}$;*
- *$q_0$ is the initial state of automaton $G_{Mealy}$, such that $q_0 \in Q$;*
- *$\Sigma$ is the input alphabet associated with $G_{Mealy}$;*
- *$\Lambda$ is the output alphabet associated with $G_{Mealy}$;*
- *$\delta$ is the transition function, defined by $\delta : Q \times \Sigma \to Q$, mapping a pair of a state and an input to the corresponding next state;*
- *$\omega$ is the output function, defined by $\omega : Q \times \Sigma \to \Lambda$, mapping a pair of a state and an input in $\Sigma$ to the corresponding output event in $\Lambda$.* ♦

An example of the state-transition diagram of a Mealy automaton is shown in Figure 3.10a. When the system is in state $x$, if the Mealy automaton receives an input symbol $\sigma_1$, the transition to state $y$ will be realized and the output symbol $\lambda_1$ generated in the process.

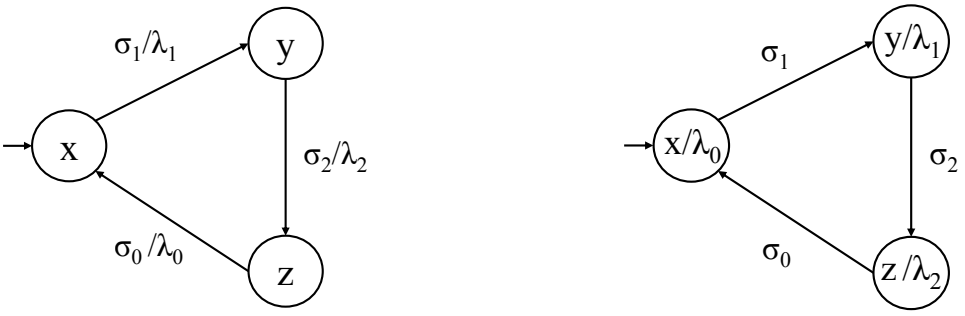
A Moore automaton is a finite-state automaton that relates an input alphabet to an output alphabet depending on its current state alone. It can be formally defined as follows:

***Definition 3.23 (Moore automaton)***
*A Moore automaton $G_{Moore}$ is defined by a sextuple $G_{Moore} = (Q, q_0, \Sigma, \Lambda, \delta, \omega)$ such that:*

- *$Q$ is a finite set containing all the states of $G_{Moore}$;*
- *$q_0$ is the initial state of automaton $G_{Moore}$, such that $q_0 \in Q$;*
- *$\Sigma$ is the input alphabet associated with $G_{Moore}$;*
- *$\Lambda$ is the output alphabet associated with $G_{Moore}$;*
- *$\delta$ is the transition function, defined by $\delta : Q \times \Sigma \to Q$, mapping a pair of a state and an input to the corresponding next state;*
- *$\omega$ is the output function, defined by $\omega : Q \to \Lambda$, mapping each state in $Q$ to the corresponding output in $\Lambda$.* ♦

An example of the state-transition diagram of a Moore automaton is shown in Figure 3.10b. When the system is in state $x$, if the Moore automaton receives an input symbol $\sigma_1$, the transition to state $y$ will be realized and the output symbol $\lambda_1$ generated upon entrance in state $y$.

As claimed in [Cas08], a Mealy automaton can be viewed as an ordinary automaton where the event set $\Sigma$ is the set of all input/output labels of the Mealy automaton. Therefore, the language

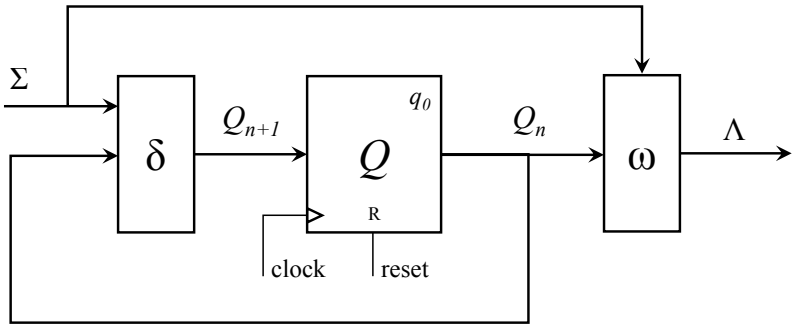



(a) Mealy automaton (b) Moore automaton

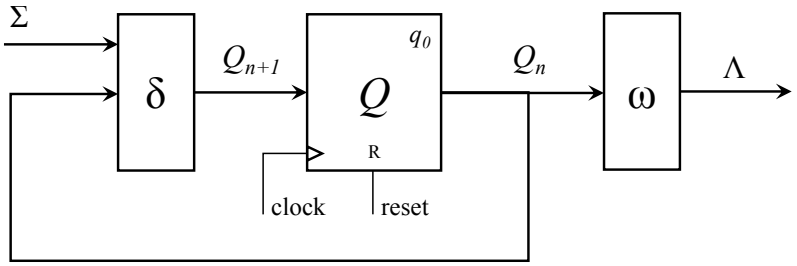
Figure 3.10 – State-transition diagrams of Mealy and Moore automata

generated by the automaton will be the set of all input/output strings that can be generated by the Mealy automaton. The same can be said about a Moore automaton, as it can always be transformed into a Mealy automaton. Thus, the operations on automata defined in Appendix B apply for Mealy and Moore automata.

Every Moore automaton can be converted to a Mealy automaton (the inverse is not always true). Depending on the system to be modeled, however, one type of automaton could be more appropriate than the other. For instance, if both machines are implemented as sequential logic circuits, as illustrated in Figure 3.11, it can be observed that the output function $\omega$ of the Moore implementation is updated at every clock cycle, whereas its Mealy counterpart is updated at each change in the input set. Thus, the sequential logic circuit corresponding to the practical




(a) Mealy implementation




(b) Moore implementation

Figure 3.11 – Sequential logic circuits of Mealy and Moore automata

implementation of a Moore automaton follows a synchronous sequential logic, as opposed to the asynchronous sequential logic of the sequential logic circuit of the Mealy machine. Hence, the interconnection of several Mealy machines implemented as sequential logic circuits could lead to possible instabilities in the system.

In consequence, the use of Moore automata is generally preferred for systems composed of interacting components equipped with a set of sensors recording the physical state of the system, such as HVDC systems, as the output of a state depends on the set of measurements of all sensors when the system is in a particular physical state. Mealy automata, on the other hand, are preferred for asynchronous systems such as communication protocols and software systems.

However, in order to implement Moore and Mealy automata in the form of sequential logic circuits (or any other synchronous and signal-based implementation), the issues presented in the next section on the theoretical framework under which the automata are designed need to be solved first, as this is often not respected in the environment where they are implemented.

### 3.5.2 Implementation issues

The authors of [Fab98] identified the main problems encountered during the implementation of abstract supervisors into PLCs. Given that PLCs are industrial digital computers, these problems are extensible to all computer-based environments and can be resumed to the following:

- *Avalanche effect*: occurs when a change in the value of an input signal activates an event that triggers many successive transitions; the controller then jumps through an arbitrary number of states;
- *Simultaneity*: relates to the incapability of the controller to distinguish the order of occurrence of two events within the same scan cycle, because the input signals are read at the beginning of the scan cycle;
- *Inexact synchronization*: is due to the communication delay between the plant and the controller. Thus, the occurrence of an event in the former is not immediately reproduced in the latter, unlike in SCT [Kum91]. This could lead to a wrong control action by the controller;
- *Causality*: arises when the implemented supervisor is forced to generate commands so that the plant can evolve in response, as opposed to the SCT, where the plant is supposed to generate all the events in its event set;
- *Choice*: appears when the controller has to choose between the generation of two or more concurrent commands, thus introducing a non-determinism issue.

While the avalanche effect issue can be resolved by means of clever programming [Fab98], the simultaneity and inexact synchronization problems are inherent to this type of implementation. Thus, the *interleave insensitivity* and the *delay insensitivity* properties are respectively introduced in [Fab98] and [Bal92] for a plant $G$ and a supervisor $S$ as a solution to the simul-

taneity and the inexact synchronization issues. These properties state that the control map of the supervisor remains unaltered in the case where two strings intertwine:

***Definition 3.24 (Interleave insensitivity)***
*A supervisor S is interleave insensitive with respect to a plant G and a sub-alphabet* $\Sigma' \in \Sigma_G$ *if for* $s_1, s_2 \in (\Sigma_G - \Sigma')^*$ *and* $\sigma' \in \Sigma'$, $ss_1s_2\sigma' \in L(G \parallel S) \Rightarrow s(s_1|||s_2)\sigma' \subseteq L(G \parallel S)$. ♦

***Definition 3.25 (Delay insensitivity)***
*A language K is said to be delay insensitive if, for* $s \in \overline{K}, \sigma_c \in \Sigma_S, \sigma_{uc} \in \Sigma_G$ $s\sigma_c, s\sigma_{uc} \in \overline{K} \Rightarrow$ $s\sigma_{uc}\sigma_c, s\sigma_c\sigma_{uc} \in \overline{K}$. ♦

These properties, however, are not often satisfied in practice. Thus, a practical solution to these problems is to fix the scan cycle of the control program to be significantly smaller than the time constants of the physical system.

Regarding the causality problem, in SCT, the plant is supposed to generate all the events in its alphabet, independently of their controllability and observability. Then, the supervisor merely enables or disables certain strings to keep the behavior of the plant within the control specifications (Figure 3.2). This abstraction considerably simplifies the theory, by keeping all implementation aspects out of the supervisor synthesis and considering the automata models as purely mathematical objects. In most real systems, however, the plant's behavior does not evolve spontaneously, but it rather evolves in response to given commands. In consequence, the *forcing event* approach, shown in Figure 3.12a, is introduced in [Bal92]: the controller generates the commands (identified as the controllable events by the author), while the plant generates the responses of the system (i.e. the uncontrollable events). Because the controller has to generate commands in accordance with the supervisor's control map but also with the physical limitations of the plant, the automaton recognizing the supremal controllable language is implemented. This implies that only centralized architectures can be realized.

In opposition, the authors of [Cha99] introduce the *supervised control* approach (shown in Figure 3.12b), which clearly differentiates the supervision and command generation tasks within the controller. The supremal controllable language is ensured by a closed-loop control, as an automaton modeling the plant generates all the events from the I/O signals, while the supervisor disables them according to its control policy. This separation of tasks allows to modify the supervisor structure independently of the command generation, thus offering the possibility to integrate different architectures, such as the modular control [Que02; Lop12; Lea12; Vie17].

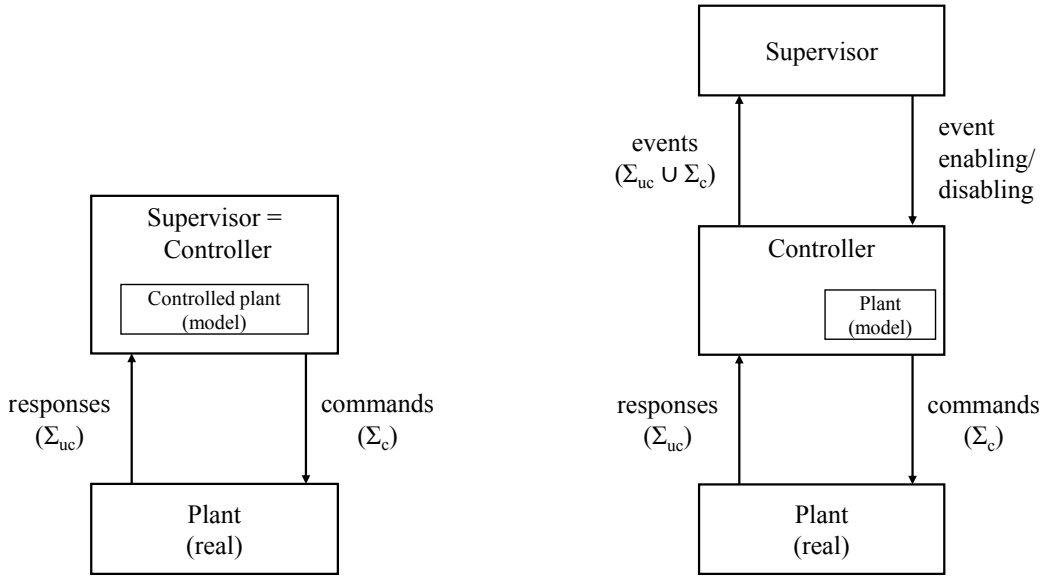
Furthermore, the synthesized supervisor obtained within the SCT framework is minimally restrictive, that is, it respects the specifications while allowing the plant the largest possible amount of freedom in its event generation. This means that the plant will eventually have to choose between alternative paths offered by the supervisor at a given state. However, because

the implemented controller generates the commands associated to the controllable events, only one of the concurrent controllable events that may be executed at a given state must be generated. Otherwise, the controller would not be deterministic, i.e. each of its outputs would not be uniquely defined for a given sequence of inputs. Different criteria have been proposed to resolve the choice issue [Lop12; Mal02; Die02; Lea12], with no established consensus. On the other hand, when an uncontrollable event and a controllable event are concurrent, the fixed criteria is to rank first the uncontrollable event in the control program so as to execute it first.

### 3.5.3 Multilevel implementation

Previous works, however, are often applied to manufacturing systems, which are purely event-driven systems and thus the proposed methods are often not well adapted to hybrid system applications [Kou00], where the continuous-time signals need to be abstracted so as to isolate their discrete behavior. Hence, a control system similar to the one presented in [Vie17], where three levels can be distinguished (Figure 3.13), would be more appropriate:

- *Modular Supervisors*: This level is composed of the set of modular supervisors, as well as a disabling signals function, which joins the disabling signals for the same event generated by the various supervisors. Thus, the MS level receives the signals associated to the *events* in its alphabet from the Product System (PS) level, and sends back the *disabling signals* associated to the controllable events.
- *Product System*: This level comprises the set of modular plants, as well as a logical function, which ensures that only one event is treated at a time. The PS level communicates the signals associated to each event in its alphabet to the MS, and receives in return the




(a) Forcing event approach (b) Supervised control approach

Figure 3.12 – Approaches for the implementation of automata

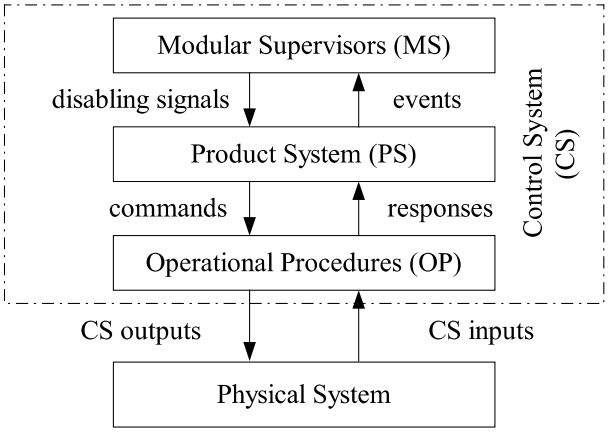



Figure 3.13 – Control architecture proposed in [Vie17]

corresponding disabling signals. Then, once a controllable event is authorized, the PS generates the associated *commands* and sends it to the Operational Procedures (OP) level, while receiving the *responses* associated to the uncontrollable events by the physical system from the OP level.

- *Operational Procedures*: This level consists of a set of functions called Operational Procedures that detail the low-level control of each equipment in the physical plants. Each OP associates itself with a unique controllable event and corresponds to the activities related to a given task. Eventually, a unique or several uncontrollable events might also be associated to an OP. Hence, the OP level acts as an interface between the physical system and the control system. The *CS inputs* and *CS outputs* are respectively determined according to the sensors and the actuators constituting each individual equipment. Then, the OP level transmits the corresponding system responses to the PS level and attributes a given value to the actuators signals upon the reception of a command.

Therefore, in order for the different control levels to communicate, an input and an output alphabet is attached to the supervisor and plant automata, which transforms them into Moore or Mealy variants. However, the transition functions of the automata may need to be modified so that the Moore or Mealy machines remain deterministic when the I/O alphabets are added. Thus, two algorithms, given in Appendix D, that convert each plant automaton $G_x$ into an equivalent[2] automaton $\ddot{G}_x$ in view of their transformation into Moore machines are defined in [Vie17]. Thereby, the algorithm 1 converts each $G_x$ plant model into an equivalent automaton $\dot{G}_x$ where all states with self-loop transitions have been replaced by a pair of equivalent states. Therefore, the outputs associated to the events labeling the self-loop transitions are differentiated from those associated to the event "entering" the state. As for the algorithm 2, it converts the automaton $\dot{G}_x$ into an equivalent automaton $\ddot{G}_x$ where every transition reaching a particular state of $\ddot{G}_x$ is always caused by an occurrence of the same event. Consequently, each state associates

[2]Two automata are said to be equivalent when both their generated and marked languages are equivalent.

itself with a single event in $\ddot{G}_x$. The supervisor automata, on the other hand, do not need any specific modification before their transformation into Moore machines, as the disabling signals generated in the states, which form the output alphabet of the Moore machine, are independent of the input alphabet.

Also, because of the particularity of the example presented in [Vie17] (where each plant represents a machine that starts a task and waits until the task is finished), the choice issue is not treated by the authors as there are no concurrent controllable events in the plants.

Finally, once the considered automata are converted into Moore machines, they can be implemented in the language chosen by the developer. For PLC applications, the international standard IEC 61131-3 [Int03] establishes the programming principles to be followed and defines the languages to be used, namely: Structured Text (ST), Instruction List (IL), Ladder Diagram (LD), and Sequential Function Chart (SFC). Because most of the articles treating the SCT are related to the PLC-based control of manufacturing systems, most of the works in the literature try to adapt the SCT framework to these programming languages [Zay17], especially to the most popular graphical languages: Ladder Diagram (LD) and Sequential Function Chart (SFC). While the methods developed in [Lau97; Fab98; Que02; Gou04; Lea12] are all based on LD, those presented in [Cha99; Vie17] are SFC-based. The particular syntax of those languages, however, offers little portability for the proposed methods to be used outside PLC applications, which is likely to be the case for HVDC systems. Given the need for a highly responsive and customized control, common-user oriented languages like C code might be more appropriate [Eco08], since they allow to integrate homogeneously the discrete-event controllers and the continuous-time controllers of the grid.

## 3.6 Conclusion

Given the need for an automated supervisory control that coordinates the protection and control schemes of an HVDC grid via event-driven commands in diverse situations of the system, while preventing new negative interactions from arising, a control design method based on automata modeling and SCT-based supervisor synthesis is retained in this PhD thesis. Indeed, the formal framework presented in this chapter allows to specify the requirements to be respected by the desired supervisory control, thus preventing the proposed solution from creating failures in the HVDC system.

Furthermore, the use of the SCT allows to adapt the architecture of the supervisory control to the corresponding application. For instance, given the distributed nature of HVDC grids (converter stations located throughout the grid), their hierarchical control structure (local protection and control schemes interacting with a control center) and the diverse situations experienced by the system during its operation (fault treatment, power balance restoration, etc.), a dedicated supervisory control should be designed by horizontal, vertical and modal decomposition

approaches. Concerning the vertical decomposition, the compositional approach is preferred as it allows to define several levels of control during the design phase and is not limited by a predefined structure. This aspect is particularly interesting for HVDC systems. Indeed, the lack of existing MTDC grids demands for a flexible design method that integrates new aspects with the minimum impact on the existing structure, given that additional control levels might be necessary in the future as the comprehension of the system increases. As for the horizontal decomposition, the decentralized approach is preferred as it allows to enforce the specifications in the system (involving interconnected converter stations and common phenomena) by means of communication. However, a thorough analysis of the communication requirements of HVDC systems should be performed before the vertical decomposition of the supervisory control is treated, which is out of the scope of this PhD thesis. Thus, only the vertical and modal decomposition are treated in our work.

Finally, several methods are proposed in the literature for the practical implementation of the supervisory control architecture synthesized by means of the SCT. However, almost the entirety of the related works are applied to the control of manufacturing systems, which can be considered as pure DES. Thus, the proposed control structures are not always adapted to the control of hybrid systems, such as HVDC grids. In addition, the methods in the literature try to adapt the proposed solutions to PLC-related implementation languages. However, because of the fast coordination needed from the supervisory control of an HVDC system, PLCs are not likely to be used. In consequence, an implementation method that adapts the control architecture obtained with the SCT to the specificity of HVDC systems is proposed in the next chapter.

# 4

# Supervisory control: design and implementation

## Contents

## 4.1 Introduction

As pointed out in previous chapters, the high reliability and the fast coordination of actions required for the operation of an HVDC system calls for an automated supervision of the system based on discrete-event control. Because of the hybrid nature of HVDC systems, however, the discrete-event control needs to interact with the physical system via an interface, as illustrated in Figure 4.1. This interface translates the continuous-time measurements received from the sensors to discrete events. Inversely, it associates a set of signals sent to the continuous-time control loops and the protection relays of the system (which ultimately enforce the control and protection actions), to the discrete-event commands given by the supervisory control. Thus, the discrete-event control along with the dedicated interface constitutes the ASC, while the physical system and the associated control, protection and meter devices compose the plant to be supervised. In this chapter, a systematic approach for the design and implementation of such a supervisory control is presented. The proposed design and implementation method (illustrated in Figure 4.2 in the form of an UML activity diagram) is inspired from the compositional approach, which constructs the necessary models at different levels of abstraction.

The first step concerns the modeling of the generic behavior of the components in a converter station (simply referred to as "station" in the following) (Section 4.2) in order to be able to synthesize a supervisory control later on. To do this, it is first necessary to conduct a functional analysis, given in Section 4.2.1 and Section 4.3.1, of the plant so as to establish a com-

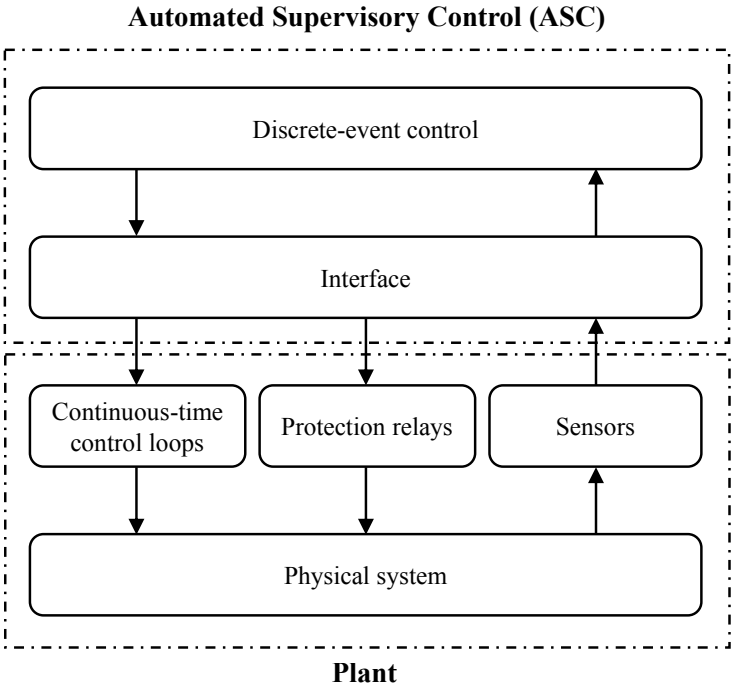



Figure 4.1 – Supervisory control structure in an hybrid system

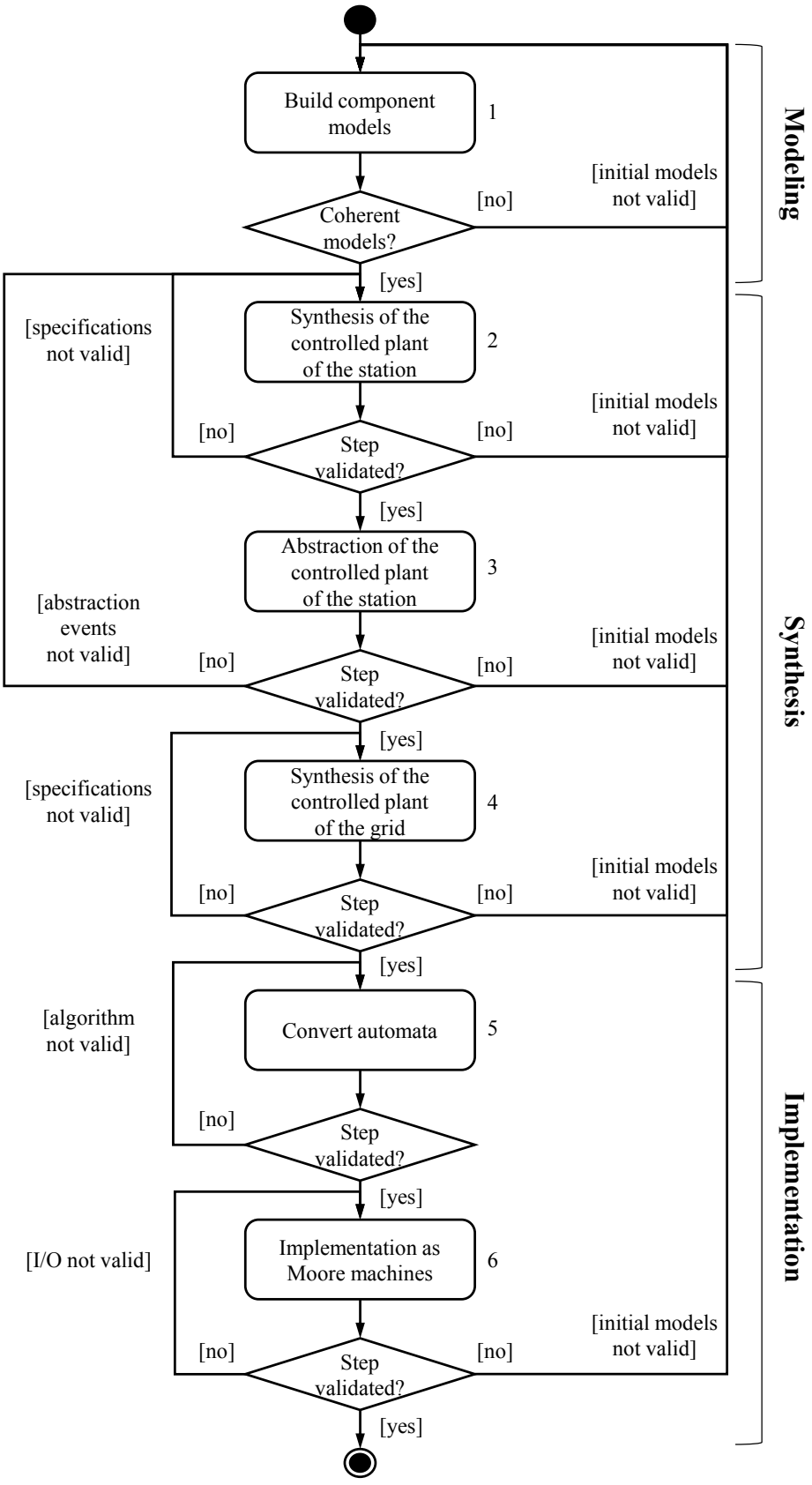



Figure 4.2 – Proposed design and implementation approach

mon framework between the electrical engineering experts and the ASC designer that allows to correctly model the system components. Step 2 concerns the construction of the controlled plant of a station by means of synthesis, such as defined in the SCT (Section 4.3.2). During Step 3, the obtained models are abstracted according to the compositional approach, so that the abstracted models have a reduced size while being equivalent from a supervision perspective (Section 4.3.2). These reduced models are then combined in Step 4 so as to model the grid's plant and synthesize the corresponding supervisor (Section 4.3.3). Finally, in the implementation stage (Section 4.4), the designed automata are first converted in Step 5 by means of the algorithms given in Appendix D so as to obtain language-equivalent automata where each state is reached by a unique event. Then, an I/O alphabet is added to the converted models in Step 6 in order to obtain the Moore machines that constitute the discrete-event control.

## 4.2 Generic component models

Although the number of components composing each subsystem in Figure 2.9 might vary depending on the grid's configuration (cf. Section 2.3), the type of components used remains almost invariable for any MMC-based HVDC system. Hence, in terms of supervision, it is advantageous to consider an object-oriented approach for the modeling of the component's behavior: the behavior inherent to the component can be considered in a generic model of the component, whereas the behavior related to the configuration of the system or to a specific operational situation is treated in an instance of the generic model.

In this section, a functional and monitoring analysis is performed first on the components in each station subsystem in Figure 2.9 so as to identify their generic behavior (Step 1 of Figure 4.2). The proposed functional and monitoring analysis are intended to establish a standardized method for the study of future components, allowing to set a collaboration framework between the system engineers and the ASC designer. The results of the above analysis are then used for the construction of the generic component models under the form of automata.

### 4.2.1 Functional and monitoring analysis

The following functional and monitoring analysis aims to identify the generic functionalities of the main components in a station in view of their interaction with the supervisory control. Therefore, while the functional analysis highlights how the supervisory control can manage the system components (via the continuous-time controllers and the protection relays) so as to enforce the resulting control law; the monitoring analysis identifies the qualitative Regions of Operation (ROs) in which the measurements from the converter's and cable's sensors can be partitioned, so as to provide the supervisory control with the information necessary for the latter to be aware of the state of the system.

Because the physical components and the associated control, protection and meter devices are regarded as a unique plant by the supervisory control, the converter and the associated control loops and sensors are simply referred to as the converter in the following. Similarly, a circuit breaker and its associated relay will be referred to as a circuit breaker and the term cable will designate the cable itself along with the corresponding sensors.

Without loss of generality, the analyses presented in this section can be extrapolated to other entities in interaction with the supervisory control, such as the dispatch controller sending the calculated references and control parameters to the station components.

### Converter

Each station component carries out a set of basic functions, which we refer to as service functions (SFs), within the operation of the system. For instance, the continuous-time control loops of a converter regulate the continuous-time variables of the system to the control references provided by the dispatch controller through their joint actions. The main functions carried out by the converter controller are identified as: *active power regulation*, *DC voltage regulation*, *internal energy regulation* and *reactive power regulation*. These service functions are accomplished through a series of basic Function Blocks (FBs) that contain the corresponding control loops and algorithms. In Figure 4.3, a zoom is made on the outer loops of Figure 2.13 in order to illustrate the functional arrangement of the outer loops. The FBs can then be enumerated depending on the service function they are associated with: FB 1.1 is used for active power regulation, FB 2.1 and FB 2.2 for DC voltage regulation, FB 3.1 and FB 3.2 for internal energy control, FB 4.1 and FB 4.2 for reactive power regulation.

However, there exist restrictions on the concurrent action of the different FBs. For example, an individual MMC can only regulate the active power or the DC voltage. Thus, if a converter is selected to regulate the DC voltage, it can accomplish it either by means of FB2.1 (a control method where the voltage regulation is realized solely by one station) or by means of FB2.2 (a control method where all the stations participate in maintaining the DC voltage around its rated value). In opposition, if the same converter is selected to regulate the active power, the FBs associated to the DC voltage regulation need to be deactivated. Similarly, the reactive power is either regulated to a reference value (FB4.1) or adjusted to maintain the voltage of the AC side of the station (FB4.2). Furthermore, the amount of energy stored in the MMC is regulated by adjusting the difference between the power entering and leaving the station. Whether the power enters the station by the AC or the DC side needs to be selected as well. In addition, the control loops presented above have an influence on the system only if the whole converter controller is activated (or deblocked), whereas if it is deactivated (or blocked), the control signal sent to the IGBTs of the SMs is zero, meaning the converter is not controllable.

In consequence, the configuration of the converter controller is determined by the possible combinations of concurrent FBs, which might be more or less appropriate depending on the

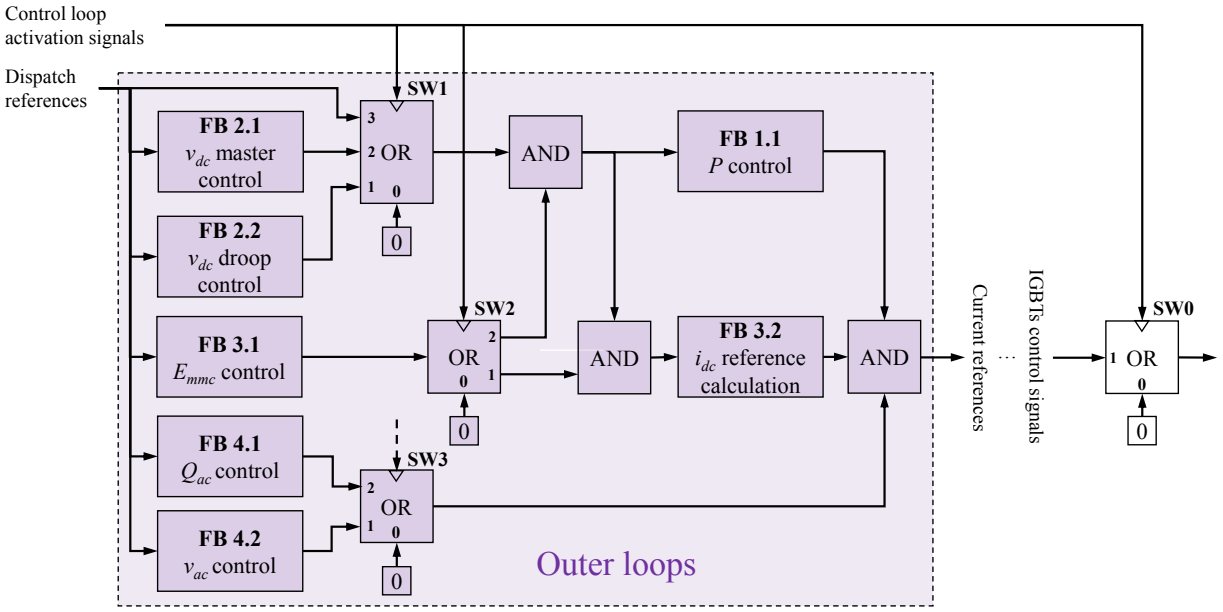



Figure 4.3 – FBs of the MMC controller

situation of the system. Thus, it is up to the supervisory control to manage the activation of the different control loops for each operating condition. As illustrated in Figure 4.3, a switch can be placed in the functional diagram whenever there exists a restriction on the concurrent operation of two or more FBs. The supervisory control can then activate the appropriate FB by

| SW0 | SW1 | SW2 | SW3 | Configuration |
|---|---|---|---|---|
| 0 | x | x | x | MMC_Blocked |
| 1 | 0 | 0 | 0 | MMC_Deblocked |
| | 0 | 1,2 | 1,2 | MMC_P_Control_Deactivated |
| | 1 | - | - | MMC_Vdc_Droop_Control |
| | 2 | - | - | MMC_Vdc_Master_Control |
| | 3 | - | - | MMC_P_Slave_Control |
| | 1,2,3 | 0 | 1,2 | MMC_E_Control_Deactivated |
| | - | 1 | - | MMC_Edc_Control |
| | - | 2 | - | MMC_Eac_Control |
| | 1,2,3 | 1,2 | 0 | MMC_Q_Control_Deactivated |
| | - | - | 1 | MMC_Qac_Control |
| | - | - | 2 | MMC_Vac_Control |

Table 4.1 – Configurations of the MMC controller

selecting the switch position corresponding to it. Eventually, if a switch is in the zero position, the corresponding variable is not regulated.

In summary, taking into account the activation constraints of the FBs, the control configurations of an MMC are determined by the position of the four independent switches, i.e. (SW0, SW1, SW2, SW3), given in Table 4.1. Whenever a switch position is irrelevant because another switch invalidates its effect, the corresponding cell is marked with a cross. Also, as each independent switch is analyzed separately, the cells not related to the considered switch are marked with an hyphen.

For the supervisory control to discriminate automatically the appropriate configuration of the controller, however, a continuous monitoring of the converter is necessary. Because the operating status of an MMC is strongly dependent on the total energy stored in the SMs, the former is mostly determined by the evolution of the MMC's inner voltage $v_{mmc}$, defined as the average of the six total arm voltages. The continuous state-space in which this variable evolves can be partitioned into qualitative states, referred to as regions of operation, as illustrated in Figure 4.4.

Hence, the RO 1 and RO 0 of the MMC voltage describe respectively whether the SM capacitors have been charged sufficiently to synthesize an AC voltage at least equivalent to the AC grid's Root Mean Square (RMS) voltage or not, i.e. if it is possible to activate the controllers or not, as pointed out in Section 2.3.2. Although the limit between both ROs depends on the topology (symmetric or asymmetric) of the HVDC link, this limit is commonly chosen between 70% and 80% of the rated DC voltage [Shi17]. The RO 2, on the other hand, describes that the

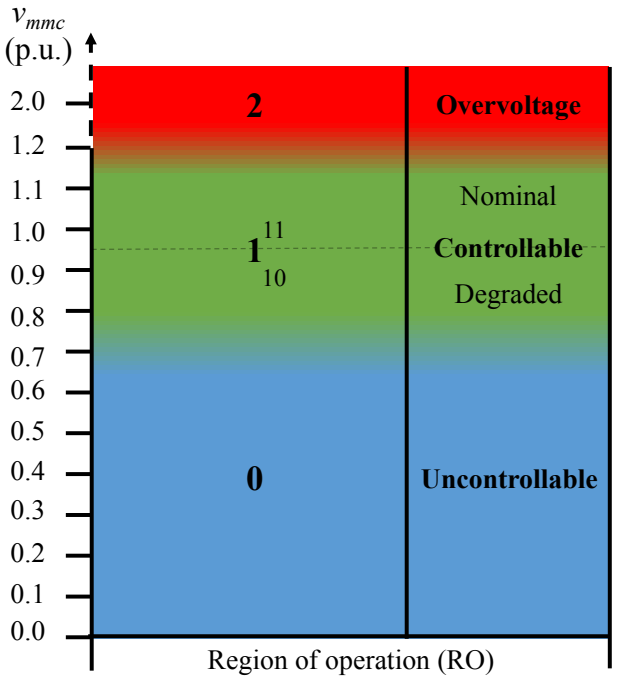


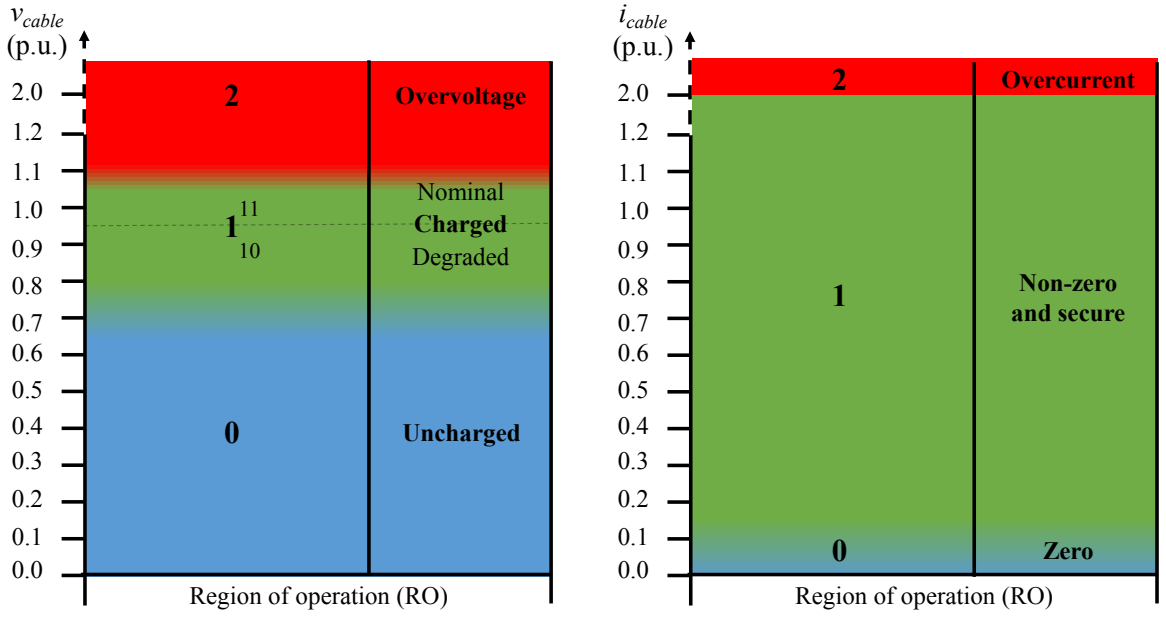

Figure 4.4 – ROs of the converter voltage

MMC is working in overvoltage and so the component has a high probability of failure. The limit between RO 1 and RO 2 depends on the voltage supported by the SM capacitors and is considered to be around 120% of the rated voltage [Shi17].

For supervision purposes, however, intermediate qualitative states can be defined within each RO. For instance, two separate qualitative states are identified in the RO 1 of $v_{mmc}$: the state 10 corresponds to a controllable operation of the MMC with no margin for reactive power regulation while in state 11 the voltage is higher than a predefined threshold marking the minimum level for reactive power regulation. In consequence, the state 10 corresponds to a controllable but degraded operation of the MMC while the state 11 corresponds to a controllable and nominal operation of the same. Consequently, the quantitative continuous-time values of the converter voltage have been grouped into discrete and qualitative regions of operation which can then be processed by the supervisory control.

### Cable

Given that the regulation of the cable variables is realized via the converter controller, no functional analysis is needed for this component. However, it is necessary to perform a monitoring analysis so as to be able to identify the state of the cable at each time instant. Ultimately, the state of the cable can be inferred from the observation of two variables: the DC voltage and the DC current. Indeed, the DC power transferred by a cable can be inferred from the product between these two variables. Then, similarly to the MMC inner voltage, their continuous-time values are partitioned into qualitative regions of operation, as shown in Figure 4.5.




(a) Cable voltage

(b) Cable current

Figure 4.5 – ROs of the cable variables

Concerning the ROs of the DC voltage in the cable, because the DC voltage is created by means of the converter controller, RO 1 expresses the fact the MMC has been charged to a level where it can operate, in opposition to RO 0. Thus, the RO 10 that corresponds to a degraded level of the cable's voltage and the RO 11 corresponding to a nominal level of the same should be differentiated similarly to the MMC's voltage. Finally, RO 2 describes an overvoltage situation, similarly to the MMC's voltage. The limit between RO 1 and RO 2 is determined by the maximum voltage level supported by the DC cable. Typically, an increase of the operating voltage by 8 to 10% reduces the service life by half [Bru06].

To conclude, the state of the DC current in the cable is described through three regions of operation, depending on its amplitude level. In RO 0, for instance, the DC current circulating through the cable is null. However, in order to prevent possible malfunction of the supervisory control due to the current ripple, the limit between RO 0 and RO 1 is arbitrarily set beyond zero. In this thesis, the limit is set at 0.05% of the rated current's value. In consequence, RO 1 describes the fact that the level of the DC current is non-zero and that no physical limit of the cable is violated, as opposed to RO 2. The limit between RO 1 and RO 2 depends on the thermal limit of the cable, but is commonly considered to be 200% of the rated current [Aur17].

### Circuit breaker

The circuit breakers and the rest of devices in the breaking and Pre-Insertion Resistor (PIR) modules (cf. Figure 2.14) controlled by means of a dedicated protective relay typically perform *fault prevention*, *fault identification* and *fault isolation* functions. Therefore, a functional analysis similar to the one presented for the converter should be carried out so as to arrange the algorithms and basic function blocks that allow to realize these functions according to their activation constraints and identify in consequence the points of interaction with the supervisory control. However, because the protection subsystem is designed to respond autonomously to the appearance of short-circuit faults in the system, it is not within the scope of this PhD thesis to thoroughly analyze its inner structure. Thus, we consider in our work that it is possible for the supervisory control or any external agent such as a human operator to interact with the relay and request the closing or the opening of the circuit breaker. The relay then analyzes whether the current and voltage conditions around the CB would damage the physical device or not, and accepts or rejects the request in consequence. Furthermore, in order to simplify the modeling, the high-speed switch is not considered in our study and the PIR module is considered to be dependent on the state of the circuit breaker.

In consequence, the generic behavior of the devices in Figure 2.14 is reduced in our work to a simplified behavior of the circuit breaker and the associated relay. However, an exhaustive functional analysis as proposed in this section for converters and cables certainly needs to be realized if the supervisory control is expected to have a larger influence on the decision-making process of the relay.

### 4.2.2 Model construction

The generic models of the component plants are to be obtained through the modeling of the control-related behavior and/or the physical-related behavior of each different component $C_i$ in a station, with $i \in \mathbb{N}$. The control-related behavior is to be modeled when necessary by relating a discrete event to each value that a digital signal interacting with the supervisory control may take, as described in the functional analysis. On the other hand, the physical behavior corresponds to a continuous-time evolution of the variables of the system (voltage, current, etc.) identified in the monitoring analysis. These continuous-time dynamics can be abstracted so as to model the underlying qualitative states and the corresponding events labeling the transitions between them, as defined below [Vás16]:

***Definition 4.1 (Event generation by qualitative abstraction of the continuous behavior)*** *Consider a continuous-time variable $x$ whose continuous state-space can be partitioned into a set, denoted by $D(X)$, of qualitative states limited by hypersurfaces that constitute the associated qualitative variable $X$. The domain $D(X)$ of $X$ is obtained through the function $f_{qual} : D(x) \to D(X)$ that maps the continuous values of $x$ to the qualitative states. Thus, for a hypersurface $h_{ij}(x)$ $(i, j = 1, \ldots, n$ and $i \neq j)$, $f_{qual}(x)$ is defined such that:*

$$f_{qual}(x) := \begin{cases} q_i & \text{if } x < h_{ij}(x) \\ q_j & \text{if } x \geqslant h_{ij}(x) \end{cases}$$

*The set of qualitative states is then defined as the union of all the states generated by $f_{qual}$ such that $X = \{q_1, \ldots, q_n\}$. The change of value of $X$ is defined by the transitions between the qualitative states, such that $\Sigma_X$ is the finite set of events associated to the transitions and $\delta : X \times \Sigma_X \to X$ is the partial transition function. The corresponding event generator is defined by the function $f_{X \to \sigma}$, such that $f_{X \to \sigma} : X \times \delta(X, \Sigma_X) \to \Sigma_X$. Considering a pair of qualitative states $\{q_i, q_j\}$, $f_{X \to \sigma}$ is defined as follows:*

$$\{\forall (i, j) \in (1, \ldots, n) | i \neq j\} \to f_{X \to \sigma} := \begin{cases} e_i & \text{if } q_j \to q_i \\ e_j & \text{if } q_i \to q_j \end{cases}$$

*The set of generated events corresponds then to the union of all the events labeling the transitions between two qualitative states limited by a hypersurface $h_{ij}(x)$, such that $\Sigma_X = \{e_1, \ldots, e_n\}$.* ♦

The generic behavior of each component $C_i$ is then modeled under the form of a generic automaton $G^{C_i}$ that generates a language $L(G^{C_i})$ covering the different behaviors of the component independently of the operating condition of the system, according to the following definition:

***Definition 4.2 (Generic model of a component)***
*The generic behavior of a component $C_i$ is modeled by an automaton $G^{C_i} = (Q^{C_i}, \Sigma^{C_i}, \delta^{C_i}, q_0^{C_i}, Q_m^{C_i})$, where:*

- *$Q^{C_i}$ is the set of states;*
- *$\Sigma^{C_i}$ is the set of component events $C_i$ and includes the disjoint subsets $\Sigma_c$ and $\Sigma_{uc}$ of the controllable and uncontrollable events of $C_i$, i.e. $\Sigma^{C_i} = \Sigma_c^{C_i} \cup \Sigma_{uc}^{C_i}$ and $\Sigma_c^{C_i} \cap \Sigma_{uc}^{C_i} = \emptyset$;*
- *$\delta^{C_i}$ is the partial transition function, defined by $\delta^{C_i} : Q^{C_i} \times \Sigma^{C_i} \to Q^{C_i}$;*
- *$q_0^{C_i}$ is the initial state, such that $q_0^{C_i} \in Q^{C_i}$;*
- *$Q_m^{C_i}$ is the set of marked states, such that $Q_m^{C_i} \subseteq Q^{C_i}$.* ♦

By definition, no particular operating condition is considered during the modeling of the generic behavior of a component and so all the states of $G^{C_i}$ are marked. Furthermore, in order for the automaton to generate a language, an initial state is to be chosen by the designer when modeling the generic behavior of $C_i$ (cf. Definition 3.1). The set $Q_m^{C_i}$ of marked states, as well as the initial state $q_0^{C_i}$, can be later modified by the designer when instantiating $G^{C_i}$ in a specific situation of the system. In addition, the alphabet of each $G^{C_i}$ is specific to the type of component.

### Converter

The qualitative states identified in Figure 4.4 for the converter's voltage, i.e. (U)ncontrollable, (D)egraded and controllable, (N)ominal and controllable and (O)vervoltage, can then be defined with respect to the hypersurfaces limiting each of them (4.1), as illustrated in Figure 4.6:

$$f_{qual}(v_{mmc}) = \begin{cases} \text{U} & \text{if } v_{mmc} < h_{01}(v_{mmc}) \\ \text{D} & \text{if } h_{01}(v_{mmc}) \leq v_{mmc} < h_{11}(v_{mmc}) \\ \text{N} & \text{if } h_{11}(v_{mmc}) \leq v_{mmc} < h_{12}(v_{mmc}) \\ \text{O} & \text{if } v_{mmc} \geq h_{12}(v_{mmc}) \end{cases} \tag{4.1}$$

All the qualitative states of the continuous-time voltage in the MMC are then included in the set of states $Q_{ph}^{mmc} = \{\text{U}, \text{D}, \text{N}, \text{O}\}$ of the automaton $G_{ph}^{mmc} = (Q_{ph}^{mmc}, \Sigma_{ph}^{mmc}, \delta_{ph}^{mmc}, q_{0,ph}^{mmc}, Q_{m,ph}^{mmc})$ modeling the physical behavior of the converter, such that $Q_{m,ph}^{mmc} = Q_{ph}^{mmc}$ and $q_{0,ph}^{mmc} = \text{U}$. The set of events $\Sigma_{ph}^{mmc} = \{\text{MMC\_V0}, \text{MMC\_V10}, \text{MMC\_V11}, \text{MMC\_V2}\}$ that label the different transitions between the states generated in (4.1) are then obtained via the abstraction function in (4.2), as shown in Figure 4.6.

$$f_{Q_{ph}^{mmc} \to \Sigma_{ph}^{mmc}} = \begin{cases} \text{MMC\_V0} & \text{if D} \to \text{U} \\ \text{MMC\_V10} & \text{if U} \to \text{D} \vee \text{N} \to \text{D} \\ \text{MMC\_V11} & \text{if D} \to \text{N} \vee \text{O} \to \text{N} \\ \text{MMC\_V2} & \text{if N} \to \text{O} \end{cases} \tag{4.2}$$

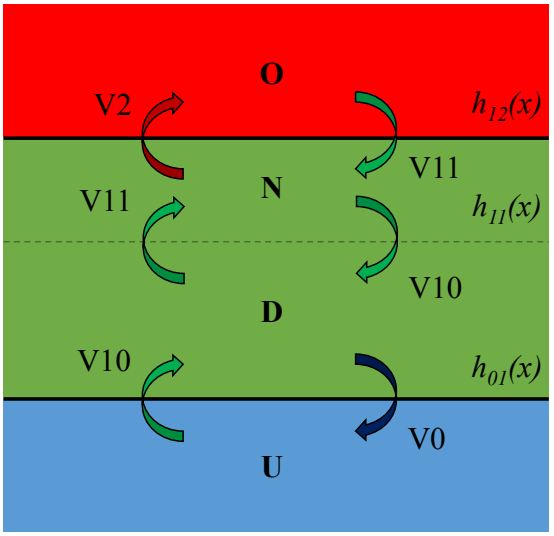



Figure 4.6 – Qualitative states and associated transitions of $v_{mmc}$

Because the events generated in (4.2) are associated to the physical behavior of the MMC and dependent on the MMC voltage measurements, they are all uncontrollable, i.e. $\Sigma_{ph}^{mmc} = \Sigma_{uc,ph}^{mmc}$, and $\Sigma_{c,ph}^{mmc} = \emptyset$. The qualitative physical behavior of the MMC is then modeled by the automaton $G_{ph}^{mmc}$ shown in Figure 4.7. The uncontrollable events are preceded by an exclamation mark and any transition labeled by at least one uncontrollable event is pictured in red.

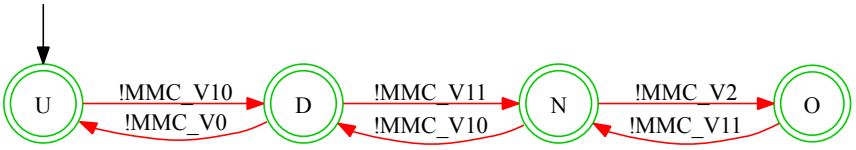



Figure 4.7 – $G_{ph}^{mmc}$

Secondly, the behavior related to the MMC controller is modeled by an automaton $G_{ctrl}^{mmc} = (Q_{ctrl}^{mmc}, \Sigma_{ctrl}^{mmc}, \delta_{ctrl}^{mmc}, q_{0,ctrl}^{mmc}, Q_{m,ctrl}^{mmc})$. The set of states $Q_{ctrl}^{mmc}$ of this automaton can be inferred from the controller configurations identified during the functional analysis, summarized in Table 4.1. As explained in the previous section, the controller can be in two possible states: either it is (B)locked and no control loop can be activated, or it is (D)eblocked and in that case the rest of configurations have an influence on the system (but not on the controller itself, which remains deblocked). The set of states of the MMC controller is thus defined to be $Q_{ctrl}^{mmc} = \{\mathrm{B}, \mathrm{D}\}$.

Because each controller configuration corresponds to a particular position of the switches in Figure 4.3, which can be modified by the supervisory control, each event related to the MMC controller is defined as the particular change in the activation signals that allows to reach a particular configuration (i.e. a particular switch position). For example, the “MMC_Deblock” event allows to set the MMC controller as deblocked, while the “MMC_Block” event puts the

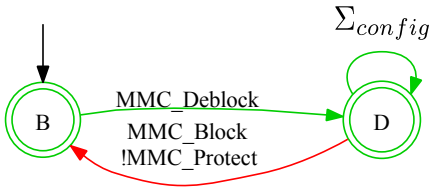



Figure 4.8 – $G^{mmc}_{ctrl}$

control-loops in a blocked state. Similarly, the rest of configuration events allow the supervisory control to modify the switches position in order to impose a given control configuration (MMC_Vdc_Droop_Control, MMC_P_Slave_Control, MMC_Eac_Control, etc.).

All these events are controllable and can be disabled by the ASC, with the exception of the "MMC_Protect" event, which is generated by an internal protection algorithm of the MMC that blocks the control-loops once the current inside the MMC arms exceeds a predefined security threshold. In this way, the IGBTs and the SM capacitor are protected in case of a fault in the cable as the current is forced to circulate through the anti-parallel diodes in the SMs (see Figure 2.11).

The totality of these events are included in the alphabet $\Sigma^{mmc}_{ctrl}$, such that $\Sigma^{mmc}_{ctrl} = \Sigma^{mmc}_{uc,ctrl} \cup \Sigma^{mmc}_{c,ctrl}$, with $\Sigma^{mmc}_{uc,ctrl} = \{\text{MMC\_Protect}\}$ and $\Sigma^{mmc}_{c,ctrl} = \Sigma^{mmc}_{ctrl} \setminus \Sigma^{mmc}_{uc,ctrl}$. The controller behavior is then modeled by means of the plant automaton $G^{mmc}_{ctrl}$ given in Figure 4.8, such that $Q^{mmc}_{m,ctrl} = Q^{mmc}_{ctrl}$ and $q^{mmc}_{0,ctrl} = \text{B}$. The events in $\Sigma^{mmc}_{ctrl}$ can in turn be divided into two subsets: the subset $\Sigma_{config}$ that comprises the events determining a control configuration, without modifying the state of the controller; and the subset $\Sigma_{activ} = \{\text{MMC\_Deblock}, \text{MMC\_Block}, \text{MMC\_Protect}\}$ that comprises the events blocking or deblocking the controller. In consequence, $\Sigma^{mmc}_{ctrl} = \Sigma_{activ} \cup \Sigma_{config}$ with $\Sigma_{activ} \bigcap \Sigma_{config} = \emptyset$.

At last, the complete behavior of the MMC is defined by the interaction between the converter and its controller, although some physical constraints need to be modeled by means of plant automata. For instance, the converter can only be charged to a value higher than the peak value of the AC grid's voltage if the energy control loops are previously activated. Furthermore, the configuration of the energy control cannot be modified until the nominal voltage level is reached or until the converter returns to the initial discharged condition (automaton $G^{mmc}_{pc1}$ of Figure 4.9a). In consequence, for the nominal voltage level to be reached, the converter controller needs to be first deblocked, as modeled by the automaton $G^{mmc}_{pc2}$ of Figure 4.9b. Finally, for the MMC to be discharged, the converter controller needs to be deblocked first, as the antiparallel diodes in the SMs allow the charging of the SM capacitors but not their discharging (automaton $G^{mmc}_{pc3}$ of Figure 4.9c).

The generic automaton $G^{mmc}$ modeling the global behavior of the MMC is then obtained by means of the parallel composition between the given automata, such that $G^{mmc} = G^{mmc}_{ctrl} \parallel G^{mmc}_{ph} \parallel G^{mmc}_{pc1} \parallel G^{mmc}_{pc2} \parallel G^{mmc}_{pc3}$. Hence, the resulting automaton, given in Figure 4.10, represents

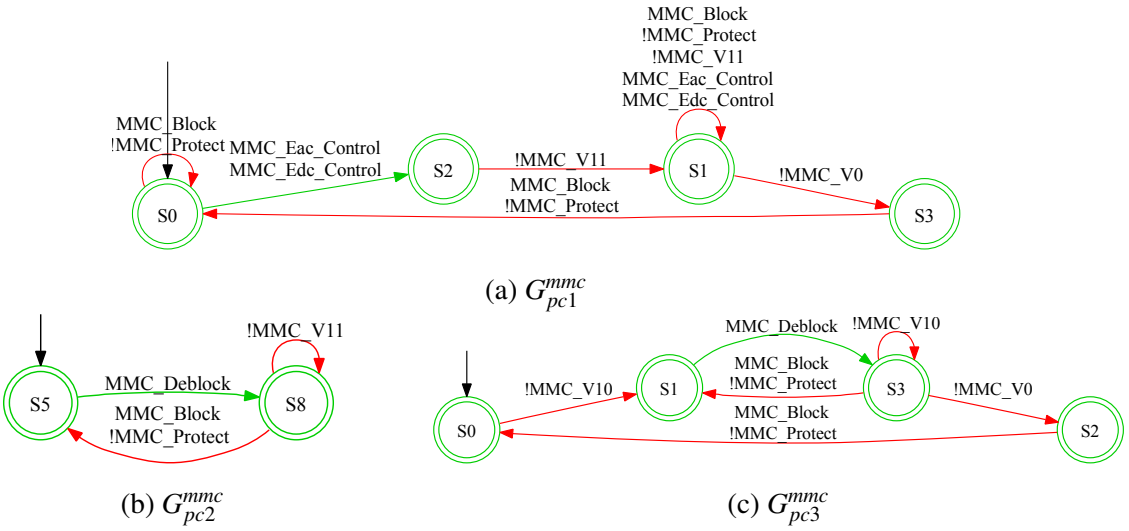



Figure 4.9 – Physical constraints between the MMC and its controller

the generic behavior of the converter without consideration of the operating conditions of the system, such that $Q^{mmc}_m = Q^{mmc}$ and $\Sigma^{mmc} = \Sigma^{mmc}_{ctrl} \cup \Sigma^{mmc}_{ph}$. The initial state $q^{mmc}_0$ is chosen by default to be the state where the converter is blocked and uncharged, but this might be modified depending on the situation of the system, as explained later in this manuscript.

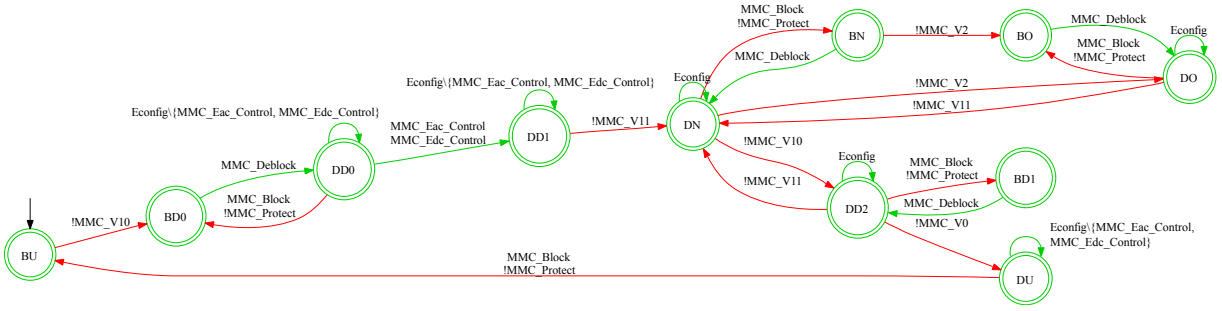
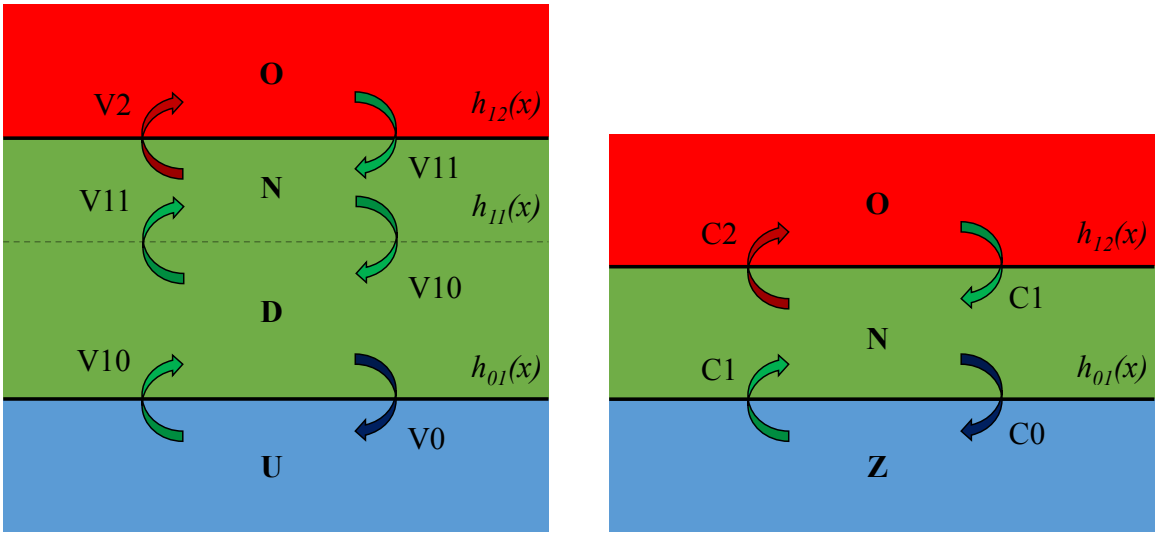



Figure 4.10 – Generic automaton $G^{mmc}$ of an MMC

### Cable

As stated in the monitoring analysis of Section 4.2.1, the behavior of a DC cable is determined by two variables: the DC voltage $v_{cable}$ and the DC current $i_{cable}$. Similarly, to the MMC voltage, the qualitative states of cable's voltage and current can be directly inferred from the hypersurfaces separating the ROs given in Figure 4.5, as shown in Figure 4.11.

Therefore, the states (U)ncharged, (D)egraded, (N)ominal and (O)vervoltage of $v_{cable}$ are




(a) $v_{cable}$ (b) $i_{cable}$

Figure 4.11 – Qualitative states and associated transitions

included in the set of states $Q^{cable}_{v_{cable}} = \{\mathrm{U}, \mathrm{D}, \mathrm{N}, \mathrm{O}\}$, according to the following function:

$$f_{qual}(v_{cable}) = \begin{cases} \mathrm{U} & \text{if } x < h_{01}(x) \\ \mathrm{D} & \text{if } h_{01}(x) \leq x < h_{11}(x) \\ \mathrm{N} & \text{if } h_{11}(x) \leq x < h_{12}(x) \\ \mathrm{O} & \text{if } x \geq h_{12}(x) \end{cases} \tag{4.3}$$

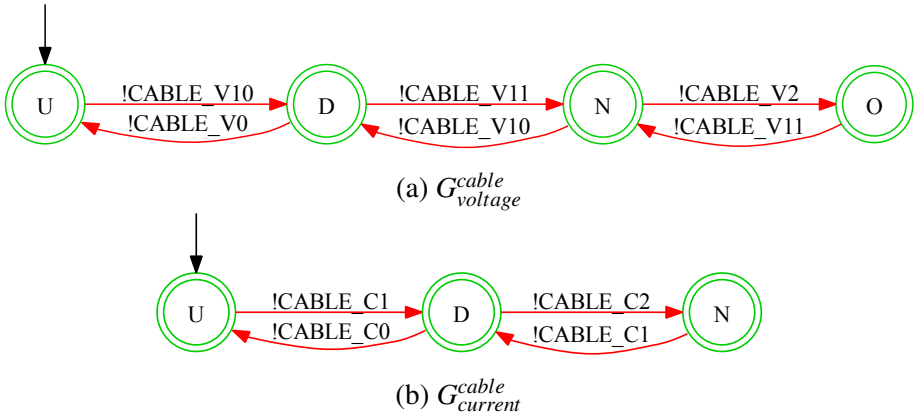
In turn, the set of events $\Sigma^{cable}_{v_{cable}} = \{\mathrm{CABLE\_V0}, \mathrm{CABLE\_V10}, \mathrm{CABLE\_V11}, \mathrm{CABLE\_V2}\}$ associated to the change of state of $v_{cable}$ is generated as defined in (4.4):

$$f_{Q^{cable}_{v_{cable}} \rightarrow \Sigma^{cable}_{v_{cable}}} = \begin{cases} \mathrm{CABLE\_V0} & \text{if } \mathrm{D} \rightarrow \mathrm{U} \\ \mathrm{CABLE\_V10} & \text{if } \mathrm{U} \rightarrow \mathrm{D} \vee \mathrm{N} \rightarrow \mathrm{D} \\ \mathrm{CABLE\_V11} & \text{if } \mathrm{D} \rightarrow \mathrm{N} \vee \mathrm{O} \rightarrow \mathrm{N} \\ \mathrm{CABLE\_V2} & \text{if } \mathrm{N} \rightarrow \mathrm{O} \end{cases} \tag{4.4}$$

As for the DC current, the qualitative states, shown in Figure 4.11b, correspond to those identified in Section 4.2.1, that is, (Z)ero current, (N)on-zero and secure current and (O)vercurrent. These qualitative states are included in the set of states $Q^{cable}_{i_{cable}} = \{\mathrm{Z}, \mathrm{N}, \mathrm{O}\}$ and are formally generated by $f_{qual}(i_{cable})$ as defined in (4.5):

$$f_{qual}(i_{cable}) = \begin{cases} \mathrm{Z} & \text{if } x < h_{01}(x) \\ \mathrm{N} & \text{if } h_{01}(x) \leq x < h_{11}(x) \\ \mathrm{O} & \text{if } x \geq h_{12}(x) \end{cases} \tag{4.5}$$

Then, the set of events $\Sigma^{cable}_{i_{cable}} = \{\mathrm{CABLE\_C0}, \mathrm{CABLE\_C1}, \mathrm{CABLE\_C2}\}$ associated to the




(a) $G^{cable}_{voltage}$

(b) $G^{cable}_{current}$

Figure 4.12 – Automata modeling the voltage and current behavior of a cable

transitions between the states in $Q^{cable}_{i_{cable}}$ are defined as shown in (4.6):

$$f_{Q^{cable}_{i_{cable}} \to \Sigma^{cable}_{i_{cable}}} = \begin{cases} \text{CABLE\_C0} & \text{if N} \to \text{Z} \\ \text{CABLE\_C1} & \text{if Z} \to \text{N} \vee \text{O} \to \text{N} \\ \text{CABLE\_C2} & \text{if N} \to \text{O} \end{cases} \tag{4.6}$$

The transitions of the voltage and current of a cable through the associated qualitative states are thus respectively modeled by the automata $G^{cable}_{v_{cable}} = (Q^{cable}_{v_{cable}}, \Sigma^{cable}_{v_{cable}}, \delta^{cable}_{v_{cable}}, q^{cable}_{0,v_{cable}}, Q^{cable}_{m,v_{cable}})$ and $G^{cable}_{i_{cable}} = (Q^{cable}_{i_{cable}}, \Sigma^{cable}_{i_{cable}}, \delta^{cable}_{i_{cable}}, q^{cable}_{0,i_{cable}}, Q^{cable}_{m,i_{cable}})$ in Figure 4.12, such that $Q^{cable}_{m,v_{cable}} = Q^{cable}_{v_{cable}}$, $q^{cable}_{0,v_{cable}} = \text{U}$ and $Q^{cable}_{m,i_{cable}} = Q^{cable}_{i_{cable}}$, $q^{cable}_{0,i_{cable}} = \text{Z}$. The generic behavior of the cable is then determined by the combined evolution of the voltage and the current, which can be modeled by the automaton $G^{cable} = (Q^{cable}, \Sigma^{cable}, \delta^{cable}, q^{cable}_0, Q^{cable}_m)$ ,shown in Figure 4.13, obtained via the parallel composition of $G^{cable}_{v_{cable}}$ and $G^{cable}_{i_{cable}}$, such that $G^{cable} = G^{cable}_{v_{cable}} \parallel G^{cable}_{i_{cable}}$. The generic

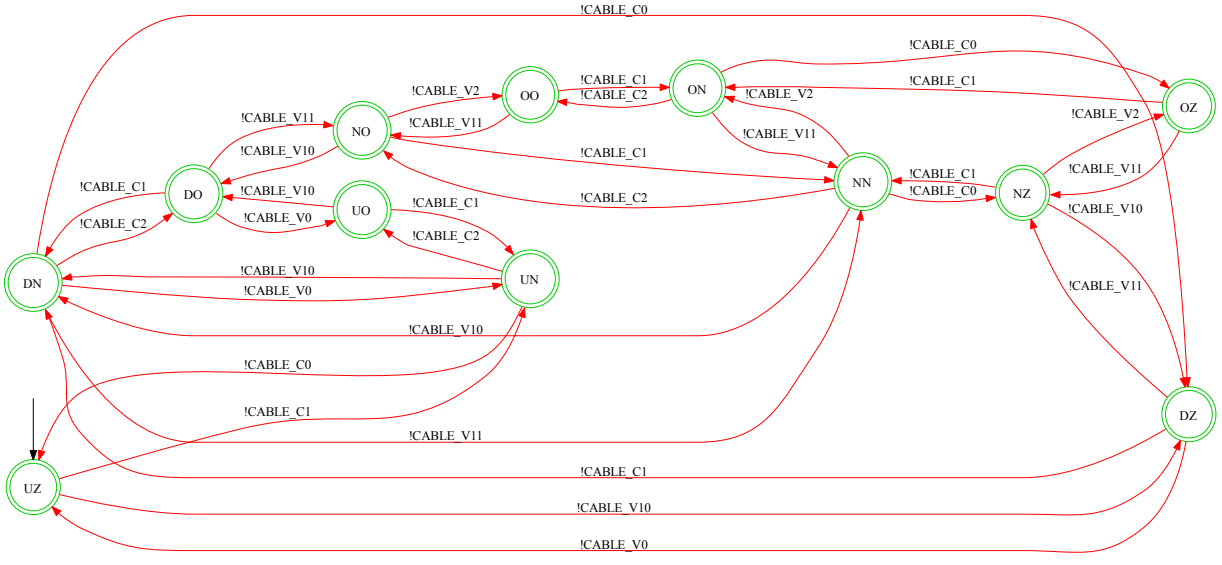



Figure 4.13 – Generic automaton $G^{cable}$ of a cable

automaton $G^{cable}$ is thus defined to be such that $Q^{cable} = Q^{cable}_{v_{cable}} \times Q^{cable}_{i_{cable}}$, $Q^{cable}_m = Q^{cable}$ and $q^{cable}_0 = (q^{cable}_{0,v_{cable}}, q^{cable}_{0,i_{cable}})$. To conclude, because the physical behavior of the cable is modeled for monitoring purposes, all the events in $\Sigma^{cable}$ are uncontrollable as they are generated from the system measurements, such that $\Sigma^{cable} = \Sigma^{cable}_{v_{cable}} \cup \Sigma^{cable}_{i_{cable}}$, with $\Sigma^{cable}_c = \emptyset$.

### Circuit breaker

As stated in the previous section, it is not within the scope of this PhD thesis to thoroughly model the behavior of the plants included in the protection subsystem (cf. Figure 2.14), and we consider it in our work as a simplified behavior of the circuit breaker and the dedicated protective relay.

Therefore, we merely model in the generic automaton $G^{cb}$ of Figure 4.14 the evolution of a circuit breaker's state, which can either be (O)pened or (C)losed, following an Open or Close command from the protective relay. However, it is possible for the supervisory control or any external agent such as a human operator to interact with the relay and request the closing or the opening of the CB. The relay then analyzes whether the current and voltage conditions around the CB would damage the physical device or not, and accepts or rejects the request (i.e. it outputs an Open or Close command) in consequence. Hence, a request does not necessarily imply a change of the state of the breaker, as represented by the associated self-loop transition in $G^{cb}$. The events Open and Close are uncontrollable, as they are supposed to be generated by the protection relay autonomously and thus cannot be disabled by an external controller; whereas the request events RequestOpen and RequestClose are controllable and generated by a control agent external to the protection subsystem (such as the supervisory control).

Thus, $G^{cb}$ is defined by $\Sigma^{cb} = \{\text{CB\_RequestClose}, \text{CB\_RequestOpen}, \text{CB\_Close}, \text{CB\_Open}\}$, $Q^{cb} = \{\text{O}, \text{C}\}$, $Q^{cb}_m = Q^{cb}$ and $q^{cb}_0 = \text{O}$, with $\Sigma^{cb}_c = \{\text{CB\_RequestClose}, \text{CB\_RequestOpen}\}$ and $\Sigma^{cb}_{uc} = \{\text{CB\_Close}, \text{CB\_Open}\}$ such that $\Sigma^{cb} = \Sigma^{cb}_{uc} \cup \Sigma^{cb}_c$.

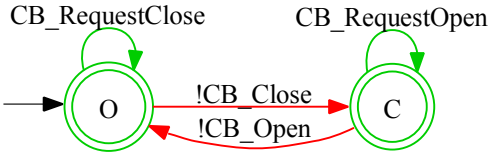



Figure 4.14 – Generic automaton $G^{cb}$ of a circuit breaker

## 4.3 Control synthesis

At this point, the generic behavior of the components comprised in a station has been modeled under the form of automata. The coherence of the obtained models with respect to the reality must be validated by both the ASC designer and the system experts in the common framework

set by the functional and monitoring analysis, as an error in the initial models will be carried over the following phases.

Once the components models have been validated, it is necessary to determine the sequential order in which a set of function blocks should be arranged so that a full sequence of control/protection actions can be realized at the grid level. Then, the specifications to be respected need to be modeled, as well as the possible physical constraints of the system according to the compositional approach, in order for the designer to synthesize a dedicated supervisory control. To facilitate the understanding of the method proposed in this section for the synthesis of a supervisory control, its different steps will be illustrated by a case study corresponding to the start-up of a point-to-point HVDC link, in order to show the models obtained at each step. The example has been chosen to be illustrative enough of future applications while remaining relatively simple and easy to understand.

### 4.3.1 Description of an operational procedure

Once the analysis of the service functions of each component has been realized, it is necessary to determine the order in which a set of FBs should be arranged for a particular sequence of control and/or protection actions, referred to as an operational procedure (OP). In each step of the considered procedure, the system engineer shall describe the function blocks that are activated according to the terminology proposed for the functional and monitoring analysis, so that the description is comprehensible by the supervisory control designer, thus facilitating the construction of the ASC.

⌜ *Example*

The description of an operational procedure is illustrated in the following through the start-up of a point-to-point link [Rom17]. The system considered for our case study is a symmetric monopolar point-to-point HVDC link, illustrated in Figure 4.15, consisting of two stations whose positive and negative DC poles are linked by two cables of opposite voltage polarity. Inside each station, the MMC is connected or disconnected from the AC side of the station via an ACCB, while a PIR module limits the surge current during the uncontrolled charging. On the DC side, the MMC is connected or disconnected from the cables via two DCCBs. The PIR module is not needed in the considered example for the DC side of the station.

The start-up procedure takes place at the beginning of a system operation or after a blackout where the whole HVDC grid has been discharged. The objective is to pre-charge all the MMCs and DC cables of the system to a voltage level that allows the converters to generate an AC sinewave of an amplitude at least equivalent to that of the AC grid. However, as explained in Section 2.3.2, the MMC's rated voltage is chosen to be higher than the minimum required for the MMC to be controllable in order to be able to regulate the reactive power. For this, several strategies can be used. In [Li15], the authors propose to charge the MMCs from the adjacent AC grids, first by an uncontrolled charging where the converter controller is blocked; and then by a controlled charging where the converter controller is deblocked and the voltage and internal

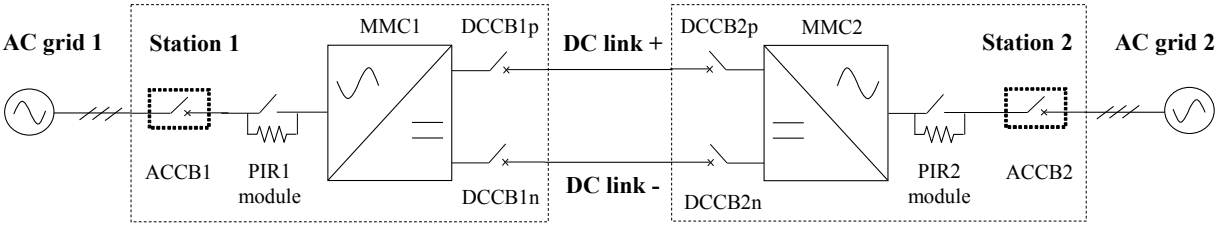



Figure 4.15 – Symmetric monopolar point-to-point link

energy functions activated. Following this approach, the DC cables start to be charged once all the MMCs have been energized. However, not all the adjacent AC grids might be stable enough to provide the energy necessary to charge the MMC. In [Das11], and also in [Li15], the authors propose to energize the MMCs from their DC side, although this can only be realized if the DC links have been previously charged. Therefore, this strategy needs to be combined with an AC-side energization of some of the converters in order to be effective.

Hence, in [Yu13; Gao14], a self-excited charging procedure involving both AC and DC-side energization is proposed. At the beginning, all the circuit breakers (ACCB and DCCBs) of the station are open, and both the capacitors in the MMC and the cables are uncharged. Two stages can then be identified in the start-up procedure: (i) an open-loop charging phase where the MMC is uncontrolled and the capacitors control is blocked; and (ii) after the capacitor has been charged sufficiently to deblock the MMC, a closed-loop charging phase where the MMC capacitors are controlled. Both MMCs in the HVDC link are charged from one AC grid. Thus, two roles can be distinguished depending on their location: the MMC connected to the supplying AC side is called source converter; on the contrary, the MMC charged passively from the DC link and disconnected from its own AC side during the start-up is referred to as the remote converter. The steps of the considered procedure are detailed next. The expected voltage evolution of the arms in the source and remote MMCs is shown in Figure 4.16a, along the different steps of the presented start-up procedure (Figure 4.16b to Figure 4.16e).

1. Because both MMCs are disconnected from their adjacent AC grids, it is first necessary to close all the DCCBs in the positive (DCCB1p, DCCB2p) and negative (DCCB1n, DCCB2n) poles so that the remote converter and the cables can be charged at the same time when the source converter is charged. The PIR2 module of the remote station is bypassed as the station is disconnected from its AC grid (Figure 4.16b).

2. Then, the source MMC is connected to its AC side. It is up to the operator to determine which AC grid is to be used to charge the HVDC system. Without loss of generality, it is considered here that the ACCB1 is closed. As the current cannot be controlled in the blocked state, the PIR1 module is inserted at this moment (Figure 4.16c).

3. After the closure of the ACCB1, the current enters the system from the feeding AC grid and the uncontrolled charging phase begins. At the end of the uncontrolled charging, the voltage in MMC1 is equal to the peak value of the phase-to-phase transformer's secondary AC voltage $U^{ac}_{RMS}$ (commonly around 320 kV) and the MMC controller is now capable of

reproducing the voltage on its AC side (Figure 4.16a). Taking 640 kV as the base value of the per unit system for the MMC's and cables rated voltage, at 0.7 p.u. the voltage of the source converter enters in a controllable (but degraded) region of operation and the PIR1 module is bypassed in consequence. In opposition, the voltage in the arms of the remote converter corresponds to half this level as all its SMs in each phase are in series, given that the MMC2 is disconnected from its AC side (Figure 4.16d). The DC voltage in the cables follows an evolution similar to the one of the source converter.

4. The controller of MMC1 is then deblocked and the voltage (FB1.1, FB2.1) and internal energy control (FB4.1) is activated (Figure 4.16a). The first controlled charging begins. The arms of the source converter are now charged from the adjacent AC grid by means of the converter controller, which also charges the voltage in the DC cables. The controller of the remote converter remains blocked during this stage (Figure 4.16a). At the end of the first controlled charging, the voltage in the arms of the source MMC and the cables is equal to the rated voltage 1 p.u., which corresponds to the nominal RO. The voltage in the arms of the remote converter is equal to half this value (Figure 4.16a). The controller of the remote MMC is then deblocked[1] and the internal energy control (FB4.1) is activated. The active power is regulated to be zero by means of FB1.1.

5. The second controlled charging begins. The arms of the remote converter are charged from the DC cables by means of the converter controller, while the source converter compensates for the voltage drop in the cables by injecting current from the connected AC grid via FB2.1. At the end of the second controlled charging, the voltage in the arms of the source and remote MMCs is equal to the rated voltage of 640 kV (Figure 4.16a), and the cables are charged to ±320 kV.

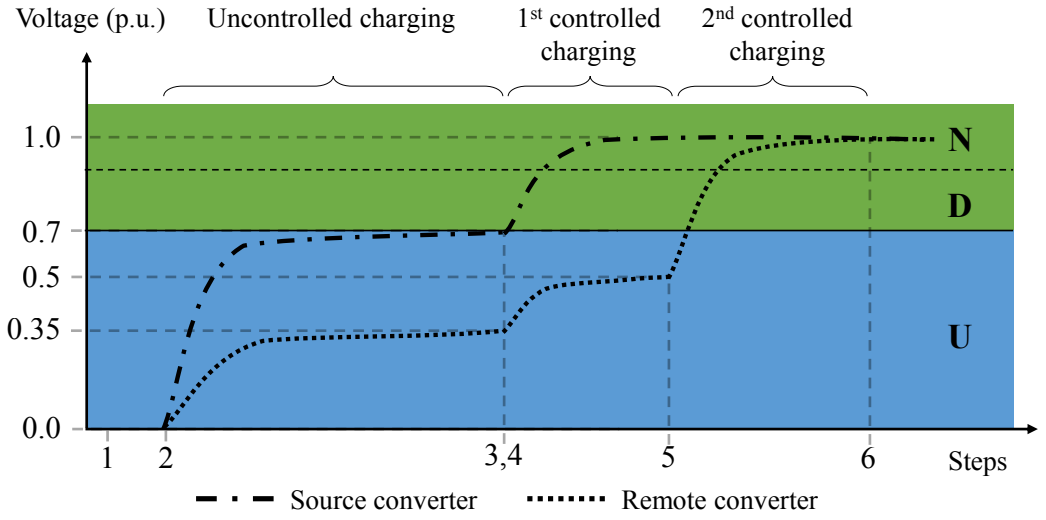
6. Finally, the ACCB2 of the remote station is then closed in order to connect the adjacent AC grid. The active power is still regulated to be zero at this point by MMC2's controller. The procedure is terminated and the operation of the system can start (Figure 4.16e).

Once the operational procedure has been described in collaboration with the system experts according to the framework proposed during the functional and monitoring analysis, we proceed to build the particular plant models and synthesize the corresponding supervisors. ⌟

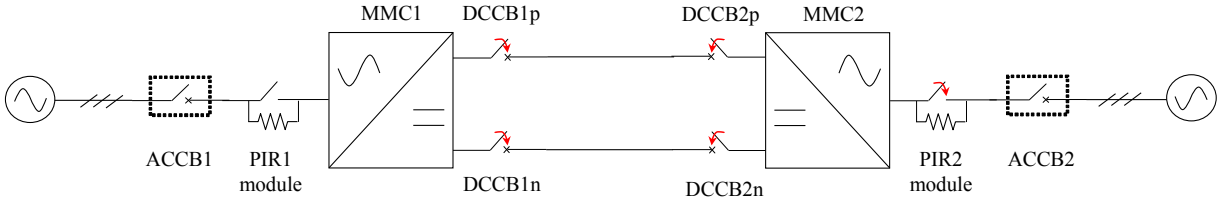
### 4.3.2 Controlled plant of the station

Given that an HVDC grid comprises a set of geographically distributed stations, each one comprising in turn a set of different components, the global plant of the entire grid is represented by a set $\mathcal{G}$ of automata, which includes each component plant $G^{C_i,P_j}$ of the system ($i \in \mathbb{N}$) and the automata modeling the physical constraints due to the concurrent functioning of several components.

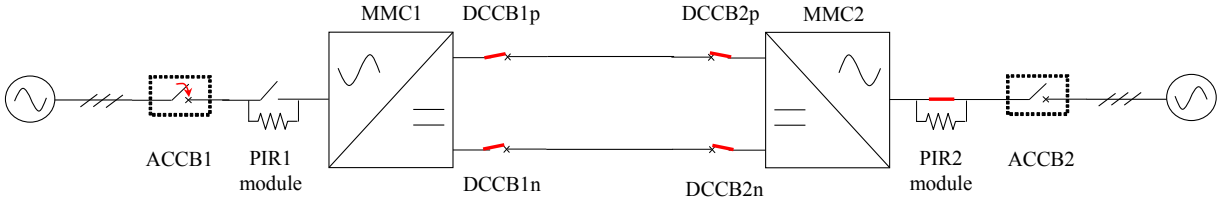
[1] Because the remote converter is not connected to its AC grid, the controllability limitations explained in Section 2.3.2 do not apply, given that no AC sinewave is generated.



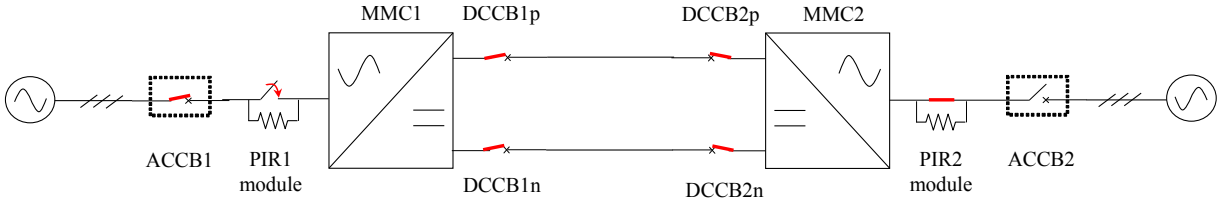

(a) MMC's voltage behavior



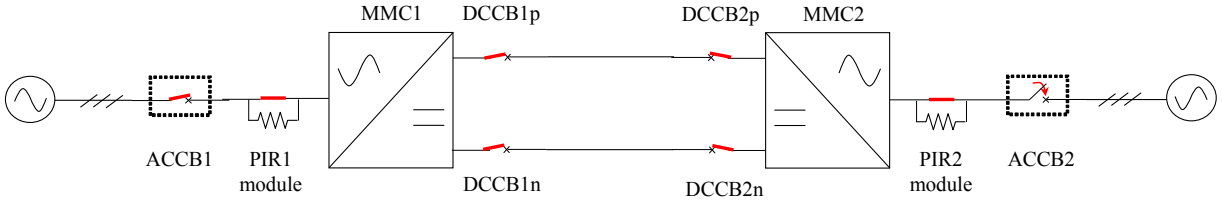

(b) Step 1




(c) Step 2




(d) Step 3




(e) Step 6

Figure 4.16 – Start-up procedure

Since the alphabet of the components in a grid is exclusive to each component, there does not exist shared events between them, that is, $\Omega = \bigcap_{i=1}^{n} \Sigma^{C_i} = \emptyset$. Nevertheless, because of the geographical proximity, common electromagnetic phenomena appear between the components, notably between those in the same station. In consequence, regarding the design of the supervisory control, because there do not exist common events between the components of the system, it is advantageous to compose first the plants in each station so that new local events can be abstracted from the resulting automaton (cf. Section 3.4.2). The abstracted models can then be used to obtain the controlled behavior of the grid. This approach based on compositional synthesis allows obtaining models where only the pertinent behaviors are considered at each level of detail. In this PhD thesis, all the stations of the system are considered to be identical in their configuration (number and type of components), therefore only the controlled behavior of one station is considered without loss of generality.

### Plant model construction

First, it is necessary to restrict the language $L(G^{C_i})$ generated by the generic component type model $G^{C_i}$ to the language $L(G^{C_i,P_j})$ generated by the particular model $G^{C_i,P_j}$ of a component in an operational procedure $P_j$ $(j = 1,\ldots,n)$. Hence, the generic automaton of a component type is first instantiated as many times as there are components of that type in the station.

Then, the behaviors of the component that do not appear in the studied procedure are not considered, i.e. the events that are not necessary for the design are removed via the projection operation (cf. Definition A.4) and the resulting automaton is trimmed in order to remove the blocking states (see Definition B.3). Because the alphabet of each particular component is component-specific, the events in the generic automaton are renamed so as to make the name of the particular component visible. An index is also added to the events if several components of the same type are located on the same side of the station. Furthermore, the initial and marked states of the component $C_i$ need to be selected by the designer according to the initial and final states of $P_j$. The resulting particular model $G^{C_i,P_j}$ is defined as follows:

***Definition 4.3 (Particular model of a component)***
*The behavior of a particular component $C_i$ in a procedure $P_j$ is modeled by an automaton $G^{C_i,P_j} = (Q^{C_i,P_j}, \Sigma^{C_i,P_j}, \delta^{C_i,P_j}, q_0^{C_i,P_j}, Q_m^{C_i,P_j})$, where:*

- *$Q^{C_i,P_j}$ is the set of states, such that $Q^{C_i,P_j} = \{q \in Q^{C_i} | (\exists s \in \Sigma^{C_i,P_j,*})[\delta(q_0,s) = q \wedge \delta(q,s) \in Q_m^{C_i}]\}$;*
- *$\Sigma^{C_i,P_j}$ is the set of events of component $C_i$ in $P_j$, such that $P_{C_i,P_j}: \Sigma^{C_i} \rightarrow \Sigma^{C_i,P_j}$ is the projection of alphabet $\Sigma^{C_i}$ onto $\Sigma^{C_i,P_j}$, with $\Sigma^{C_i,P_j} \subseteq \Sigma^{C_i}$. Once the projection is realized, the events in $\Sigma^{C_i,P_j}$ are renamed in order to indicate the component's particular name, and the component's index if necessary;*
- *$\delta^{C_i,P_j} = \delta|_{Q^{C_i,P_j} \times \Sigma^{C_i,P_j} \rightarrow Q^{C_i,P_j}}$ is the transition function;*

- $q_0^{C_i,P_j}$ *is the initial state of the component* $C_i$ *in* $P_j$*, such that* $q_0^{C_i,P_j} \in Q^{C_i,P_j}$*;*
- $Q_m^{C_i,P_j}$ *is the set of marked states of the component* $C_i$ *in* $P_j$*, such that* $Q_m^{C_i,P_j} \subseteq Q^{C_i,P_j}$. ♦

⌜ *Example*

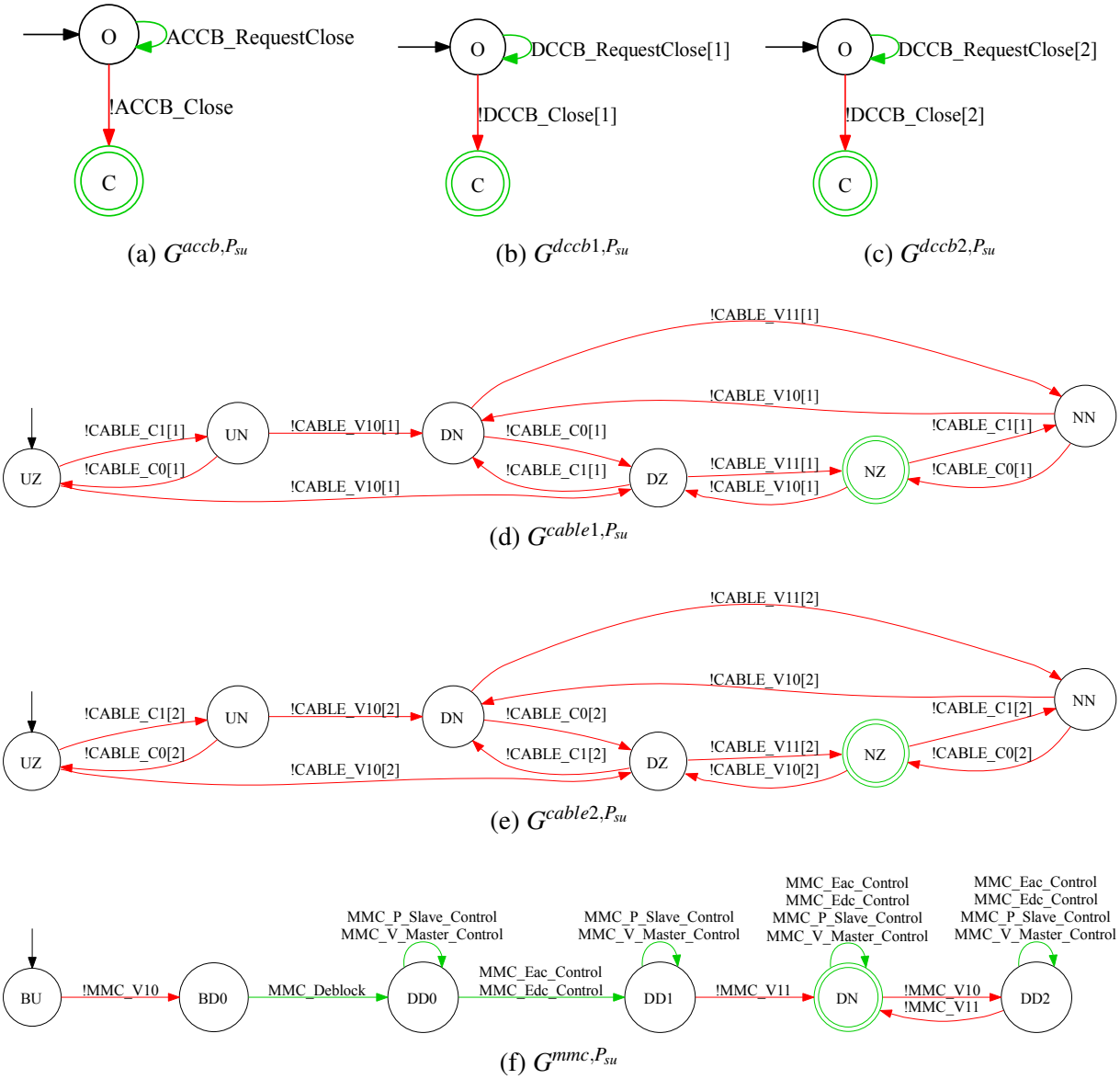
During the start-up of a point-to-point link, denoted by $P_{su}$, the component plants correspond to the automata of the different components of a station (see Figure 4.15): one ACCB, two DCCBs, two DC links and one MMC. The PIR module is commanded through the signals sent to the ACCB and so it is not necessary to model it. The generic automata (cf. figures 4.10, 4.13, 4.14) are then instantiated as many times as necessary for each type of component, that is, $G^{cb}$ is instantiated three times (one ACCB and two DCCBs), $G^{cable}$ is instantiated two times and $G^{mmc}$ is instantiated one time.

Considering the procedure presented in Section 4.3.1 and illustrated in Figure 4.16, the initial state of the MMCs and the cables of the system corresponds to the state where these components are not charged, the current in the system is zero and both the AC and HVDC grids are disconnected from the stations (O). The marked state, on the other hand, corresponds to the state where the AC and DC sides of the station are connected to the adjacent grids, the MMCs and the cables are energized but no current circulates through the grid. Thus, the initial and marked states are respectively selected for the particular automata to be: BU and DN for the MMC; UZ and NZ for both cables; O and C for the three circuit breakers.

Furthermore, the station components do not reproduce the entirety of the behaviors modeled by their generic automaton during the start-up procedure. For instance, the circuit breakers of the station are open at the beginning of the procedure and after their closure they will not be opened again. Also, if no fault occurs during the start-up (which is the case in our example), no overvoltage or overcurrent appears in the different components. Finally, only the voltage, power and energy are regulated by the MMC controller at this stage and the converter and cables are obviously not discharged during the start-up. In consequence, the events that are removed from the generic automata after the projection operation are the following: {MMC_V0, MMC_V2, $\Sigma_{config} \setminus$ {MMC_P_Slave_Control, MMC_Vdc_Master_Control, MMC_Eac_Control, MMC_Edc_Control}} for the MMC, {CB_Open, CB_RequestOpen} for the circuit breakers and {CABLE_V0, CABLE_V2, CABLE_C2} for the cables.

The remaining events are then renamed so as to avoid repetition and to differentiate the events of each individual component. As an example, the three CBs are differentiated by the label of their events, which indicates if the component is an ACCB or a DCCB, and their index, which indicates if the DCCB is located at the positive pole (index 1) or at the negative pole (index 2). To conclude, the resulting models $G^{accb,P_{su}}$, $G^{dccb1,P_{su}}$, $G^{dccb2,P_{su}}$, $G^{cable1,P_{su}}$, $G^{cable2,P_{su}}$ and $G^{mmc,P_{su}}$, shown in Figure 4.17, correspond to the particular behavior of these components during start-up. ⌟

The operation of the station components during a procedure, however, needs to respect a set of elementary physical constraints that need to be explicitly modeled. These physical constraints




(a) $G^{accb,P_{su}}$ (b) $G^{dccb1,P_{su}}$ (c) $G^{dccb2,P_{su}}$

(d) $G^{cable1,P_{su}}$

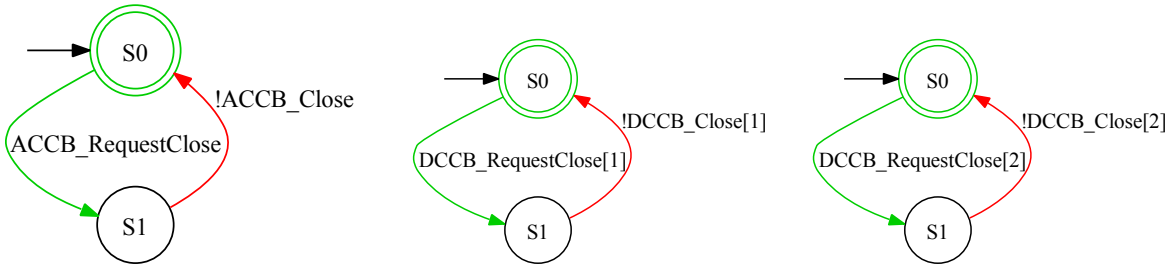
(e) $G^{cable2,P_{su}}$

(f) $G^{mmc,P_{su}}$

Figure 4.17 – Components in a station for the start-up of a point-to-point link

can be related either to the reduction of a component's behavior in $P_j$, or to the constraints imposed by the simultaneous operation of several components within the station. Different types of constraints are distinguished by means of the control flow decomposition presented in [Fen06]. Therefore, if an elementary constraint automaton $G^{C_i}_{pc_k}$ $(k = 1,\ldots,n)$ shares events with only one component model $G^{C_i,P_j}$, it represents a constraint on the said component's behavior during $P_j$. Otherwise, it represents a constraint on the concurrent behavior of several components and is denoted by $G^{st}_{pc_k}$. The uncontrolled plant of the station in $P_j$ is then obtained via the parallel composition of the different plant automata, such that $G^{st,P_j} = (\| G^{C_i,P_j}) \| (\| G^{C_i}_{pc_k}) \| (\| G^{st}_{pc_k})$, with $\Sigma^{st,P_j} = \bigcup_{i=1}^{n} \Sigma^{C_i,P_j}$ and $Q^{st,P_j} = (\bigtimes_{i=1}^{n} Q^{C_i,P_j}) \times (\bigtimes_{k=1}^{n} Q^{C_i}_{pc_k}) \times (\bigtimes_{k=1}^{n} Q^{st}_{pc_k})$.

At this stage, the synthesis triple is defined to be $(\mathcal{G};\emptyset;\mathrm{id})$, such that $\mathcal{G}$ is the set including the automata $G^{st,P_j}$ of each station.




(a) $G^{accb}_{pc1}$ (b) $G^{dccb1}_{pc1}$ (c) $G^{dccb2}_{pc1}$

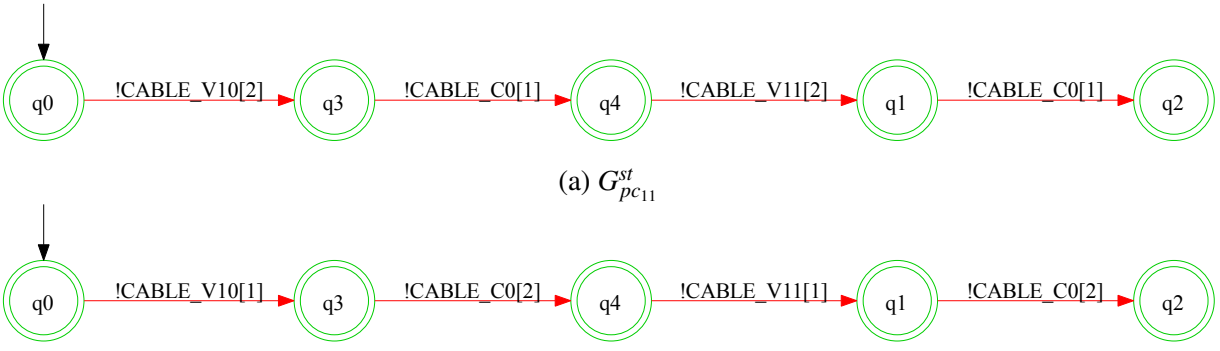
Figure 4.18 – Example of component physical constraints

⌜ *Example*

The physical constraints specific to each component in the station during the start-up need to be modeled. For instance, because we make the assumption that no short-circuit fault appears during the start-up, the conditions that allow the opening/closing of the CBs are respected at all times and so an external closing request results invariably in the closing of the circuit breakers, as shown in Figure 4.18.

Then, some of the plant automata modeling the physical constraints due to the concurrent operation of the station components are presented in Figure 4.19. For instance, automata $G^{st}_{pc_{11}}$ and $G^{st}_{pc_{12}}$ model the fact the DC current is zero once the corresponding voltage level is reached in both cables, i.e. a steady state is reached at the end of each charging phase (cf. Figure 4.16a).

The uncontrolled plant of the station $G^{st,P_{su}}$ is then obtained by the following parallel composition $G^{st,P_{su}} = (\| G^{st}_{pc_k}) \| (\| G^{C_i}_{pc_k}) \| (G^{accb,P_{su}} \| G^{dccb1,P_{su}} \| G^{dccb2,P_{su}} \| G^{cable1,P_{su}} \| G^{cable2,P_{su}} \| G^{mmc,P_{su}})$. ⌟




(a) $G^{st}_{pc_{11}}$

(b) $G^{st}_{pc_{12}}$

Figure 4.19 – Example of station level physical constraints

### Specification model construction

The model $G^{st,P_j}$ obtained in the previous step preserves synthesis equivalence but needs to be restricted in order to respect the specifications imposed by the operational procedure to the components of the system (and in consequence, to the station). It is thus advantageous to construct the model of the controlled station via monolithic synthesis at this stage, as a reduced


(a) $E_{11}^{st,P_{su}}$ (b) $E_{12}^{st,P_{su}}$

Figure 4.20 – Example of station level specifications

model is easier to be manipulated and that synthesis equivalence is preserved [Moh14].

Hence, in addition to the plant model, it is necessary to have a model of the specifications to perform a synthesis and obtain the controlled plant of the station. The SCT allows to build this model of the specifications by the parallel composition of the models representing each individual specification. Thus, if we call $E^{st,P_j}$ the model of the specifications to be respected in $P_j$, the models $E_l^{st,P_j}$ (with $l \in \mathbb{N}$) represent the elementary specifications.

⌜ *Example*

In our example, the specifications related to the behavior of the station during the start-up concern the deblocking of the automaton and the closing of the AC and DCCBs. As an example, the automata in Figure 4.20 model the request to connect the station to the adjacent AC grid, which can happen only once it has been connected to the HVDC link. Otherwise, the remote station and the DC cables would not be energized. ⌟

### Controlled plant synthesis

Then, once the model of the uncontrolled station $G^{st,P_j}$ and the specifications model $E^{st,P_j}$ are obtained, we are able to build the controlled plant $H^{st,P_j}$ of the station during the operational procedure $P_j$, which is obtained according to the following definition[2]:

***Definition 4.4 (Controlled plant of the station $H^{st,P_j}$)***
$H^{st,P_j} = (Y^{st,P_j}, \Sigma^{st,P_j}, \tau^{st,P_j}, y_0^{st,P_j}, Y_m^{st,P_j})$ *such that* $L_m(H^{st,P_j}) = [L_m(G^{st,P_j} \parallel E^{st,P_j})]^{\uparrow c}$ ♦

At this stage, which corresponds to the end of Step 2 in Figure 4.2, the synthesis triple is defined as $(\mathcal{G};\mathcal{S};\mathrm{id})$. The set $\mathcal{S}$ is formed by the controlled plants $H_i^{st,P_j}$ of each station ($i \in \{1,\ldots,n\}$, $n$ being the number of stations composing the grid), which can be regarded as supervisors that coordinate the components within the stations. However, as it is still necessary to restrict the concurrent operation of the different stations, the automata $H_i^{st,P_j}$ are considered as uncontrolled plants at the grid level. Thus, the set $\mathcal{G}$ is equally composed by the automata $H_i^{st,P_j}$ of each station.

[2] As a reminder, the $\uparrow c$ operator calculates the supremal controllable sublanguage, which is the largest sublanguage of the plant that meets the specifications (see Section 3.3.3).

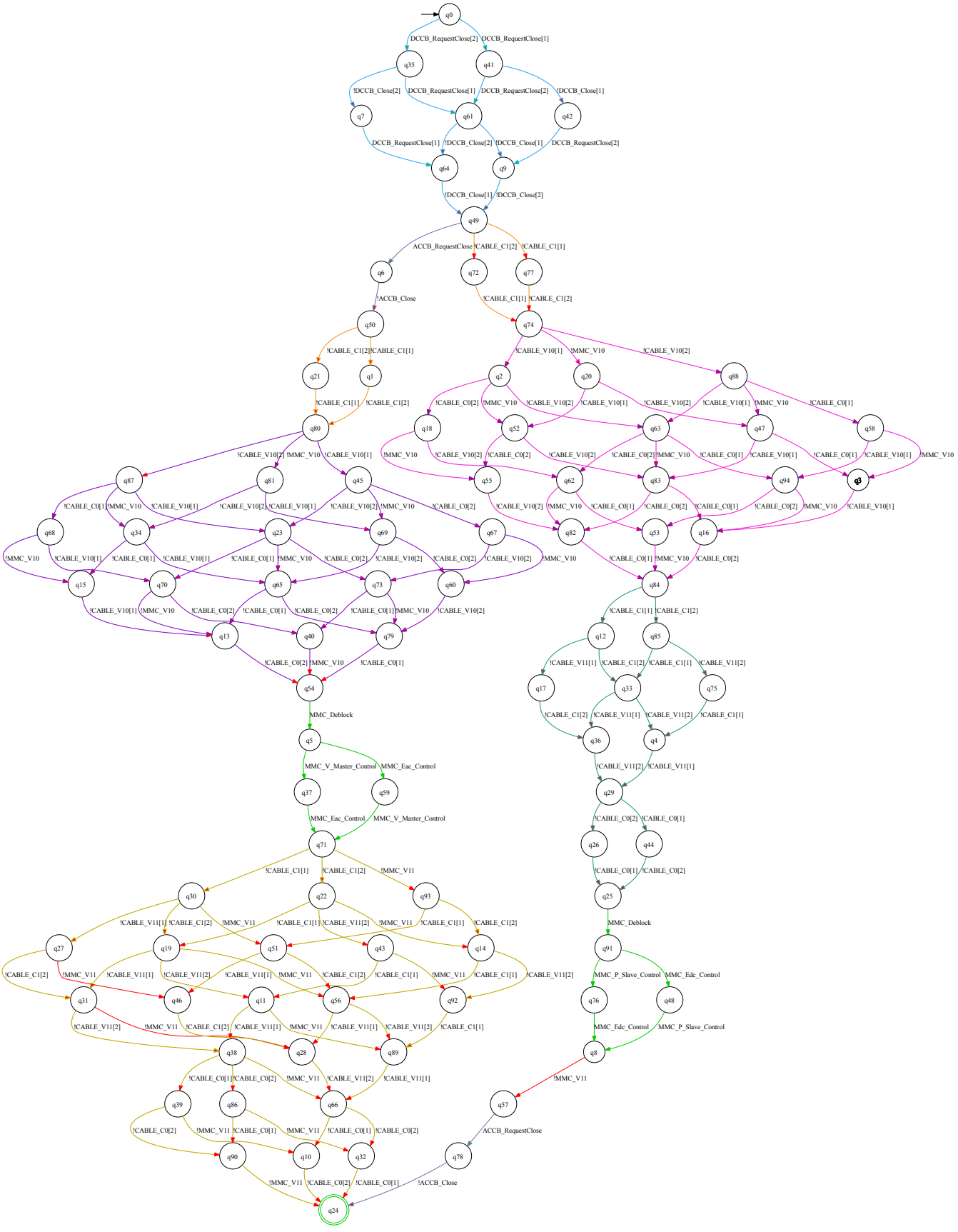



Figure 4.21 – Controlled plant of the station $H^{st,P_{su}}$ during the start-up

⌜ *Example*

The controlled plant of the station $H^{st,P_{su}}$, illustrated in Figure 4.21, is then obtained by synthesis of the uncontrolled plant of the station during the start-up and the specifications during the procedure. ⌟

### Abstraction of the controlled plant

The automata resulting from the synthesis at each station might be of a large size, which makes it difficult to manipulate and combine them at the grid level. Therefore, we proceed to abstract these models in order to remove those events that are irrelevant for the synthesis result at the grid level. Despite the fact that $\bigcap_{i=1}^{n} \Sigma^{C_i,P_j} = \emptyset$, it is possible for the designer to abstract from $G^{st,P_j}$ *a posteriori* those sequences of events that represent physical phenomena, common to several components. For this purpose, it is necessary to identify those sublanguages within the automaton that start from a unique state $q_0^{abs}$ and reach a unique state $q_n^{abs}$, such that a unique transition for each event in the subalphabet $\Sigma^{abs}$ exists for all the strings included in the language, in no particular order:

***Definition 4.5 (Abstractable language)***
*Given the set of states $Q^{st,P_j}$ of the uncontrolled plant $H^{st,P_j}$, a subset of states $Q^{abs}$ such that $Q^{abs} = \{q_0^{abs}, q_z^{abs}, \ldots, q_n^{abs}\}$ is abstractable if the following conditions are respected:*

- *there exists a unique state $q_0^{abs} \in Q^{abs}$ that is reached from the complement set $Q^{abs,\complement} \subseteq Q^{st,P_j}$;*
- *there exists a unique state $q_n^{abs} \in Q^{abs}$ from which the complement set $Q^{abs,\complement} \subseteq Q^{st,P_j}$ is reached.*

*We are then able to define a sub-automaton $G^{abs} = (Q^{abs}, \Sigma^{abs}, \delta^{abs}, q_0^{abs}, Q_m^{abs})$ such that $Q_m^{abs} = \{q_n^{abs}\}$. The language $L_m(G^{abs})$ generated by this sub-automaton can be abstracted if:*

- *the language marked by $G^{abs}$ is prefix-closed, i.e. $L_m(G^{abs}) = \overline{L(G^{abs})}$;*
- *all the events in $\Sigma^{abs}$ are included once in any string of $L_m(G^{abs})$, i.e. $\{\forall \sigma \in s | s \in L_m(G^{abs})[\exists! \delta(q_z^{abs}, \sigma)]\}$, with $z = \{0, \ldots, n\}$;*

*If the above conditions are respected, the sub-automaton $G^{abs}$ can then be abstracted and replaced by a sub-automaton $G^{abs\prime} \subseteq H^{st,P_j}$. These automaton is defined such that $Q^{abs\prime} = \{q_0^{abs}, q_n^{abs}\}$, $Q_m^{abs\prime} = \{q_n^{abs}\}$, $q_0^{abs\prime} = q_0^{abs}$ and $\delta^{abs\prime}(q_0^{abs}, \sigma^{abs\prime}) = q_n^{abs}$, where $\Sigma^{abs\prime} = \{\sigma^{abs\prime}\}$ is an event that indicates the end of the abstractable strings, such that $\Sigma^{abs\prime} \cap \Sigma^{st,P_j} = \emptyset$.* ♦

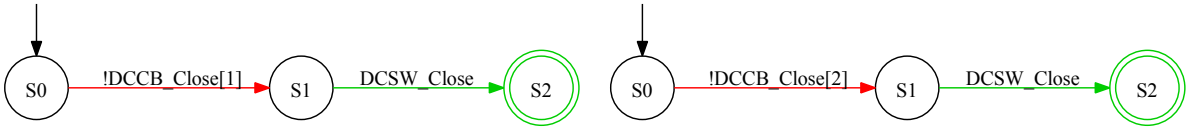
The importance of such languages relies on the occurrence of all the events comprised in them, and not so much on their order of appearance. In consequence, each of these languages can be abstracted by introducing an "end of string" event $\sigma^{abs\prime}$ that indicates the reaching of $q_n^{abs}$, such that $\delta(q_n^{abs}, \sigma^{abs\prime}) \in Q^{st,P_j}$. These events serve only design purposes, as they have no

physical relation with any of the station components.

The controllability of $\sigma^{abst}$ is to be defined by the designer according to the controllability of the events in the string: if all the events in $G^{abs}$ are uncontrollable, then $\sigma^{abst}$ is equally uncontrollable; on the other hand, if all the events in $G^{abs}$ are controllable, $\sigma^{abst}$ is defined to be controllable. Finally, if some of the events contained in an abstractable language are controllable while the others are uncontrollable, the controllability of $\sigma^{abst}$ is to be defined by the designer according to the physical meaning of the string. For instance, an uncontrollable event could be dependent upon the occurrence of a previous controllable event. As a result, the complete language could be considered as controllable and the abstraction event defined as so. The relations between the events in $\Sigma^{abst}$ and the component-specific events are then modeled in the form of elementary automata, denoted by $G_k^{abst}$. Also, it might be necessary to modify some of the previous physical constraints and specification automata. The controlled plant of the station $H^{st,P_j}$ is then synthesized again, such that $L_m(H^{st,P_j}) = [L_m(G^{st,P_j} \parallel E^{st,P_j})]^{\uparrow c}$, with $G^{st,P_j} = (\parallel G^{C_i,P_j}) \parallel (\parallel G_{pc_k}^{C_i}) \parallel (\parallel G_{pc_k}^{st}) \parallel (\parallel G_k^{abst})$. This model thus integrates the alphabet of $H^{st,P_j}$ and $\Sigma^{abst}$.

Then, because the abstraction events introduced during the synthesis of the controlled plant of the station $H^{st,P_j}$ indicate the end of a sequence of events, independently of the order of occurrence, all the events included in the abstractable language can be projected out of $H^{st,P_j}$ while preserving the synthesis equivalence. Indeed, these events can now be considered to be local to each station and thus only the abstraction events are pertinent for the synthesis approach at the grid level (Step 3 of Figure 4.2). This type of abstraction does not introduce non-determinism as each abstractable language is replaced by a single event. Thus, renaming is not needed in our work. Therefore, the synthesis triple is defined to be equal to $(\mathcal{G};\mathcal{S};\mathrm{id})$, with $\mathcal{G} = \{\tilde{H}_1^{st,P_j}, \tilde{H}_2^{st,P_j}, \ldots, \tilde{H}_n^{st,P_j}\}$ and $\mathcal{S} = \{H_1^{st,P_j}, H_2^{st,P_j}, \ldots, H_n^{st,P_j}\}$.

To conclude, although there do not exist shared events between the station's models ($\Omega = \emptyset$), the rest of events in the station are needed for the supervision and coordination of the stations, contrarily to the events projected out of $H^{st,P_j}$. In consequence, the alphabet of the abstracted model $\tilde{H}^{st,P_j}$ is obtained by means of the projection $P_{\Sigma,\tilde{\Sigma}}\colon \Sigma^{st,P_j} \to \tilde{\Sigma}^{st,P_j}$. The abstracted model is then manipulated at the grid level as an uncontrolled plant automaton to be restricted by the grid supervisor.




(a) $G_{11}^{abst}$ (b) $G_{12}^{abst}$

Figure 4.22 – Example of abstraction automata

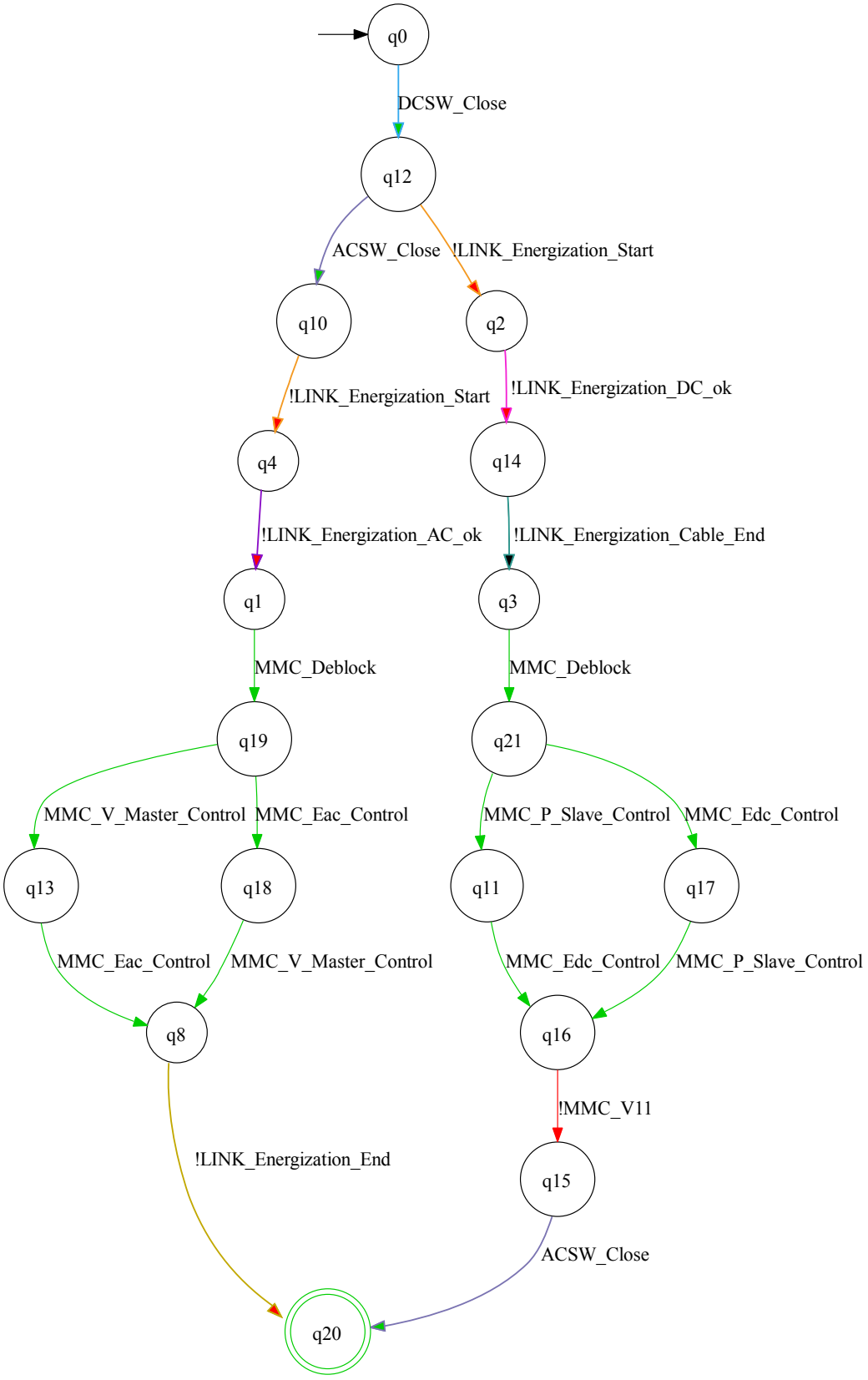



Figure 4.23 – Abstracted model $\tilde{H}^{st,P_{su}}$ of the controlled plant

*Example*

The different sets of abstractable strings are identified in Figure 4.21 and the transitions in each abstractable language are colored. For instance, the set of strings that begin at q0 and end at q8 model the fact that all the DCCBs of the station must be closed before any other action is performed. As modeled in Figure 4.18, the closing of the DCCBs is realized only once a closing request is received. Hence, the automata in Figure 4.22a and Figure 4.22b introduce the event DCSW_Close, which represents the closing of the DC switch-gear of the station when all the DCCBs are closed. In this case, because the closing of the DCCBs, which is uncontrollable, is preceded by a controllable external request, the whole sequence of events (request plus closing) is considered to be controllable; and so DCSW_Close is defined as such.

The controlled plant $H^{st,P_{su}}$ of Figure 4.21 is then recalculated, taking into account the different automata introducing the abstraction events. In our example, the set of events $\Sigma'$ that can be projected out of $H_{st}^{P_{su}}$ is $\Sigma' = \Sigma^{accb,P_{su}} \cup \Sigma^{dccb1,P_{su}} \cup \Sigma^{dccb2,P_{su}} \cup \Sigma^{cable1,P_{su}} \cup \Sigma^{cable2,P_{su}}$. The resulting abstracted automaton $\tilde{H}^{st,P_{su}}$, such that $\tilde{\Sigma}^{st,P_{su}} = \Sigma^{st,P_{su}} \setminus \Sigma'$, is given in Figure 4.23. The transitions corresponding to the new abstraction events have been highlighted with the same color used in Figure 4.21 for the abstractable languages the former have replaced.

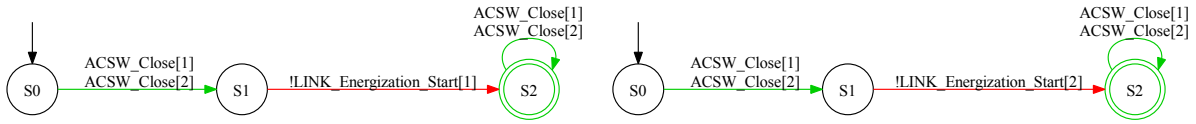
As it can be observed, the size of the model has been reduced considerably with respect to the automaton $H^{st,P_{su}}$ of Figure 4.21, while preserving the synthesis equivalence: only 22 states and 25 transitions have been retained from the previous 76 states and 114 transitions. In consequence, this step facilitates greatly the designer's work during the following stages where the controlled plant of the stations are combined between them, especially since the number of stations in the system and the complexity of their operation are expected to increase.

### 4.3.3 Controlled plant of the grid

#### Plant model construction

At this stage, because we consider the stations to share the same types and number of components, the abstracted model of the controlled plant of the stations is instantiated as many times as there are stations in the grid and its events are renamed so as to reflect by means of an index to which station they correspond (this might not be necessary in the case where the stations are not identical).

In consequence, there do not exist shared events between the reduced models of the stations, and the only method that allows to further simplify the considered automata is the monolithic synthesis. Before this, however, it is necessary to model the physical constraints imposed by the interconnection of the different stations [Rom17]. Each individual physical constraint is modeled in the form of an automaton, denoted by $G_{pc_k}^{gr}$. The uncontrolled plant $G^{gr,P_j}$ of the system at the grid level during the procedure $P_j$ is then defined such that $G^{gr,P_j} = (\|_{k\in\mathbb{N}} G_{pc_k}^{gr}) \| (\|_{i=1}^{n} \tilde{H}_i^{st,P_j})$.




(a) $G^{P_{su}}_{pc_{11},gr}$ (b) $G^{P_{su}}_{pc_{12},gr}$

Figure 4.24 – Example of grid level physical constraints

⌜ *Example*

The abstracted model of the station's controlled plant is instantiated two times, one per station, and the events are renamed by adding the station index so that the two alphabets are differentiated. An example of plant automata modeling the physical constraints due to the concurrent operation of the stations is presented in Figure 4.24. The automata $G^{gr}_{pc_{11}}$ and $G^{gr}_{pc_{12}}$ model the fact that the DC current cannot physically appear in the DC cables until the one of the two station has been connected to an adjacent AC grid. The uncontrolled plant of the grid $G^{gr,P_{su}}$ during start-up is then obtained by the following parallel composition $G^{gr,P_{su}} = \tilde{H}^{P_{su}}_{st1} \parallel \tilde{H}^{P_{su}}_{st2} \parallel (\|_{k\in\mathbb{N}}\, G^{gr}_{pc_k})$. ⌟

### Specification model construction

In addition to the plant model, it is necessary to have a model of the specifications to perform a synthesis and obtain the controlled plant of the grid. The SCT allows to build this model of the specifications by the parallel composition of the models representing each individual specification. Thus, if we call $E^{gr,P_j}$ the model of the specifications to be respected in $P_j$, the models $E^{gr,P_j}_l$ ($l \in \mathbb{N}$) represent the individual specifications.

⌜ *Example*

The specification automaton $E^{gr,P_{su}}_1$ is given in Figure 4.25 as an example. This automaton models the fact that the remote MMC's controller is deblocked only once the source MMC and the cables have been energized to its nominal voltage, for the two possible scenarios where one converter works as the source MMC and the other one as the remote converter. ⌟

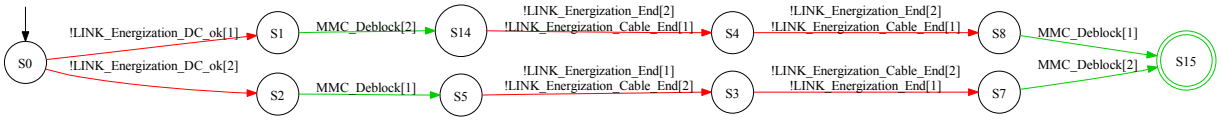



Figure 4.25 – Example of grid level specification

### Controlled plant synthesis

Then, once the plant models and specifications during the considered procedure are obtained, we are able to build the controlled plant $H^{gr,P_j}$ of the grid in $P_j$, which is obtained according to the following definition:

***Definition 4.6 (Controlled plant of the grid $H^{gr,P_j}$)***
$H^{gr,P_j} = (Y^{gr,P_j}, \Sigma^{gr,P_j}, \tau^{gr,P_j}, y_0^{gr,P_j}, Y_m^{gr,P_j})$ *such that* $L_m(H^{gr,P_j}) = [L_m(G^{gr,P_j} \parallel E^{gr,P_j})]^{\uparrow c}$ ♦

At the end of this stage, which corresponds to the Step 4 of Figure 4.2, the set of supervisors is formed by the intermediate supervisors of the station, which restrict the components behavior, and the controlled plant $H^{gr,P_j}$ representing the supremal controllable sublanguage ($H^{gr,P_j} = \sup\mathcal{C}(\mathcal{G})$) for the uncontrolled plant of the grid ($\mathcal{G} = \{G^{gr,P_j}\}$) with respect to $E^{gr,P_j}$. The synthesis triple at the end of the design is thus defined to be $(\emptyset;\mathcal{S};\mathrm{id})$, with $\mathcal{S} = \{H^{gr,P_j}, H_1^{st,P_j}, H_2^{st,P_j}\}$. The global synthesis result is then obtained by composing all the automata in $\mathcal{S}$. To conclude, although the synthesis result is equal to the controlled plant of the grid, the approach presented in this section eases the synthesis of the controlled plants of the system. Indeed, the decomposition between the components of each station has allowed to identify the abstractable strings, which reduce greatly the size of the models while preserving synthesis equivalence.

⌜ *Example*

The controlled plant of the grid $H^{gr,P_{su}}$, illustrated in Figure 4.26, is then obtained by synthesis of the uncontrolled plant of the grid and the specifications at the grid level during the start-up. ⌟

### 4.3.4 Validation of the supervisory control

During the design of the supervisory control (Steps 2 to 4 in Figure 4.2, page 65), the main work of the designer is to model the specifications and physical constraints of the system at the different synthesis levels, along with the identification of the abstractable strings so as to simplify the models as much as possible.

In the case where the controlled plant (either at the station or the grid level) in a procedure $P_j$ does not exist, this can be due to specifications $E_l^{P_j}$ being too restrictive, to a wrong modeling of the particular component dynamics in the procedure via the $G^{C_i,P_j}$ automata, or to an impossibility to control the plant. Generally, if the designer has correctly formalized the specifications and the component dynamics, the lack of a controlled plant entails a modification of the specifications in $P_j$, which will lead to a new synthesis study for the concerned operational procedure. It is also possible that the desired modifications relate to the dynamics of a component $C_i$, in which case, it is the initial step of its modeling that is to be revisited, as well as the consequent synthesis steps. Finally, if the controlled plants have the expected behavior and are validated by all the collaborators, we are certain that the specifications are respected by definition.

However, independently of the desired modifications, the synthesis of a supervisory control for a procedure requires either a designer with a deep knowledge of the system, or a close cooperation between the designer and the expert engineers. Hence the importance of the previous monitoring and functional analysis that help structure the work of each collaborator.

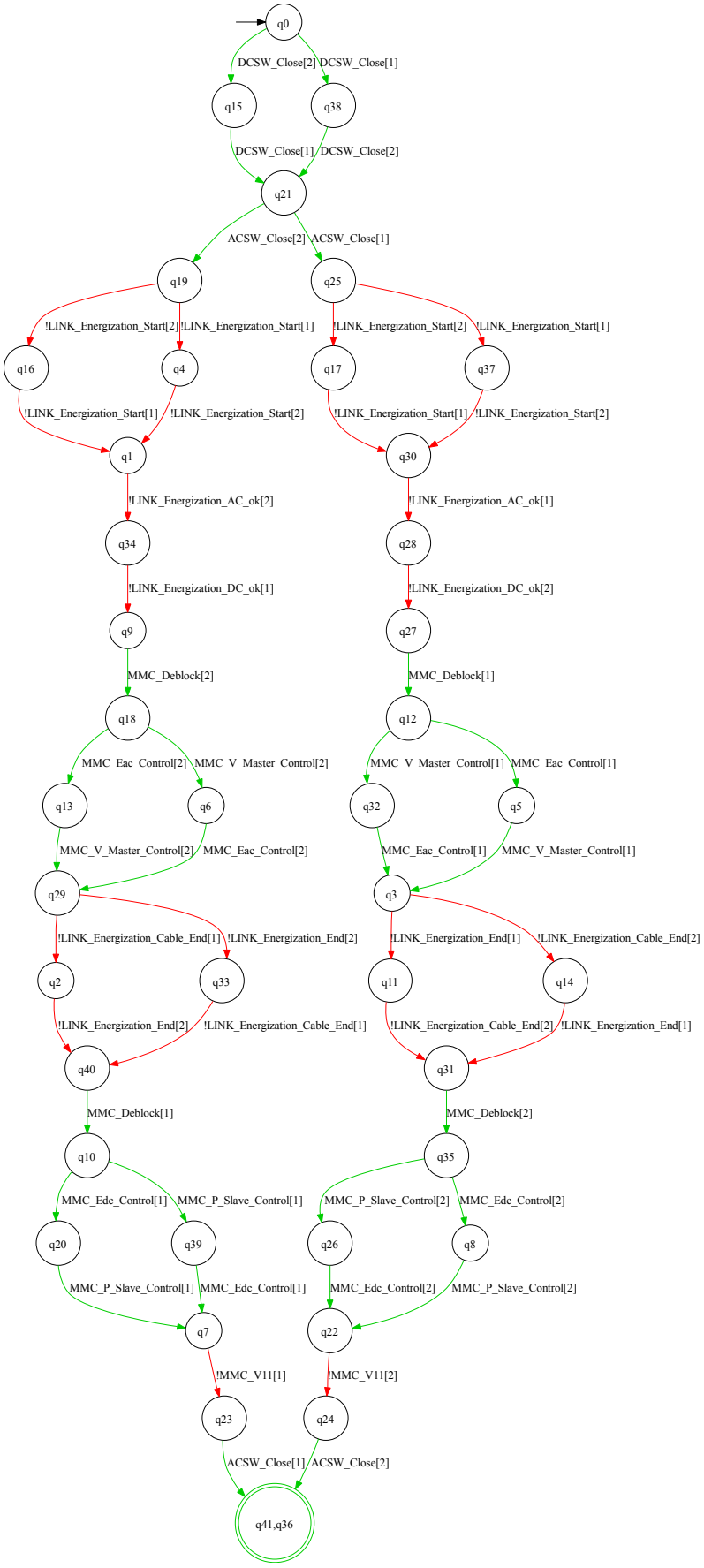



Figure 4.26 – Controlled plant of the grid $H^{gr,P_{su}}$ during start-up

Furthermore, a group of files have been written during this thesis so as to set a common framework between the system experts and the ASC designer. These files impose a predefined information structure in order to ensure the correct modeling of any given component and any given procedure.

⌜ *Example*

At this stage, the existence of the controlled plant models shows that a solution is possible so that the system can operate during the start-up while respecting the constraints of the specifications. In our example, the dynamics obtained correspond to those expected. ⌟

## 4.4 Implementation

As stated in Section 3.5, there exists a gap between the asynchronous event-driven framework proposed in the SCT and the synchronous signal-based devices (commonly, computer-based devices) in which the supervisory control is implemented. The implementation methods proposed in the literature, however, are not often adapted to the control of hybrid systems, such as HVDC grids. In consequence, an implementation method capable of integrating the designed supervisory control in interaction with the continuous-time physical system is proposed in this section. Because the dedicated supervisory control is to be distributed (horizontal, vertical and modal decomposition), the *supervised control* approach presented in [Cha95] is considered. The proposed control architecture, given in Figure 4.27, is based on a multilevel implementation, inspired from [Vie17].

As a result, the discrete-event control presented in Figure 4.1 is divided into two distinct levels, such that the practical realization of the controlled plant of the grid is implemented in a *Supervision* level that realizes supervision functions, while the practical realization of the controlled plants of the different stations are implemented in a *Logic control* level that realizes classical control functions. Finally, as stated in Section 4.1, the *Interface* level relates the continuous-time signals from the sensors and the digital signals sent to the actuators of the physical system to their discrete-event counterpart used in the ASC [Rom19]. In addition, because of the need for a highly responsive and customized supervisory control in HVDC systems, PLCs are likely not adapted for HVDC applications, as explained in Section 3.5.3. Furthermore, any control and protection program for HVDC systems is likely to be previously tested in an offline simulation software, which is usually based on common user-oriented languages. In consequence, the implementation method proposed in this PhD thesis is based on C code.

Furthermore, by reason of the multilevel structure presented in Figure 4.27, several input and output symbols will be attached to the different events of the models to be implemented. Therefore, contrary to traditional Moore machines, a transition may be triggered by several input symbols from different control levels. Similarly, it may be necessary to generate several

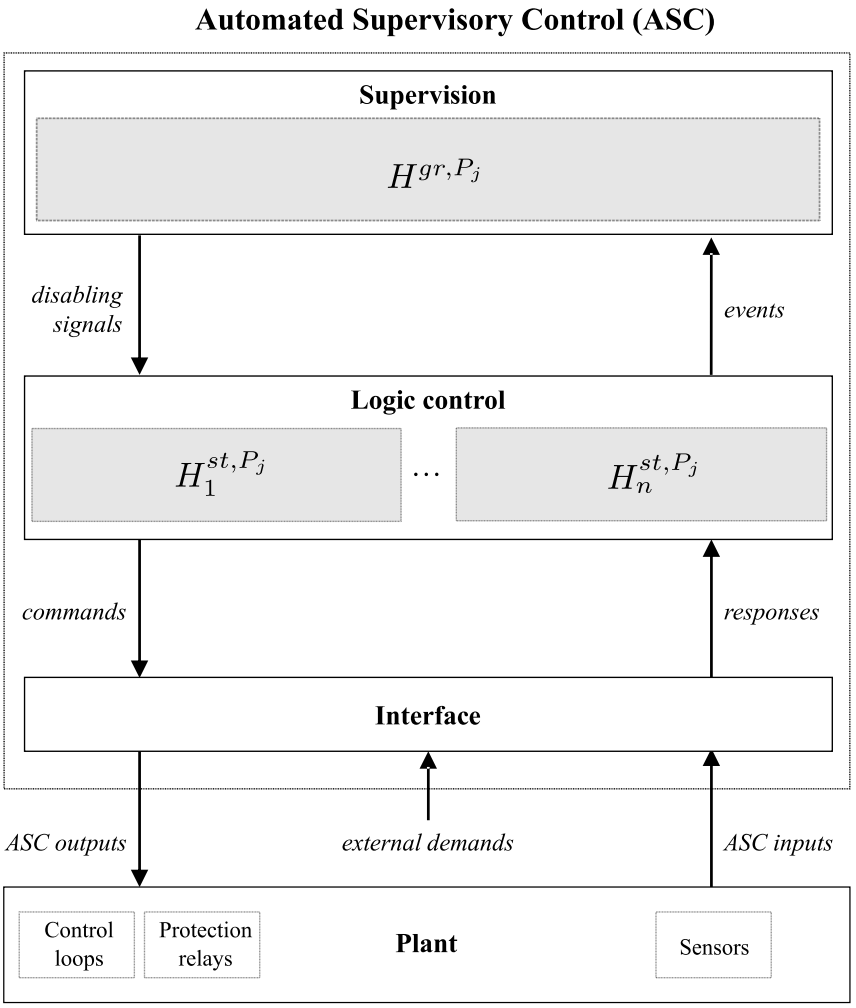



Figure 4.27 – Control structure of a compositional supervisory control for HVDC systems

output symbols within each state. In consequence, we propose in this PhD thesis the use of a particular case of Moore machine, which we call *extended Moore machine*, such that each transition of the implemented automata is associated with a set of input Boolean conditions and each state is associated to a set of output symbols. The *extended Moore machine* is obtained by adapting the extended finite state machine [Che93], which follows a Mealy machine structure, and is defined as follows:

***Definition 4.7 (Extended Moore machine)***
*An extended Moore machine $M$ is defined as a 7-tuple $M = (Q, q_0, \Sigma, \Lambda, \Phi, \delta, \omega)$ such that:*

- *$Q$ is a finite set containing all the states of $M$;*
- *$q_0$ is the initial state of $M$, such that $q_0 \in Q$;*
- *$\Sigma$ is the input alphabet associated with $M$;*
- *$\Lambda$ is the output alphabet associated with $M$;*
- *$\Phi$ is a set of input functions $\varphi_i : 2^{\Sigma} \rightarrow \{0,1\}$, associating a set of Boolean conditions to*

*the input symbols related to each transition;*

- $\delta$ *is the transition function, defined by* $\delta : Q \times \Phi \rightarrow Q$*, mapping a state and an input function to the corresponding next state;*
- $\omega$ *is the output function, defined by* $\omega : Q \rightarrow 2^{\Lambda}$*, mapping each state in* $Q$ *to the corresponding output symbols in* $\Lambda$*.* ♦

The rest of this section details the implementation-related activities 5 and 6 of Figure 4.2. The detail of the activities within Step 6 *Implementation as Moore machines* is given in Figure 4.28. The implementation method is illustrated through the manipulation of the models obtained in Section 4.3 for the start-up of a point-to-point link.

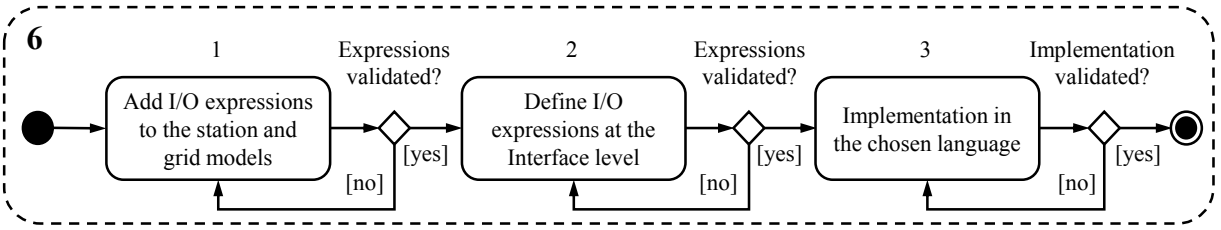



Figure 4.28 – Workflow of the implementation method

### 4.4.1 Automata conversion

In order for the different control levels to communicate between themselves, an input and output alphabet is attached to the obtained automata, which transforms them into Moore or Mealy machines (cf. Section 3.5.1). However, the transition functions of the theoretical automata may need to be previously modified so that the language recognized by the automata remains unchanged and the implemented state-machines remain deterministic when the I/O alphabets are added.

Thus, the algorithms proposed by [Vie17] for the conversion of the plant automata in view of their transformation into Moore machines are considered in this PhD thesis (cf. Appendix D). The controlled plant of the grid $H^{gr,P_j}$ can be directly transformed into an extended Moore machine $M^{gr,P_j} = (Q^{gr,P_j}, q_0^{gr,P_j}, \Sigma^{gr,P_j}, \Lambda^{gr,P_j}, \Phi^{gr,P_j}, \delta^{gr,P_j}, \omega^{gr,P_j})$, because the disabling signals generated in the states, which form the output alphabet $\Lambda^{gr,P_j}$ of the Moore machine, are independent of the transition's input alphabet $\Sigma^{gr,P_j}$. This is not the case for the plant automata $\tilde{H}_i^{st,P_j}$, and thus the two algorithms proposed in [Vie17] convert the plants $\tilde{H}_i^{st,P_j}$ into equivalent automata $\ddot{\tilde{H}}_i^{st,P_j}$ where each state is reached by a unique event. As a result, the resulting extended Moore machines $M_i^{st,P_j}$ remain deterministic when adding the I/O alphabets as the outputs generated at each state are known and differentiated for a given set of inputs (Step 5 of Figure 4.2).

*Example*

The algorithms defined in [Vie17] are applied to the model of the controlled plant of the station given in Figure 4.23. Because the plant $\tilde{H}_i^{st,P_{su}}$ does not present any self-loop transitions, the first algorithm does not apply, as $\tilde{H}_i^{st,P_{su}} = \dot{H}_i^{st,P_{su}}$. Thus, only the second algorithm is used. The converted plant $\ddot{H}_i^{st,P_{su}}$ is shown in Figure 4.29. The model of the controlled plant of the grid (Figure 4.26) does not need any transformation before the addition of an I/O alphabet.

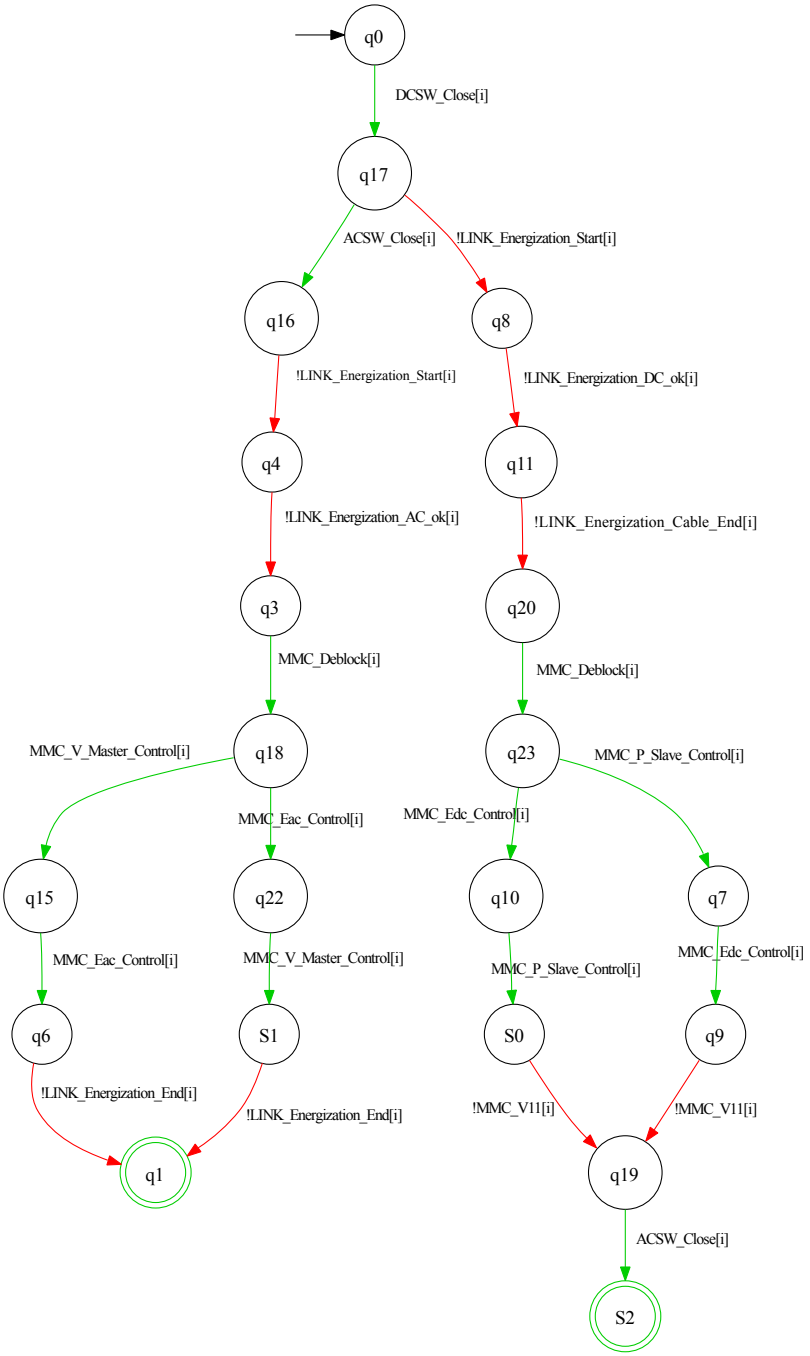



Figure 4.29 – Converted plant $\ddot{H}_i^{st,P_j}$ $(i \in \{1,2\})$

### 4.4.2 Supervision level

The Supervision level contains a realization of the supervisor automaton $H^{gr,P_j}$. It prevents the Logic Control level from generating prohibited strings of commands, so that the plant respects the behavioral specifications defined by the designer. The Supervision level takes as inputs the Boolean *events* signals, denoted by $\sigma_v$, generated by the Logic Control (Figure 4.27). Thus, the input alphabet $\Sigma_v^{gr,P_j}$ of the related Moore machine $H^{gr,P_j}$ is formed by a set of $\sigma_{v,i}$. Then, the input and output alphabets are added to the supervisor automaton $H^{gr,P_j}$ (Step 6.1 of Figure 4.28).

The input functions $\varphi_k^{gr}$ labeling each transition of $M^{gr,P_j}$ are defined by the expression $(\varphi^{gr})$:

$(\varphi^{gr})\ \ \sigma_v$

This condition is fulfilled whenever the associated event signal $\sigma_v \in \Sigma_v^{gr,P_j}$ is activated by the Logic Control. On the other hand, the output functions $\omega_k^{gr}$ generated according to $(\omega^{gr})$ in any given state of $M^{gr,P_j}$ are formed by the different *disabling signals* $\sigma_v d$ associated to the controllable events $\sigma_v \in \Sigma_{v,c,i}^{st,P_j}$ that must be disabled next by $M^{gr,P_j}$, with $\sigma_v d \in \Lambda^{gr,P_j}$:

$(\omega^{gr})\ \ \sigma_v d := x, (x \in \{0,1\})$

⌜ *Example*

In order to implement the controlled plant of the grid during the start-up $H^{gr,P_{su}}$ of Figure 4.26, a set of I/O functions is attached to the model according to $(\varphi^{gr})$, $(\omega^{gr})$. For instance, the event ACSW_Close[1] in the transition defined by the function $\delta(q6, \text{ACSW\_Close[1]}) = q16$ is replaced by the logical condition $\varphi_1^{gr} : \text{ACSW\_Close[1]}_v$, as described in $(\varphi^{gr})$. Once the transition $\delta(q6, \varphi_1^{gr}) = q16$ is realized, because none of the controllable events in $\Sigma_v^{st,P_j}$ (Figure 3.10b) is authorized by $H^{gr,P_{su}}$ in $q16$, the attached disabling signals are generated in the state $q16$ as defined by $(\omega^{gr})$. ⌟

### 4.4.3 Logic control level

The Logic Control level contains a realization of each converted model $\ddot{H}_i^{st,P_j}$. It is in charge of forcing controllable events to occur into the plant through Boolean-valued *command* signals, while taking into account the control map of the Supervision level and the physical plant evolution through the *disabling signals* and the *response* signals, respectively.

In view of the complexity of power systems operation, the supervisory control should not be isolated from the human operator at the control center so that the latter is able to choose, by means of controllable events, which of the control commands is the best to be triggered at a given state. Obviously, the operator's requests should be expressed before the system is put into operation in order to meet the fast response requirement due to the DC voltage dynamics (cf. Section 2.4). Therefore, the implemented controller should allow for the generation of some controllable events and the associated commands following external demands, as opposed

to previous works, where it is assumed that the commands are autonomously generated by the controller in a predetermined manner [Bal92; Cha95; Que02; Vie17].

Hence, the notion of *triggering* is introduced. The set of events triggered internally by the controller, denoted by $\Sigma_{\odot}$, is distinguished from the set of externally triggered events, denoted by $\Sigma_{\perp}$, such that for a given station $\Sigma_{v,i}^{st,P_j} = \Sigma_{\odot} \cup \Sigma_{\perp}$. Furthermore, the notion of triggering allows the implemented controller to be deterministic at all times since it is up to the TSO to choose which command should be generated at a given state when two or more controllable events are enabled. As uncontrollable events $\Sigma_{v,u,i}^{st,P_j}$ are generated by the physical system itself, they are externally triggered by definition ($\Sigma_{v,u,i}^{st,P_j} \subseteq \Sigma_{\odot}$). On the other hand, a subset of the controllable events $\Sigma_{v,c,i}^{st,P_j}$ to be specified by the designer are internally triggered ($\Sigma_{\perp} \subseteq \Sigma_{v,c,i}^{st,P_j}$).

Thus, for a set of events $\Sigma_v = \Sigma_{v,u,i}^{st,P_j} \cup \Sigma_{v,c,i}^{st,P_j}$, the following relation is satisfied:

$$(\Sigma_{v,u,i}^{st,P_j} \subseteq \Sigma_{\odot}) \wedge (\Sigma_{\perp} \subseteq \Sigma_{v,c,i}^{st,P_j}) \wedge (\Sigma_{\odot} \cup \Sigma_{\perp} = \Sigma_{v,i}^{st,P_j}).$$

Thus, the input conditions associated with each transition in the extended Moore machines $M_i^{st,P_j}$ ($i \in \{1,\ldots,n\}$) is defined by the controllability and triggering properties of the event labeling them (Step 1 of Figure 4.28). Whenever an event $\sigma_v \in \Sigma_{v,i}^{st,P_j}$ associated with such a transition is controllable ($\sigma_v \in \Sigma_{v,c,i}^{st,P_j}$) and triggered locally by the controller ($\sigma_v \in \Sigma_{\odot}$), the only condition for the transition to occur is that the disabling signal $\sigma_v d \in \Lambda^{gr,P_j}$ is not activated, giving as a result the input function ($\varphi_a^{st}$):

($\varphi_a^{st}$) NOT $\sigma_v d$

Also, whenever the event $\sigma_v$ is controllable ($\sigma_v \in \Sigma_{v,c,i}^{st,P_j}$) and triggered by a request external to the controller ($\sigma_v \in \Sigma_{\perp}$), the transition is crossed if the response signal $rsp\sigma_v \in \Sigma_{v,i}^{st,P_j}$, which indicates the occurrence of an external request, is activated, and that the corresponding disabling signal $\sigma_v d \in \Lambda^{gr,P_j}$ is deactivated. The Boolean expression ($\varphi_b^{st}$) is then defined:

($\varphi_b^{st}$) $rsp\sigma_v$ AND NOT $\sigma_v d$

Finally, whenever the event $\sigma_v$ that labels the transition is uncontrollable, and thus externally triggered by definition ($\sigma_v \in \Sigma_{v,u,i}^{st,P_j}$), the transition is crossed when the input response signal $rsp\sigma_v \in \Sigma_{v,i}^{st,P_j}$ indicates the occurrence of an uncontrollable event, after the continuous-time *CS inputs* fulfill certain predefined conditions. The input function ($\varphi_c^{st}$) is in this case:

($\varphi_c^{st}$) $rsp\sigma_v$

On the other hand, the output functions associated to each state of $M_i^{st,P_j}$ are defined by the controllability of the unique event reaching the state in the converted automaton, with the exception of the initial state $q_{0,i}^{st,P_j}$, which is not associated to any event. In all cases, $M_i^{st,P_j}$ generates an event signal $\sigma_v$, such that $\sigma_v \in \Lambda_{merge,st_i}^{P_j}$, so that the implemented controlled plant of the grid $M^{gr,P_j}$ is updated to the current system state. Furthermore, if the event reaching a given state is controllable, the output command signals attached to each controllable event in

$\ddot{\ddot{H}}_i^{st,P_j}$ (denoted by $cmd\sigma_v$, such that $cmd\sigma_v \in \Lambda_i^{st,P_j}$), is sent to the Interface level so that the *CS outputs* signals are modified accordingly, as defined in expression $(\omega_a^{st})$:

$(\omega_a^{st})$ $\sigma_v := 1; cmd\sigma_v := 1$

On the contrary, no command signal is generated in the case of uncontrollable events, as defined by the output function $(\omega_b^{st})$:

$(\omega_b^{st})$ $\sigma_v := 1$

⌜ *Example*

In order to implement the controlled plants of the stations $\ddot{\ddot{H}}_i^{st,P_{su}}$ during the start-up (Figure 3.10b), input and output functions are attached to the models according to expressions $(\varphi_a^{st})$ to $(\omega_b^{st})$.

For example, the event End_OM1 in the transition defined by $\delta(q9, \text{MMC\_Deblock[i]}) = q21$ is replaced by the logical condition NOT MMC_Deblock[i]$_v$d, as described in $(\varphi_a^{st})$, given that it is an internally triggered and controllable event. Consequently, the output function $\omega_{a,1}^{st}(q21, (\text{MMC\_Deblock[i]}_v, \text{cmdMMC\_Deblock[i]}_v))$ attached to q21 is defined according to $(\varphi_a^{st})$, such that MMC_Deblock[i]$_v := 1$ and cmdMMC_Deblock[i]$_v := 1$. ⌟

### 4.4.4 Interface level

In addition to the two levels defined in [Cha95], a third level that interfaces the ASC system with the continuous-time physical system is added in Figure 4.27. The Interface level is connected to the physical system through its sensors and actuators. Thus, following the activation in the Logic Control module of a command, the Interface modifies the *ASC outputs* sent to the actuators. On the other hand, when the value of the continuous-time signals (voltage, current, etc.) measured by the sensors and sent to the *ASC inputs* reaches a given region of operation defined by the corresponding thresholds (figures 4.4, 4.5), a response signal indicating the occurrence of a particular event in the physical system is activated and sent to the Logic Control level. Otherwise, in the case an external operator makes the choice of a specific control action (*external demands*), the input signal is then a Boolean signal and therefore the predefined threshold in the input expression is defined to be 0 or 1.

Each particular output expression is to be defined by the designer (Step 3 of Figure 4.28). The input and output expressions are typically related to the languages that were abstracted during the synthesis of the controlled plant of the station (cf. Section 4.3.2). For instance, the events that were projected out of the controlled plant were events included in the alphabet of the components. Hence, they represent in a detailed manner the evolution of the component variables (DC current, DC voltage, etc.). In consequence, for each of these events, the associated variable is included in the expression and a condition on the threshold indicating the occurrence of the event is defined. In this manner, we write a condition on the variable associated to each of the events included in the abstracted language. Although the sequential behavior of the

language is lost, the acknowledgment of the state of the variables suffices in our work, as we are not interested in the order of occurrence of the events in the abstracted language.

⌜ *Example*

As an illustration, the response associated to the event LINK_Energization_AC_ok can be generated from the following expression in our example:

$$\text{IF } ((v_{mmc} \geq V_{th_1}) \wedge (v_{cable_1} \geq V_{th_1}) \wedge (v_{cable_2} \geq V_{th_1}) \wedge (i_{cable_1} = 0) \wedge (i_{cable_2} = 0)) \text{ THEN}$$
$$\text{rspLINK\_Energization\_AC\_ok}_v := 1,$$

where $V_{th_1}$ corresponds to the threshold indicating the minimum voltage necessary for the MMC to be controlled.

The event LINK_Energization_AC_ok is generated whenever the MMC voltage has reached the minimum threshold necessary for the converter to be controlled (event MMC_V10, variable $v_{mmc}$), the cables voltages have reached that same level (event CABLE_V10, variable $v_{cable}$) and the current in the cables is zero (event CABLE_C0, variable $i_{cable_1}$). ⌟

### 4.4.5 Implementation as C code

The input and output expressions defined for the different control levels are conceived as generic expressions that can be interpreted and implemented in the language chosen by the designer, which needs to be compatible with the implementation requirements of HVDC systems described in Section 4.4. Consequently, the language chosen for the implementation in this PhD thesis is C code as it is supported by the dedicated simulation softwares. The state-transition structure of the automata is then implemented by means of switch-case statements. However, if the use of PLCs were considered for the implementation of the ASC, the proposed method could be implemented as Structured Text (ST) language [Int03], given that ST can be easily translated to C code and vice versa [Bas07].

If the supervisory control is implemented as C code, each control level can be coded as an individual function. The function corresponding to the Logic control level comprises a set of functions, each corresponding to a controlled plant of the station. Therefore, the input and output symbols of each level, i.e. $\sigma_v, rsp\sigma_v, cmd\sigma_v, \sigma_v d$, *ASC inputs* and *ASC outputs*, are declared as integer variables that will be manipulated by the corresponding functions. It should be noted that it suffices to rank first the input conditions of an uncontrollable event to indicate its priority over a controllable event, as the code is executed line by line during a scan cycle.

The different C code functions written for each level of the supervisory control are then coordinated via an *ASC* function that is called by an external device or software and works as the “main” of the supervisory control. In consequence, the different variables are declared and initialized within this function, except for the ASC inputs and outputs, which are the parameters of the ASC function as they are associated to the control system I/O signals exchanged with

the HVDC grid. The different functions corresponding to the control levels are then called each time the ASC function is called by the external device or software and the associated variables are given as parameters. To enable the functions to modify the value of their output variables, however, the latter must be declared as pointers.

Hence, all the variables internal to the ASC are declared as *static* variables so that they are not initialized at each execution cycle, although it is important to keep in mind that they must be reset at each execution cycle so that the avalanche effect described in [Fab98] does not appear. Therefore, the supervisory control is executed in the following order:

1. Reset disabling variables $\sigma_v d$
2. Call *supervision* function
3. Reset event variables $\sigma_v$
4. Call *logic control* function
5. Reset response variables $rsp\sigma_v$
6. Call *interface* function
7. Reset command variables $cmd\sigma_v$

To conclude, the generation of the code associated to the different control levels can be largely automated given that the information necessary to construct the generic expression and their translation to the code is provided by the automata models. The interface function, however, needs the manual intervention of the designer, as it is system-specific.

⌜ *Example*

The converted plant $\ddot{\tilde{H}}_i^{st,P_j}$ of Figure 3.10b is implemented as a C code function (named *logic control*) within the supervisory control program, as shown in Listing 4.1. In Listing 4.1, the expression of the type $(\varphi_a^{st})$ for the event CloseDCSW$_i$ labeling the transition from the state q0 to the state q14 is expressed by means of the disabling variable. According to expression $(\omega_a^{st})$, two types of integer variables are then associated to the outputs generated at q14: one for the command, as the event reaching the state is controllable; the other for the event variable. The transitions leaving the new state are then defined according to the corresponding generic expressions, e.g. expression $(\varphi_c^{st})$ for the event LINK_Energization_Starti, which is uncontrollable; $(\varphi_b^{st})$ for the event CloseACSWi, which is controllable but externally triggered, as the charging AC grid is to be determined by the human operator of the HVDC grid; and so on.

```
1  void Hst_Psu1(int* DCSW_Close1, int DCSW_Close1d, ...){
2   static int state = 0;
3   switch(state){
4    case 0:
5      if(!CloseDCSW1d){
6        state = 14;
7      }
8    break;
9
10   case 14:
```

```
    *CloseDCSW1 = 1;
    *cmdCloseDCSW1 = 1;
    if(rspLINK_Energization_Start1){
      state = 22;
    }
    if(rspCloseACSW1 && !CloseACSW1d){
      state = 11;
    }
  break;
  ...
 }
}

void logic_control(int* DCSW_Close1, int DCSW_Close1d, ...){
 Hst_Psu1(&DCSW_Close1, DCSW_Close1d, ...);
 ...
}
```

Listing 4.1 – Code snippet of the *logic control* function

```
void ASC(double m_Vmmc, double m_Vcable1, int* sDCCB11, int* sDCCB12, ...){
 //Declare integer variables
 static int DCSW_Closei, ACSW_Closei, MMC_V10i, ... = 0;
 static int DCSW_Closeid, ACSW_Closeid, MMC_Deblockid, ... = 0;
 static int cmdDCSW_Closei, cmdACSW_Closei, cmdMMC_Deblocki, ... = 0;
 static int rspACSW_Closei, rspMMC_V10i, rspMMC_V10i, ... = 0;

 //Reset disabling variables
 DCSW_Closeid, ACSW_Closeid, MMC_Deblockid, ... = 0;
 //Call supervision function
 supervision(DCSW_Closei, &DCSW_Closeid, ...);
 //Reset event variables
 DCSW_Closei, ACSW_Closei, MMC_V10i, ... = 0;
 //Call logic control function
 logic_control(DCSW_Closeid, rspMMC_V101, &cmdDCSW_Closei, ...);
 //Reset response variables
 rspACSW_Closei, rspMMC_V10i, rspMMC_V10i, ... = 0;
 //Call interface function
 interface(m_Vmmc, &rspMMC_V101, cmdDCSW_Closei, &sDCCB11, ...);
 //Reset command variables
 cmdDCSW_Closei, cmdACSW_Closei, cmdMMC_Deblocki, ... = 0;
}
```

Listing 4.2 – Code snippet of the *ASC* function

Finally, the code corresponding to the ASC function is given in Listing 4.2, where the variables m_Vmmc, m_Vcable1, sDCCB1 and sDCCB2 correspond respectively to: MMC's voltage

measurement, cable's voltage measurements and the digital signals sent to both DCCBs of the station 1. ⌟

## 4.5 Conclusion

In this chapter, following a compositional approach, a supervisory control scheme that manages the automation and coordination of an HVDC system incrementally is proposed. First, the components located in a station are modeled in their generic form, which is later instantiated for each particular component for a given procedure. In order for the obtained component models to be coherent with the physical behavior of the system, however, it is necessary to perform in advance a functional and monitoring analysis in collaboration with the system experts. Then, the controlled plant of each station is synthesized after the physical constraints of the components behavior and the specifications have been identified. If possible, the resulting automaton is simplified so as to ease its manipulation by the designer. Next, the global plant of the grid is constructed by composition of the simplified models, for a grid of any size, and the global controlled plant is obtained by synthesis. Finally, an implementation method that allows to separate the supervisory control into different levels of detail (i.e. Supervision, Logic Control and Interface) is presented. Moreover, the method is conceived through generic expression that can be later implemented in C code (and could be extended to Structured Text), thus allowing a highly-reactive and customized control for the real-time operation of HVDC systems. The proposed implementation method can be largely automated so as to reduce the time of supervisory control development. To conclude, the proposed design and implementation method has been applied to the start-up of a point-to-point link as a means to illustrate the different steps of the approach.

# 5

# Mode-switching architecture

## Contents

## 5.1 Introduction

In the previous chapter, a design and implementation method based on a vertical decomposition of the system for a given operational procedure was proposed. The large number of operating conditions of HVDC systems makes it necessary for the supervisory control to follow the mode commutations of the grid. In consequence, a systematic approach, illustrated in Figure 5.1 in the form of a UML activity diagram, for the design and implementation of a mode-switching control based on the use of the modal decomposition (cf. Section 3.4.3) is presented

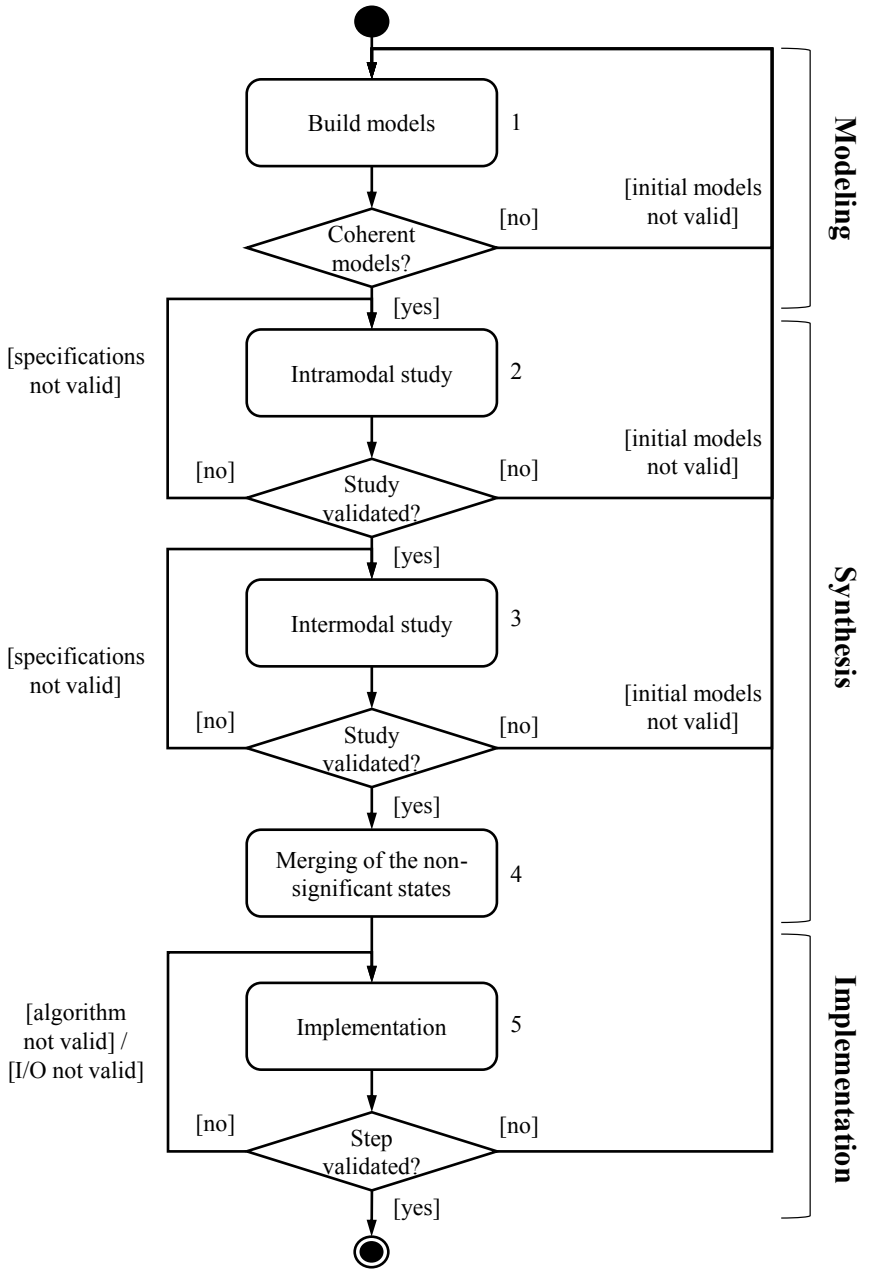



Figure 5.1 – Proposed approach for mode-switching control design

in this chapter.

Prior to the design of the ASC, it is necessary to identify the commutative behavior of the HVDC system by means of a dedicated functional and monitoring analysis, and then model the mode-switching behavior under the form of an automaton (Step 1 in Figure 5.1). Step 2 concerns the intramodal study of each mode independently of the others, through the modeling of intramodal specifications and the synthesis of dedicated supervisors. Then, because the internal controlled plants do not represent all the possible commutations with the other modes, the automata obtained during the intramodal study are extended in Step 3 by means of intermodal specifications so as to take into account the modeled commutations (Section 5.4). In the last step of the design, the obtained models are reduced and the non-significant states are merged to show the inactivity of the mode (Section 5.5). Finally, in the implementation stage (Section 5.6), a variation of the method proposed in Section 4.4 is given.

## 5.2 Modes automaton model

In order for the supervisory control to discriminate automatically the set of actions the system should undergo, a continuous monitoring of the system is necessary. Because existent solutions rely heavily on human interpretation of the general overview given by the operational states classification of Figure 2.4, new approaches need to be formulated for HVDC systems, given the fast response needed from the security control and the complexity of some coordinated actions. In consequence, the classification of the operational states of an HVDC system needs to incorporate specific information that allows to quickly monitor both the functional and security state of the grid.

### 5.2.1 Functional and monitoring analysis

In this PhD thesis, an approach that classifies the operational states of the system based on the capability of the components in the grid to transfer power and maintain the power balance is proposed. In the case of MMC-based HVDC grids, such components would be the MMCs themselves and the DC cables. The capability to transfer power of these components can be identified through the regions of operation defined in Section 4.2.1 for the MMC's voltage and the cable's voltage and current. Indeed, an MMC whose voltage is under the controllability threshold is unable to transfer the power required by the Transmission System Operator (TSO). On the other hand, if an MMC suffers overvoltage, the probability of damage increases and so its capability to transfer power is undermined. The same can be said about the cable's voltage. In the case of the current, if its value is zero, no power is transferred through the cable. Likewise, an overcurrent commonly implies the occurrence of a short-circuit fault, and in consequence, there is no power transferred, either (as the voltage in the cables collapses).

Thus, as it can be observed, the regions of operation of the different component variables determine the functional and security conditions of the grid. Then, the Operational States (OSs) of the system are defined as the set of combinations obtained via the Cartesian product between the three regions of operation (ROs) of the identified variables (MMC voltage, cable voltage and cable current). Because the number of combinations grows exponentially with the number of stations in the grid, the hypothesis is made that the same types of components in all the stations share the same region of operation (e.g. all the MMCs voltage is in RO1, all the cables current is in RO2, etc.).

However, some of the obtained combinations might never be physically reachable. Therefore, a thorough analysis should be undertaken in order to remove them from the supervisory control design. As an example, some of the physically possible operational states of any given HVDC grid are described in Table 5.1. The operational states are referred to by a number of three digits that indicates the region of operation of the three types of state variables: the first digit corresponds to $v_{mmc}$, the second digit to $v_{cable}$ and the last digit to $i_{cable}$.

Once the feasible operational states are identified, an exhaustive study needs to be realized so as to define how the operational states are reached from one another. The set of events that provoke a change of state can either correspond to control and protection actions or to disturbances external to the control system. As an illustration, some of the actions and disturbances that modify the operational states of Table 5.1 are shown in Figure 5.2. Each blue arrow in Figure 5.2 is an abstracted representation of the diverse and detailed sequences of control and protection actions (i.e. the operational procedures) that begin at a given operational state and end at a different OS, as determined by the system experts. Thus, this abstracted representation, called an Operational Mode (OM), groups under the same label all the different operational procedures that start and end at the same operational state, that is, they correspond to the same operational objective. The relationship between the service functions of the components, the operational procedures and the operational modes is illustrated in Figure 5.3.

In this manner, the following modes are identified in this PhD thesis: the start-up mode, the protection mode, the voltage restoration mode and the power ramp-up mode.

| **Operational State** | **Description** |
|---|---|
| OS000 | All the MMCs and cables are uncharged and no power is transferred |
| OS110 | All the MMCs and cables are charged to their nominal value and no power is transferred |
| OS111 | All the MMCs and cables are charged to their nominal value and power is transferred |
| OS102 | Short-circuit fault in the system (overcurrent and DC voltage collapse) |
| OS100 | Faulty cable is isolated, the MMCs are charged and the DC voltage needs to be restored |

Table 5.1 – Description of some OS of the system

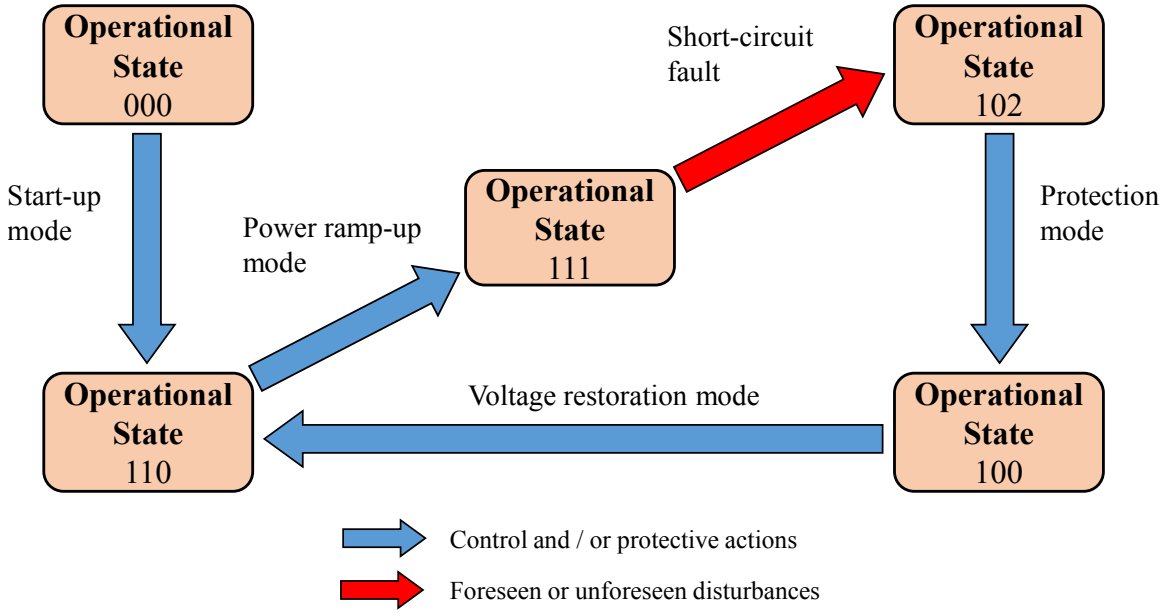



Figure 5.2 – Some OS of an HVDC system

**OM1: Start-up mode** The start-up mode includes all the procedures that take place at the beginning of a system operation or after a blackout where the whole HVDC grid has been discharged. The objective is to charge all the MMCs and DC cables of the system to their nominal voltage. For this, several strategies can be used, as described in Section 4.3.1.

**OM2: Power ramp-up mode** The power ramp mode includes all the procedures that take place once the rated voltage has been reached for all the components in the grid. Then, the power ramp-up mode involves the increase of the power in the system so that each station transfers the power determined by the dispatch controller in the control center.

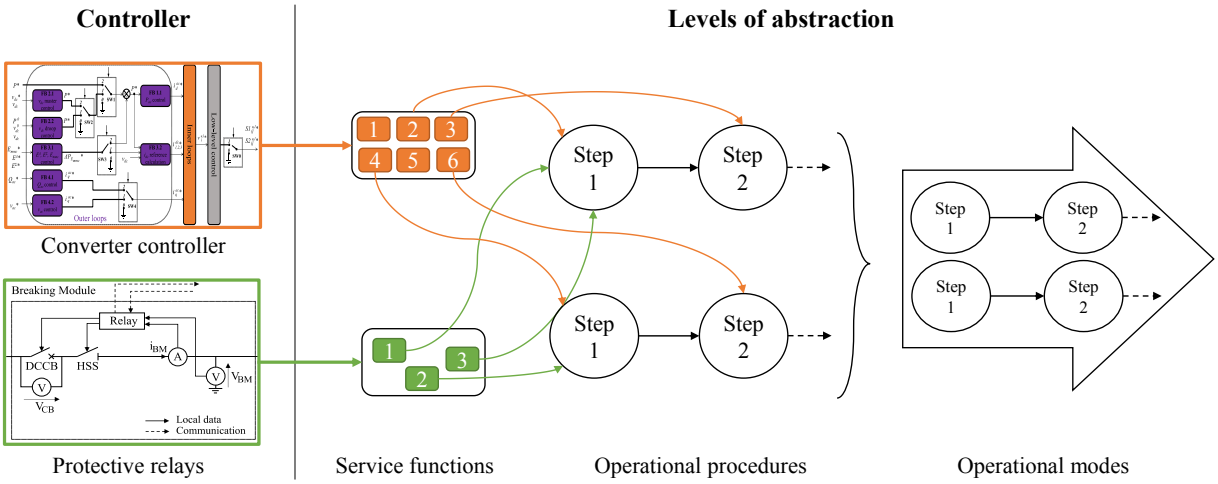



Figure 5.3 – Functional diagram of the security control system

**OM3: Protection mode** The protection mode includes the procedures that take place after a short-circuit fault appears in the system. As introduced in Section 2.3.2, two types of approaches are considered for the protection procedures: the selective approach and the non-selective approach. In [Leó16], the authors propose a selective protection procedure based on Superconducting Fault Current Limiters (SFCLs). Such components are capable of limiting the slope of a fault current to a certain thermal limit. Once this thermal limit is reached, the properties of the material composing the device change and its resistance rises sharply, thus isolating the faulty link alone. This procedure, however, is based on state-of-the-art technology which is nowadays immature and expensive. In [Lou17], the authors present a non-selective protection procedure where the fault detection algorithms in each relay detect, based on local measurements, if a short-circuit fault in the cable has occurred. Then, all the DCCBs located on the DC side of the station are tripped in order to isolate the fault from the AC grids feeding the HVDC grid and extinguish the fault current in consequence. Simultaneously, all the converter controllers are blocked in order to avoid damage in the MMC caused by an incorrect control law and the surge current. In the meantime, the relays of the system determine the location of the fault through their concurrent fault identification algorithms. The relays send then a tripping order to the DCCBs located at each end of the faulty cable. Once the short-circuit is isolated, the DCCBs at the station's output are closed again and the rest of the system can restart its operation. Currently, the actions in this mode are based on protection algorithms that act independently from the supervisory control. Because it is out of the scope of this thesis to develop a protection strategy based on DES modeling, we only consider for this mode the outputs of the protection algorithms. In this manner, the supervisory control is able to follow the internal evolution of the mode. However, these outputs are not controllable and so the supervisory control does not influence the state of the system during the protection.

**OM4: Voltage restoration mode** The voltage restoration mode includes the procedures that take place after a non-selective protection procedure, where all the healthy links in the grid have been discharged, the faulty link disconnected from the system and the MMC's controllers are blocked. Hence, in order to restart operation of the system, it is necessary to first restore the voltage in the remaining healthy links. There exist multiple variants to do this. For instance, all converter controllers can be deblocked, and similarly to the start-up strategy in [Yu13; Gao14], one MMC regulates the voltage (FB2.1) while the rest of them regulate the power to zero (FB1.1). In a variant of this procedure, the MMCs are not deblocked in a first time. Hence, the links are energized via an uncontrolled charging where the surge current is limited by the PIRs. Only at the end of this stage are the converter controllers deblocked and the voltage controlled until it reaches its rated value. A third procedure involves the deblocking of some MMC controllers while others remain disconnected from the HVDC grid at the end of the protection procedure. The disconnected MMCs can then be deblocked in order to realize stability support

of the adjacent AC grids by means of AC voltage or reactive power control (FB4.1, FB4.2).

To conclude, the functional and monitoring analysis proposed in this PhD thesis works as a two-way approach that allows to identify the operational procedures necessary for the automated operation of the HVDC system. In particular, an operational procedure can be identified from the analysis of the possible transitions between the operational states of the grid. Conversely, a transition that was rejected at first can be later considered for the grid's operation if advancements on the function blocks of the components permit to develop an adequate procedure. In consequence, the proposed method allows an adaptive design of the supervisory control.

### 5.2.2 Model construction

All the modes of the system must be defined within a set $\mathcal{M} = \{M_1, \ldots, M_j, \ldots, M_n\}$ ($j = \{1, \ldots, n\}$, $n$ being the number of modes), where $M_j$ is the name of the considered mode given by the designer in order to improve the interpretation of the models. The modes of an HVDC grid can be naturally defined as the operational modes identified in the previous analysis (cf. Figure 5.2). Furthermore, the operational states of Figure 5.2 can be considered as *static modes* from a supervision perspective since the only internal behavior that exists in this modes is the one related to the occurrence of a fault, in opposition to the OMs, which are considered as *dynamic modes* as the supervisory control has an active role in them. As stated in Section 2.3.2, pole-to-pole short-circuit faults are unlikely to occur in HVDC systems based on high-voltage cables. In addition, short-circuit faults are more likely to occur due to the aging of the cable [Des13]. Therefore, solely the pole-to-ground short-circuit faults that occur during nominal operation of the system ($M_5$) are considered in this PhD thesis. Thus, an identification number is given in Table 5.2 for each of the operational modes mentioned above in order to simplify the name of the subsequent models.

The HVDC system evolves then through the alternation of the modes in Table 5.2, such that only one $M_j (j = \{1, \ldots, 9\})$ is active at a time (cf. Section 3.4.3). The mode-switching behavior of the system, determined by the monitoring analysis described in Section 5.2.1 and represented in Figure 5.2, is modeled by the designer in the form of the modes automaton given in Figure 5.4, according to Definition 3.17. The initial state corresponds to $M_1$, where all the MMCs and cables are discharged and no current circulates through them. At first, it is necessary to activate the start-up mode ($M_2$) so as to energize the MMCs and the cables ($M_3$). Secondly, the power ramp mode ($M_4$) needs to be started in order to realize the desired power flow. Then, the nominal mode $M_5$ is reached, which corresponds to the marked state of the automaton. If an uncontrollable short-circuit fault occurs, an overcurrent appears and the voltage in the cables collapses ($M_6$). Hence, it is necessary to activate first the protection ($M_7$), then the voltage restoration ($M_9$) modes once the fault is removed ($M_8$), so as to return to a state where the cables are energized and the fault is removed from the grid ($M_3$) Finally, $M_4$ is started again in

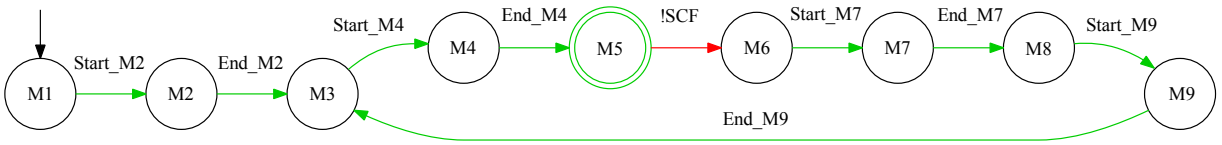



Figure 5.4 – Modes automaton $G^{\mathcal{M}}$

order to reestablish a power flow in the system and reach the marked mode $M_5$.

The set $\Sigma^{\mathcal{M}}$ of commutation events is formed by either controllable events that correspond to the start and the end of a given mode (start-up, protection...), or by uncontrollable events indicating the occurrence of a fault (e.g. a short-circuit fault). In any case, because the commutation events are associated to the reaching of a particular configuration by the whole set of components in the grid, they are not included in any of the individual alphabets of the components ($\Sigma^{\mathcal{M}} \bigcap (\bigcup_{i=1}^{n} \Sigma^{C_i}) = \emptyset$), as opposed to [Far10]. Given that no component generates a switch event, i.e. $\bigcup_{i=1}^{n} \Sigma^{C_i}_{\rightleftarrows} = \emptyset$ (see Definition 3.16), the set $\mathcal{C}^{M_j}$ of components used in a mode $M_j$ is entirely included in the set $\mathcal{C}^{M_j}_{\circlearrowleft}$ of components representing the intramodal behavior of the system in the mode $M_j$, with the set of components that generate a commutation event $\mathcal{C}^{M_j}_{\rightleftarrows} = \mathcal{C}^{M_j}_{\leftarrow} \cup \mathcal{C}^{M_j}_{\rightarrow} = \emptyset$. At this point, the particular component models (cf. Definition 4.3), denoted in the following by $G^{C_i,M_j}$, and $G^{\mathcal{M}}$ have been defined and built from the monitoring and functional analysis. Although their alphabets are disjointed, the consistency between the automata must be validated, as an error in the initial models will be carried over the following steps.

| **Name** | **Mode** |
|---|---|
| $M_1$ | OS000 |
| $M_2$ | OM1: Start-up |
| $M_3$ | OS110 |
| $M_4$ | OM2: Power ramp |
| $M_5$ | OS111 |
| $M_6$ | OS102 |
| $M_7$ | OM3: Protection |
| $M_8$ | OS100 |
| $M_9$ | OM4: Restoration |

Table 5.2 – Considered modes of the system

### 5.2.3 Validation of the consistency between models

Because the set $\mathcal{M}$ is embedded in the definition of $G^{\mathcal{M}}$, these two models are necessarily consistent with respect to each other. On the other hand, the relations between the components $G^{C_i,M_j}$, the sets $\mathcal{C}_{\circlearrowleft}^{M_j}$ of components in a mode (cf. Definition 3.16) and $G^{\mathcal{M}}$ are not formally defined. It is therefore necessary to verify that the models are is not in contradiction to each other. The following definition lists the properties to be respected to validate this consistency:

***Definition 5.1 (Consistency between models)***
*Considering the set $\mathcal{C}_{\circlearrowleft}^{M_j}$ of components in a mode $M_j$, the model $G^{C_i,M_j}$ of a component $C_i$ in $M_j$, such that $C_i \in \mathcal{C}_{\circlearrowleft}^{M_j}$, is consistent with the modes automaton $G^{\mathcal{M}}$ if the following conditions are met:*
***Condition 1**: The set of components of any mode $M_j$ of $\mathcal{M}$ without an internal behavior is empty, such that $\mathcal{C}^{M_j} = \emptyset$. In opposition, the set of components of any mode $M_j$ of $\mathcal{M}$ with an internal behavior is non-empty, such that $\mathcal{C}^{M_j} \neq \emptyset$.*
***Condition 2**: the models $G^{C_i,M_j}$ of a component $C_i$ in a mode $M_j$, such that $C_i \in \mathcal{C}_{\circlearrowleft}^{M_j}$, are consistent if the product of their initial state is equivalent to a marked state of the last mode with an internal behavior from which $M_j$ is reached. Respectively, they are consistent if the product of their marked states is equivalent to the product of the initial state of the component models in the mode with an internal behavior that is reached from $M_j$.* ♦

In the case where all these conditions are respected, the construction of the models has been well conducted and the consistency property is validated. This means that each mode with an internal behavior contains a set of components (Condition 1), and that the initial and marked states of the component models in a mode are consistent with the modes before and after the studied mode (Condition 2).

In the opposite case, the designer can identify the problem according to the conditions that were not respected and modify all or part of the previously built models. Thus, if the Condition 1 is not respected, the states of $G^{\mathcal{M}}$ are poorly defined in the monitoring and functional analysis. Finally, if the Condition 2 is violated, then the $G^{C_i,M_j}$ automata have been built incorrectly.

## 5.3 Intramodal study

The objective of the intramodal study (Step 2 in Figure 5.1) is to allow a first study of each mode independently of the others. This makes it possible to verify that the specifications to be respected are well constructed and only allow the desired internal behavior for each mode (cf. Section 3.4.3). This study is therefore conducted on a reduced behavior of the overall behavior of the system, allowing an easier interpretation of the model by the designer and thus facilitating

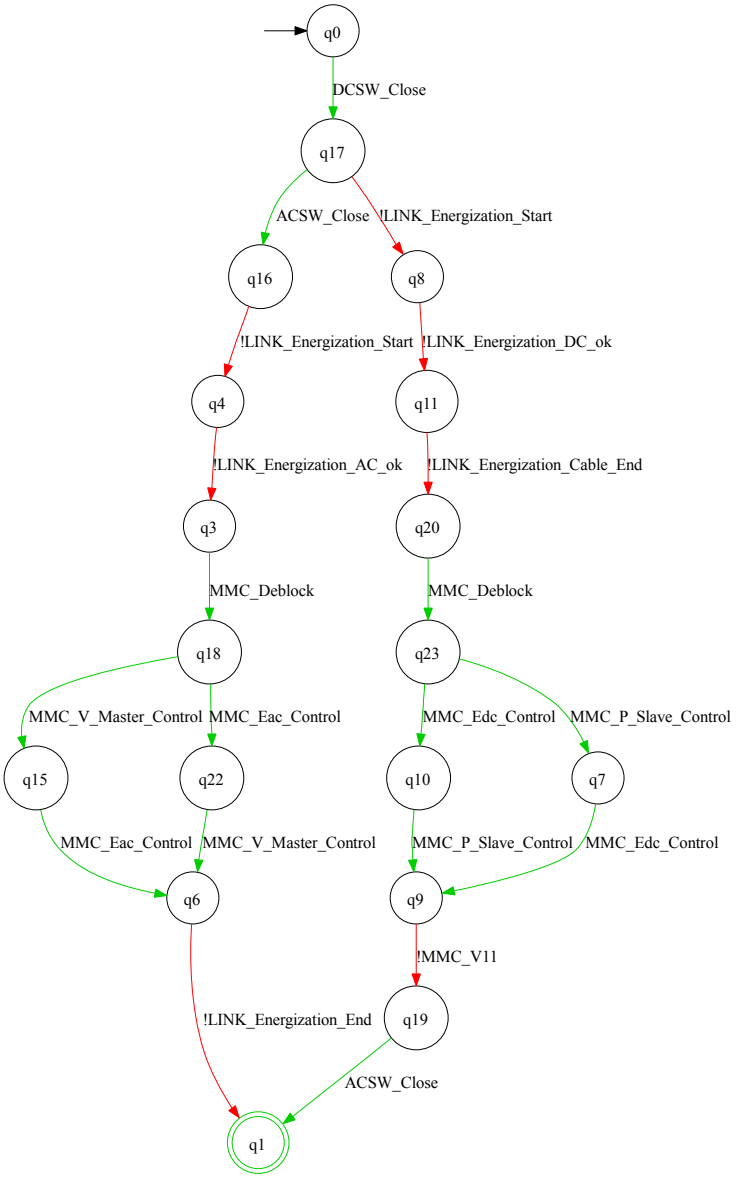



Figure 5.5 – Internal controlled plant of the station $H_{in}^{st,M_2}$ during start-up

the possible corrections.

The supervisory control internal to each mode is constructed by means of the compositional approach described in Section 4.3, such that at the end of the intramodal study the internal controlled plants of the stations and the internal controlled plant of the grid are obtained. In this PhD thesis, one procedure per mode is considered. In consequence, at the end of the intramodal study, the internal controlled plants are defined such that $H_{in,i}^{st,M_j} = H_i^{st,P_j}$ and $H_{in}^{gr,M_j} = H^{gr,P_j}$.

If the internal controlled plants have the expected behavior and are validated by all the collaborators, we are certain that the specifications of the modes are respected by definition. Nevertheless, these models do not represent all the possible commutations with the other modes. Some dynamics are missing, and so it is necessary to realize an intermodal study so as to take

into account all the mode commutations.

⌜ *Example*

As a reminder, the abstracted model of the controlled plant of the station obtained in Chapter 4 for the start-up of a point-to-point link is given in Figure 5.5. ⌟

## 5.4 Intermodal study

The objective of the intermodal study, the third step of Figure 5.1, is to extend the models internal to a mode to take into account all the possible commutations and their impact on the dynamics of the plants, while ensuring that the intermodal specifications are respected. For this, the designer must model the mode-switching specifications along with the modes automaton, so that the commutative behavior is taken into account.

### 5.4.1 Extension of the controlled plants

The modes of an HVDC system are modeled by the $G^{\mathcal{M}}$ automaton in Figure 5.4. Because the intramodal specifications (Sections 4.3.2 and 6.4.2) are built exclusively on the event sets generated by the components used in the modes, it is necessary to define intermodal specifications in order to couple both the internal alphabets $\Sigma_{in,i}^{st,M_j}$ and $\Sigma_{in}^{gr,M_j}$ of the controlled plants in a mode and the set $\Sigma^{\mathcal{M}}$ of commutation events of $G^{\mathcal{M}}$, as $\Sigma^{\mathcal{M}} \cap ((\cup_{i=1}^{n} \Sigma_{in,i}^{st,M_j}) \cup \Sigma_{in}^{gr,M_j}) = \emptyset$.

In consequence, a new model $E_{\rightleftarrows}^{M_j}$ representing the mode-switching specifications is added in this step. This model allows to place the language of the internal controlled plants in the corresponding mode of the modes automaton. Formally, the specification model $E_{\rightleftarrows}^{M_j}$ is built from the parallel composition of elementary specification automata $E_{\rightleftarrows}^{l,M_j}$, such that $E_{\rightleftarrows}^{M_j}$ is defined by $E_{\rightleftarrows}^{M_j} = (X_{\rightleftarrows}^{M_j}, \Sigma_{\rightleftarrows}^{M_j}, \xi_{\rightleftarrows}^{M_j}, x_{0,\rightleftarrows}^{M_j}, X_{m,\rightleftarrows}^{M_j})$, and $E_{\rightleftarrows}^{M_j} = \|_l E_{\rightleftarrows}^{l,M_j}$.

The designer has then at its disposal the abstracted plant of the station and the controlled plant of the grid, denoted respectively by $\tilde{H}_{in,i}^{st,M_j}$ and $H_{in}^{gr,M_j}$, the modes automaton $G^{\mathcal{M}}$ and the models of the intermodal specifications $E_{\rightleftarrows}^{M_j}$. However, at the station level, the models extending the mode-switching transitions to the internal behavior are based only on the local alphabet of the station. In consequence, they are respectively denoted by $E_{\rightleftarrows,i}^{M_j}$. $\tilde{H}_{in,i}^{st,M_j}$ and $H_{in}^{gr,M_j}$ can then be respectively extended by synthesis to the models $H_{st,i}^{M_j}$ and $H_{gr}^{M_j}$, which consider the commutative behavior of the system, as follows:

***Definition 5.2 (Extension of the controlled plants)***
$H_{st,i}^{M_j} = (Y_{st,i}^{M_j}, \Sigma_{st,i}^{M_j}, \tau_{st,i}^{M_j}, y_{0,st,i}^{M_j}, Y_{m,st,i}^{M_j})$ *such that* $L_m(H_{st,i}^{M_j}) = [L_m(H_{in,st,i}^{M_j} \parallel G^{\mathcal{M}} \parallel E_{\rightleftarrows,i}^{M_j})]^{\uparrow c}$;
$H_{gr}^{M_j} = (Y_{gr}^{M_j}, \Sigma_{gr}^{M_j}, \tau_{gr}^{M_j}, y_{0,gr}^{M_j}, Y_{m,gr}^{M_j})$ *such that* $L_m(H_{gr}^{M_j}) = [L_m(H_{in,gr}^{M_j} \parallel G^{\mathcal{M}} \parallel E_{\rightleftarrows}^{M_j})]^{\uparrow c}$. ♦

⌜ *Example*

For example, the abstracted plant of the station shown in Figure 5.5 is extended by synthesis with respect to a set of specifications that model the fact the DCSW_Close must take place after the commutation event Start_M2 in $G^{\mathcal{M}}$ and that the commutation event End_M2 takes place after LINK_Energization_End and ACSW_Close have occurred, i.e. the mode is deactivated at the station level once the MMCs and the cables are energized and the station is connected to the adjacent AC grid. The extended automaton is given in Figure 5.6. ⌟

### 5.4.2 Validation of the intermodal study

The designer has once again the opportunity to validate the built models. As in the validation stage of the intramodal study, the two questions that arise concern the existence of $H^{M_j}$ models and the behavior expected to be modeled by them. If the $H^{M_j}$ models do not exist, it will be necessary to review the commutation specifications because it is above all these specifications that can be problematic, as the models $G^{C_i,M_j}$ and $G^{\mathcal{M}}$ have already been validated. In practice, these specifications are very simple to write and it is therefore unlikely that, if these constraints effectively prevent commutations from taking place, the whole mode management model needs to be reviewed. In the case where all the models $H^{M_j}$ exist, it is possible once again that the behavior of the controlled plant is too restrictive. This restriction could only be caused by the commutation specifications. It is therefore important for the designer to identify the problematic behaviors, and to provide specifications to prevent unwanted behaviors in order to obtain validly controlled plant behavior.

### 5.4.3 Comparison with the monolithic approach

Although the proof of the equivalence between the proposed design approach and the centralized method is out of the scope of this PhD thesis, some guidelines are given next.

Because the alphabet internal to a mode and the set of commutation events are disjoint, the intramodal specification automata, as well as the physical constraint plants, should be extended in the centralized approach so as to take into account the commutative behavior. These extended models, obtained by the parallel composition of the elementary extended automata, can be respectively denoted by $E_{\circlearrowleft}$ and $G_{\circlearrowleft,pc}$. Because the compositional approach is no longer used, those automata involving the abstraction events (page 90) would not be necessary.

In summary, the controlled global plant $H = (Q, \Sigma, \delta, q_0, Q_m)$ obtained by monolithic synthesis would be calculated such that: $L_m(H) = [L_m(G \parallel E \parallel G^{\mathcal{M}})]^{\uparrow c}$, with $\Sigma = (\cup_{i=1}^{n} \Sigma^{C_i}) \bigcup \Sigma^{\mathcal{M}}$, $G = (\|_i G^{C_i}) \parallel (\|_k G_{\circlearrowleft,pc_k})$ and $E = (\|_l E_{\circlearrowleft}) \parallel (\|_l E_{\rightleftarrows})$. Hence, taking both the extended intramodal $E_{\circlearrowleft}$ and intermodal specifications $E_{\rightleftarrows}$ into account ensures the respect of both the internal and commutative constraints in the centralized approach. However, this extension can-

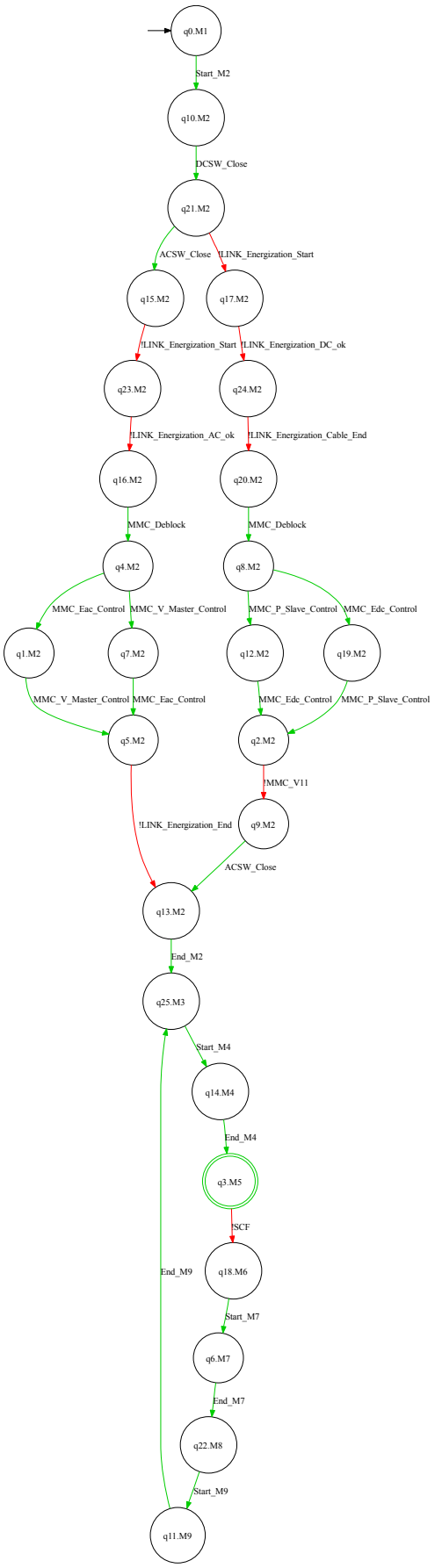



Figure 5.6 – Extended controlled plant of the station $H_{st}^{M_2}$ during start-up

not be done automatically, as the addition of the commutation events depends on the interaction of the events already present with those that will be added. This can only be carried out by an expert manually and the models might be substantially different from those in the presented approach, although the final result should be equivalent in terms of supervision.

## 5.5 Merging of the non-significant states

The last step of the mode-switching control design (Step 4 in Figure 5.1) concerns the reduction of the obtained models' complexity so as to keep only the internal behavior and the commutative behavior of each mode [Far10]. To obtain this reduced dynamics, the automaton of modes $G^{\mathcal{M}}$ allowed to include in each state of the models the information concerning the active mode. It is thus possible to isolate the states corresponding to the internal behavior of the system in each mode and to merge the states that are not included in the internal behavior of the mode (cf. Section 3.4.3). Thus, the set of states that do not correspond to the studied mode, which are called *non-significant states*, form a new state called *inactive state*, denoted by $y_{id}^{M_j}$, which represents the inactivity of the mode. This state is used to highlight the unicity of active mode, such that for $N$ modes, $N-1$ modes will be in this idle state and only one mode will be active. The new controlled plant models, where the non-significant states have been merged into the inactive state, are respectively denoted by $H_{merge,st,i}^{M_j}$ and $H_{merge,gr}^{M_j}$ for the station and grid levels.

Because parallel composition synchronizes common events, when the $N-1$ modes are in their inactive state, it must be ensured that they will not restrict the behavior of the mode that represents the current system behavior. For this reason, it must be ensured that the inactive state can generate all the possible events, either in a loop or out of this state. Thus, this step usually increases the number of transitions of the merged models given that events that are not included in the internal alphabet of the mode might be added as a self-loop transition. However, the states of the models are greatly reduced. Formally, if $Y_{mer}^{M_j}$ is the set of insignificant states, the automaton $H_{merge,gr}^{M_j}$ (cf. Definition 3.21) is built as proposed in [Far10]:

***Procedure 1***

*Considering the automata $H_{st,i}^{M_j}$, $H_{gr}^{M_j}$, $H_{merge,st,i}^{M_j}$ and $H_{merge,gr}^{M_j}$ such that $H_{merge,st,i}^{M_j}$ and $H_{merge,gr}^{M_j}$ are the models obtained by merging states of the $H_{st,i}^{M_j}$ and $H_{gr}^{M_j}$ automata, respectively, the merging procedure is described as follows:*

1. *We determine in $H_{st,i}^{M_j}$ (respectively, $H_{gr}^{M_j}$), a set of states to be merged in $Y_{mer,st,i}^{M_j} \subset Y_{st,i}^{M_j}$ (respectively $Y_{merge,gr}^{M_j} \subset Y_{gr}^{M_j}$). The states included in $Y_{merge,st,i}^{M_j}$ are the non-significant states of the mode $M_j$ whose name is not composed of $M_j$ (this particularity is given by the mode automaton $G^{\mathcal{M}}$ in the extension of the controlled plants);*
2. *All states in $Y_{merge,st,i}^{M_j}$ (respectively, $Y_{merge,gr}^{M_j}$) are replaced by a new state called $y_{id,st,i}^{M_j}$*

*(respectively, $y^{M_j}_{id,gr}$);*

3. *If the initial state is included in $Y^{M_j}_{merge,st,i}$ (respectively, $Y^{M_j}_{merge,gr}$), then $y^{M_j}_{id,st,i}$ (respectively, $y^{M_j}_{id,gr}$) is the new initial state. Otherwise, the initial state remains that of the model $H^{M_j}_{st,i}$ (respectively, $H^{M_j}_{gr}$);*
4. *If at least one marked state is included in $Y^{M_j}_{merge,st,i}$, then $y^{M_j}_{id,st,i}$ is a marked state (similarly for $Y^{M_j}_{merge,gr}$).* ♦

As a result of this merge function, the designer has a mode template that represents the internal behavior of the system in the mode in which the commutation events activate or deactivate the mode by entering or exiting the inactive state. These distinct behaviors respect the constraints related to each mode and the designer is able to extract the control law to be implemented.

⌜ *Example*

For example, all the states in the extended plant of the station that did not correspond to $M_2$ have been merged into an inactive state. Because the marked state of the extended model is not included in $M_2$, we mark the inactive state. The resulting merged model is presented in Figure 5.7. All the event of the extended model and the modes automaton have been added in a self-loop transition at the inactive state, except for the commutation event Start_M2 that activates the mode. ⌟

Then, for the global equivalent behavior (see Definition 3.20) obtained by means of the parallel composition of the $H^{M_j}_{merge}$ automata for each mode $M_j$ of $G^{\mathcal{M}}$ to be non-blocking, the following conditions should be fulfilled:

***Proposition 5.1 (Non-blocking global equivalent behavior)***
*Considering the merged models $H^{M_j}_{merge}$ for the station and grid levels in a mode $M_j$, and the modes automaton $G^{\mathcal{M}}$, the global equivalent behavior is non-blocking if the following conditions are met:*

***Condition 1**: The modes automaton $G^{\mathcal{M}}$ is non-blocking, i.e. the set of marked states can be always reached. Formally, $L_m(G^{\mathcal{M}}) = \overline{L(G^{\mathcal{M}})}$.*

***Condition 2**: The behavior of each mode $M_j$ is non-blocking at the grid level. Formally, $\forall M_j \in \mathcal{M}, \{L_m(H^{M_j}_{merge,gr}) = \overline{L(H^{M_j}_{merge,gr})}\}$.*

***Condition 3**: The behavior of each mode $M_j$ is non-blocking at the station level. Formally, $\forall M_j \in \mathcal{M}, \{L_m(H^{M_j}_{merge,st}) = \overline{L(H^{M_j}_{merge,st})}\}$.* ♦

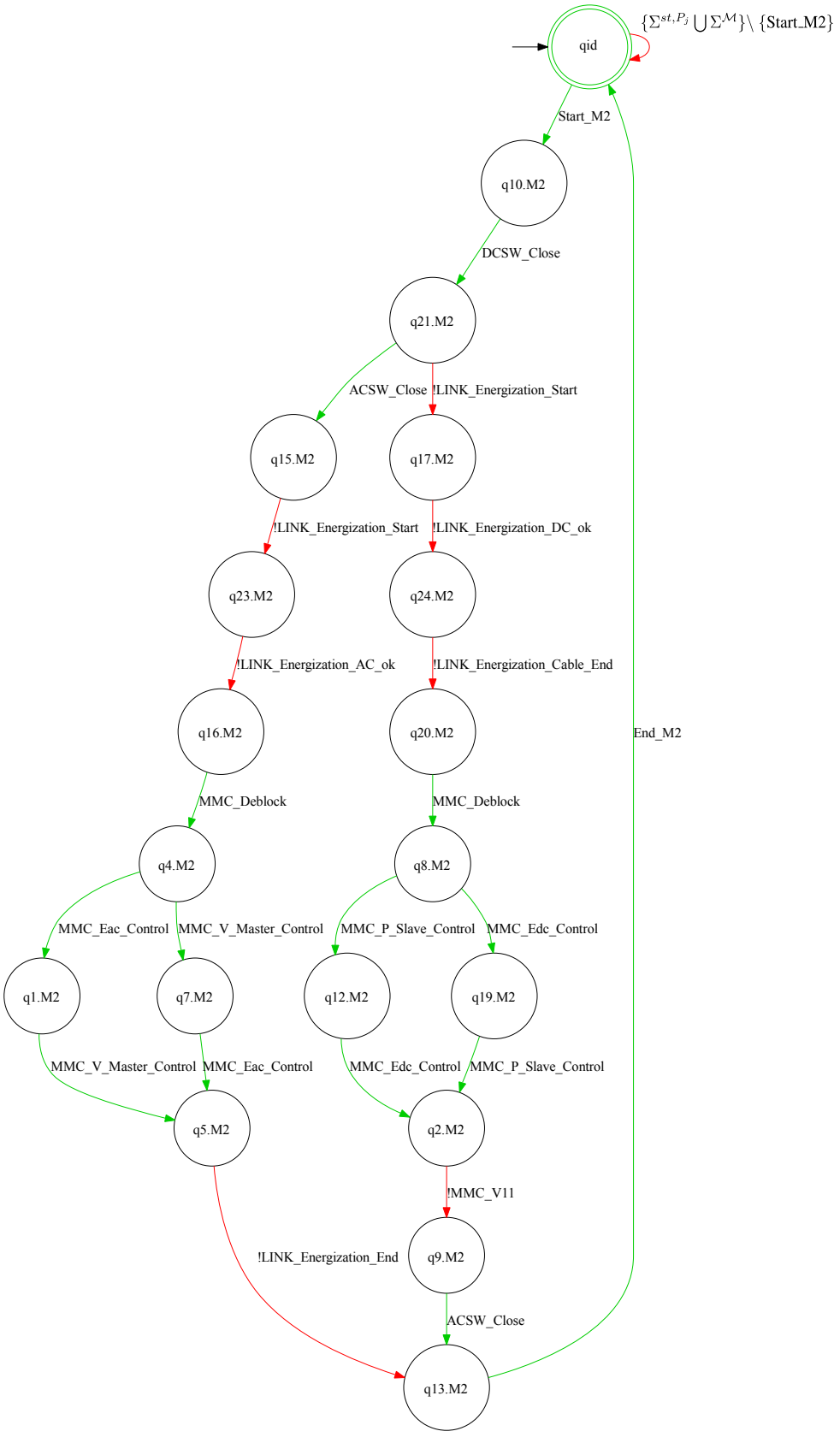



Figure 5.7 – Merged model $H_{st}^{M_2}$ of the station during start-up

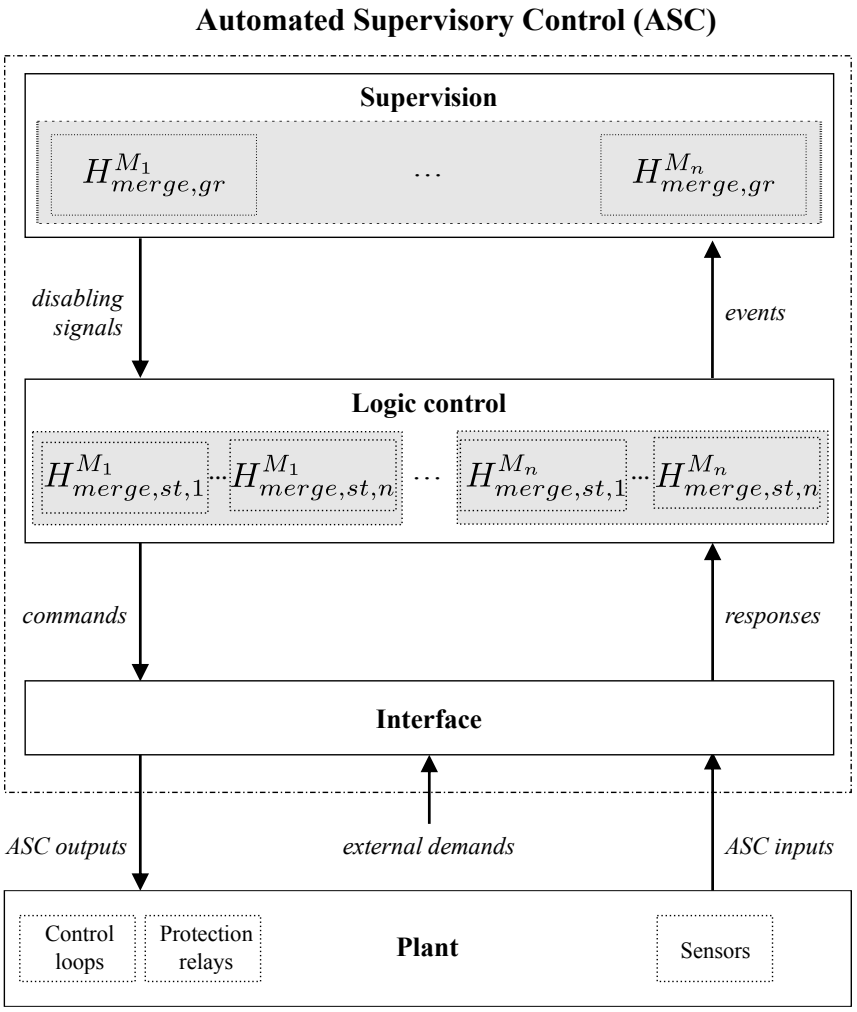



Figure 5.8 – Control structure of a mode-switching supervisory control for HVDC systems

## 5.6 Implementation

The basis for the implementation of the mode-switching control obtained as a result of the method proposed in this chapter is the implementation approach presented in Section 4.4. Hence, the implemented supervisory control is equally divided into three control levels, such that the merged controlled plant of the grid of each mode is implemented in a *Supervision* level that realizes supervision functions, the controlled plants of the different stations in each mode are implemented in a *Logic control* level that realizes classical control functions, and the *Interface* level relates the continuous-time signals from the sensors and the digital signals sent to the actuators to their discrete-event counterpart used in the ASC, as illustrated in Figure 5.8.

However, because the commutation events, if controllable, are not associated to any digital signal of the plant, they do not generate any associated command but only modifies the event signal when the attached mode-switching transition occurs. Furthermore, as some events may be shared between the models of different modes, it is necessary to distinguish at the supervision

level the mode that has generated an event, so that a command associated to a controllable event authorized by an inactive mode but disabled by the active mode is not generated. Because the mentioned problem concerns the "synchronization" of the disabling signals of the different mode models, the command and response signals are not affected, as the supervision level generates the disabling signals based only on the event signals. In consequence, the output expressions defined in Section 4.4.3 for the logic control level are redefined such that:

$(\omega_{st,a}^{M_j})$ $\sigma_v^{M_j} := 1; cmd\sigma_v := 1$

$(\omega_{st,b}^{M_j})$ $\sigma_v^{M_j} := 1$

Similarly, the input and output expressions defined in Section 4.4.2 for the supervision level are reformulated, with:

$(\varphi_{gr}^{M_j})$ $\sigma_v^{M_j}$

$(\omega_{gr}^{M_j})$ $\sigma_v d^{M_j} := 1$

Then, it is necessary to define a logical function that merges the disabling signals $\sigma_v d^{M_j}$ of each mode $M_j \in \mathcal{M}$ with respect to an event $\sigma$. This function is chosen to follow the conjunctive fusion rule described in Section 3.4.1. For this, a signal $\sigma d$ is associated to the controllable events that are generated by the extended Moore machines $M_{merge,st,i}^{M_j}$ at the logic control level.

Thus, the merged disabling signal $\sigma_v d$ is activated when at least one of the disabling signals $\sigma_v d^{M_j}$ ($M_j \in \mathcal{M}$) deactivates the occurrence of the corresponding event $\sigma$. In consequence, all the models $M_{merge,st,i}^{M_j}$ are synchronized, as they observe the same disabling signal for a given event. This is shown in expression $(\omega_{mer}^{M_j})$ for the case of two modes:

$(\omega_{mer}^{M_j})$ IF $(\sigma_v d^{M_1} = 1$ OR $\sigma_v d^{M_2} = 1)$ THEN $\sigma_v d := 1$

The logical function *merge_ds* that merges the different disabling signals is then coded in C, according to the instruction given in Section 4.4.5. As an example, a code snippet of the *merge_ds* function is shown in Listing 5.1 for the some of the events involved in the start-up of a point-to-point link (cf. Section 4.3.1):

```
1 void merge_ds(int* CloseACSW1d, int CloseACSW1d_M1, ...){
2
3   *CloseACSW1d = CloseACSW1d_M1 || CloseACSW1d_M2;
4   *CloseDCSW1d = CloseDCSW1d_M1 || CloseDCSW1d_M2;
5   ...
6 }
```

Listing 5.1 – Code snippet of the *merge_ds* function

The *merge_ds* function is then called within the main *ASC* function and after the *supervision* function. Hence, the final order of execution of the automated supervisory control program is determined to be the following:

1. Reset disabling variables $\sigma_v d$
2. Call *supervision* function

3. Call *merge_ds* function
4. Reset event variables $\sigma_v$
5. Call *logic control* function
6. Reset response variables $rsp\sigma_v$
7. Call *interface* function
8. Reset command variables $cmd\sigma_v$

To conclude, the code generation associated to the *merge_ds* function can be automated given that the information necessary to construct the generic expression is provided by the modes automaton and the controlled plants implemented at the logic control and supervision level.

## 5.7 Conclusion

In this chapter, the compositional approach proposed in Chapter 4 is extended so that the supervisory control is able to modify the dedicated control map whenever the HVDC systems commutes from one mode to another during its operation. The mode-switching approach used in this PhD thesis is based on the modal decomposition approach (cf. Section 3.4.3, [Far10]). Contrary to the work presented in [Far10], however, the commutation events in an HVDC system are not generated by a single component, but they are rather dependent on the reaching of a certain state by the whole set of components in the system.

In consequence, it is necessary to verify that the model of the mode-switching behavior is coherent with the component models for each mode. Then, the compositional approach described in Chapter 4 can be applied in the intramodal study so as to obtain a supervisory control internal to each considered mode. The obtained models are then extended in order to take into account the commutative behavior of the system in the intermodal study. If the resulting models are validated, they can be simplified by merging the non-significant states, such that only the states associated to the considered mode are preserved. Finally, the implementation method described in Section 4.4 is further developed in Section 5.6 so as to integrate the modal architecture within the proposed multilevel implementation.

# 6

# Application to a 3-terminal MTDC grid

## Contents

## 6.1 Introduction

The purpose of this chapter is to show the applicability of the compositional and modal approach proposed in chapters 4 and 5 to a large HVDC system with several modes of operation. The considered case is a bipolar MTDC system with three terminals whose operation is determined by the mode commutation given in Figure 5.2. The next section is devoted to the presentation of the studied system and the expected behavior from each of the considered modes. The rest of sections are dedicated to the different steps of the design and implementation approach, as illustrated in Figure 5.1 (page 110) and Figure 4.2 (page 65).

## 6.2 Presentation of the case study

The case study corresponds to a bipolar multi-terminal HVDC grid consisting of three converter stations per pole, as illustrated in Figure 6.1, during an operation cycle that comprises the mode-switching behavior presented in Figure 5.2.

### 6.2.1 Description of the system

The considered system comprises two poles with opposite voltage polarity, such that each pole is controlled independently. Each pole is connected to its adjacent AC grid via a set of converter stations. Furthermore, all the stations in the same pole are interconnected by means of HVDC cables that transfer the power throughout the grid. In this manner, if one cable is out of service, the power can still be transferred as there exists an alternative electrical path.

Because of the bipolar topology (see Section 2.3, page 19), the mid-point of the two cascaded converter stations is connected to an earth reference. In consequence, the lower output of the MMCs at the positive pole is connected to the earth and inversely for the MMCs in the negative pole. The earth reference is located at Station 1 while the rest of them are connected to the common reference by means of a dedicated low voltage metallic return in order to prevent the current from flowing freely into the earth.

The different cables connected to each station are joint at the DC side of the station by means of a busbar (denoted by N1, N2 and N3 in Figure 6.1), so that the sum of the power transferred by each cable connected to the bus is equal to the total power transferred through the converter. However, because each station in an MTDC architecture is connected to several cables, the configuration of the stations is slightly modified with respect to the one presented in Figure 2.9. In fact, a redundancy of the elements composing the protection subsystem on the DC side of the station (cf. Section 2.3.2, page 25) needs to be introduced to protect all the connected cables. In consequence, the DCCBs are distinguished according to their location. If the DCCB is located between the busbar and the converter, it is referred to as a converter breaker and is denoted by

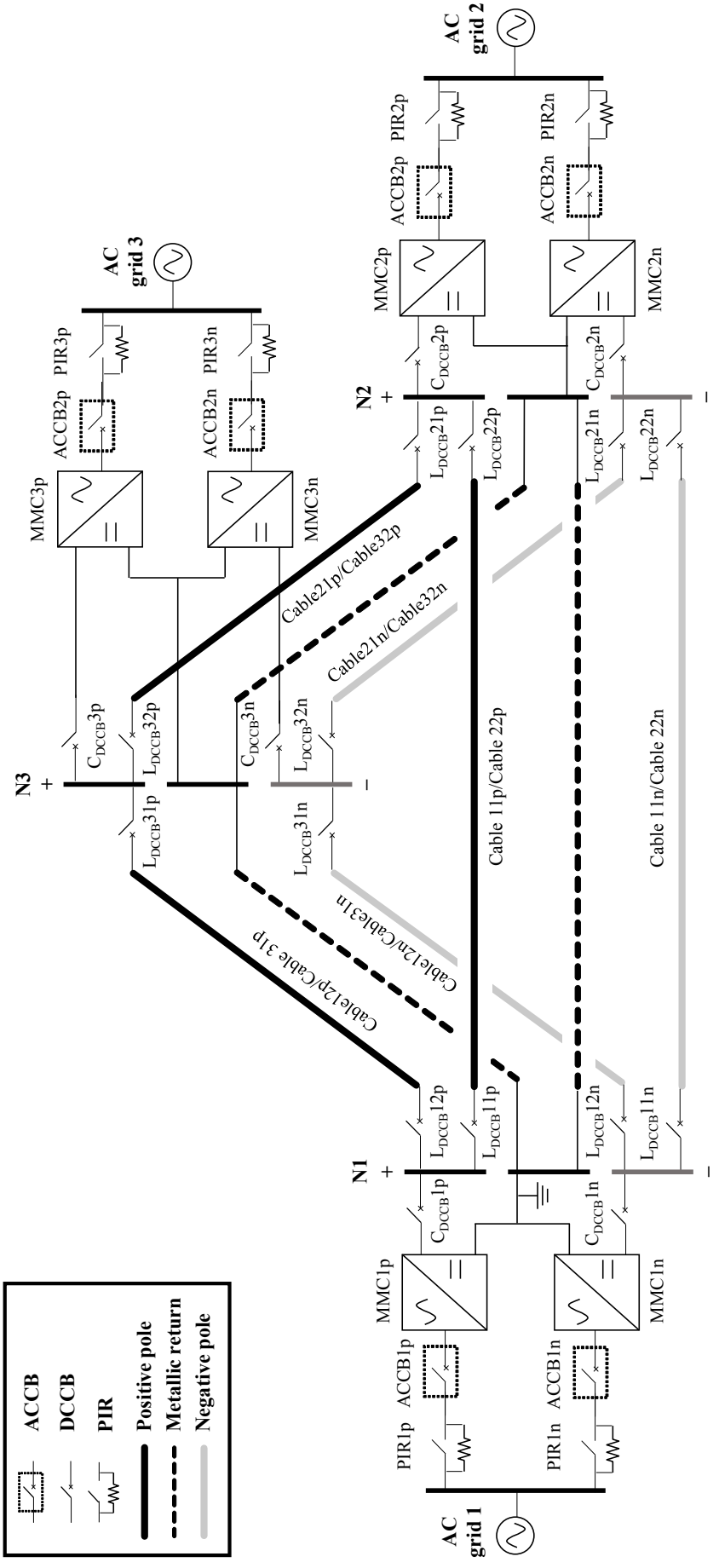



Figure 6.1 – Three terminal bipolar MTDC grid

$C_{DCCB}$. If the module is located between the busbar and the line or cable, it is called a line breaker and denoted by $L_{DCCB}$. While the algorithms implemented in the relay associated to a $C_{DCCB}$ rely on local data only, those implemented in the relay controlling a $L_{DCCB}$ can be based on information communicated from the relay located at the opposite end of the cable or the other relays of the station, in addition to the local measurements. The communicated data are either binary signals or current and voltage measurements.

Finally, because a short-circuit fault is considered to be correctly isolated by the protection subsystems in their first attempt, the breaking modules (BMs) (cf. Figure 2.14) are solely composed by the DCCBs in our case study, as the High-Speed Switches (HSSs) are not necessary. Similarly, the Pre-Insertion Resistor (PIR) modules are not needed on the DC side of the station, as opposed to the AC side. To conclude, both the ACCB and PIR module are commanded through the same protective relay, while each DCCB is commanded by a dedicated relay.

### 6.2.2 Dedicated software

To deal with this example, we will use several software. Firstly, the software Supremica [Mal17] is used for the tasks related to the design of the supervisory control, such as model construction, model abstraction and controlled plant synthesis, because of its large library of operators on automata. Secondly, in order to generate the C code resulting from the implementation method, a computer program based on Python [Oli07] has been developed within this thesis. The built software takes as input an XML file created by Supremica that contains the automata to be implemented and permits to quasi-automatically[1] generate the C code corresponding to the associated supervisory control, according to the implementation method proposed in sections 4.4 and 5.6. Finally, due to the large size of power transmission systems, their complexity and the high cost associated to their construction and their operation, any device or control that is to be used in the system should first be tested and validated in a simulation environment. Specifically, ElectroMagnetic-Transient Programs (EMTP) are software tools allowing to analyze the power system dynamics quite accurately. The EMTP software used in this PhD thesis is EMTP-RV [Mah07]. This program allows to build a model of the system to be simulated by means of a dedicated library of components including HVDC components, such as the MMC. The C code containing the supervisory control program has been implemented in EMTP-RV under the form of a Dynamic-Link Library (DLL) [Har05] in order to perform an offline simulation of the considered case study.

[1] As stated in Section 4.4, the information concerning the inputs and outputs of the automated supervisory control (associated to the corresponding events by the Interface level) must be manually entered during the implementation, given that they are specific to the system.

## 6.3 Model construction

The first step of the design of a supervisory control is the construction of the modes automaton and the models of the components that determine the internal behavior of each mode.

### 6.3.1 Modes automaton

The objective of this chapter is to develop an automated supervisory control that realizes mode management and coordinates the actions to be performed within each mode for the system presented in Figure 6.1. The modes considered for the operation of the 3-terminal MTDC grid are the nine modes listed in Table 5.2, according to the dynamics given in Figure 5.2 and modeled by the modes automaton of Figure 5.4.

- Thus, the system is at first in mode $M_1$, which corresponds to a state where all the components are discharged and all the CBs are open. Therefore, no current flows through the grid. Because it is considered that no fault occurs at this stage, this mode does not have an internal behavior.
- Then, the system enters the start-up mode $M_2$ on demand. The start-up strategy considered in this chapter is the one studied for the point-to-point link but extended to the case of a 3-terminal network. Thus, only one of the stations will be chosen as the source station, while the others will follow a passive charging from the DC grid. Consequently, as a first step, only the AC grid adjacent to the source station is connected to the HVDC system. A first uncontrolled charging phase will follow. Once the voltage level reached by the SMs of the source MMC is high enough to synthesize a waveform equivalent to the sine waves on the AC side of the station, the source MMC's controller will be deblocked and the voltage in the converter and the cables regulated to reach their nominal value. Once this controlled charging is finished, the MMC will be able to maintain the balance between the synthesized voltages on both the AC side and the DC side by means of the inner energy regulation and the DC voltage control. This will result in a controlled load of the DC cable voltage. During this phase, the voltage in the converters of the passive stations will undergo a second uncontrolled charging phase, until a voltage value of 0.5 p.u. is reached. Finally, the controllers associated to the converters in the passive stations are deblocked one at a time so that the converters are not charged simultaneously, and the voltage level of their SMs brought to 1 p.u. by means of the MMC voltage control. At this point, all the MMCs are able to replicate the AC sine waves with the desired amplitude. In consequence, the passive stations are finally connected to their adjacent AC grids.
- Once $M_2$ is completed, the system enters the mode $M_3$, which corresponds to the state where all the components in the grid are energized and the voltage is regulated but with no power transferred between the stations. Thus, the power in the grid is practically zero

in this mode. Because it is considered that no fault occurs at this stage, this mode does not have any internal behavior.

- Then, the current references can be modified (while keeping the DC voltage constant) by the human operator or the dispatch controller at the control center in order to obtain the desired power flows. To achieve this, the supervisory control activates the power ramp mode $M_4$ so as to configure the MMC controllers accordingly. Hence, those stations where the power references are imposed by the adjacent AC grid and cannot be regulated (e.g. an offshore wind farm) are configured to be controlled in power control. The other stations in the grid are configured in voltage droop control so that they are able to compensate for any power disturbance (within the control limits of the converter) while maintaining the DC voltage in the grid around its nominal value.
- At this stage, the system is in the nominal (marked) mode $M_5$, where all the MMCs are charged and controlled, and the voltage and current in the cables are regulated so as to transfer the amount of power determined by the TSO while maintaining the power balance in the system. However, a short-circuit fault can arise when the system is in $M_5$.
- If a short-circuit fault appears, the system enters the mode $M_6$ uncontrollably (and drastically). Indeed, because the cables are now directly connected to the earth, the DC voltage collapses to zero. The MMCs are automatically blocked by means of a local protection that forces the current to pass through the diodes in the SMs, in order to prevent the IGBTs and the SM capacitors from being damaged. However, if the faulty cable is not isolated, the current injected into the HVDC grid by the surrounding AC grid continues to feed the short circuit and charges the cable. Because the cable voltage does not increase due to the fault, the injected current will continue to rise abruptly, rapidly provoking an overcurrent situation. Thus, the system will remain in $M_6$ as long as the faulty cable is not isolated.
- Therefore, the protection mode $M_7$ needs to be immediately activated as soon as the fault is detected. In our case study, the non-selective procedure described in [Lou17] is considered. Hence, as soon as the measured DC current exceeds the overcurrent threshold, the converter breakers $C_{DCCB}$ before the bus voltage will be opened automatically by the protection relay of each station. In this manner, since the surrounding AC grid is disconnected from the DC system, the current feeding the fault will be extinguished after a certain time. Simultaneously, the converter will be blocked autonomously thanks to an internal protection algorithm, that is, this action will be seen as uncontrollable by the supervisor. Concurrently, the identification algorithm within each relay detects the cable impacted by the fault. Afterwards, the $L_{DCCB}$ located at each end of the faulty cable are locally commanded to open in order to isolate the latter. Finally, the converter breakers $C_{DCCB}$ in each station are reclosed so that all the stations can restart the power transfer via the remaining cables. Throughout the protection mode, the protection equipment (relays, CBs, etc.) act autonomously and so the supervisory control does not enforce any

command until the faulty cable is isolated. However, the internal behavior of the mode is modeled so that the ASC is able to follow the grid's evolution. To conclude, it is not within the scope of this PhD thesis to treat the case where the $C_{DCCBs}$ fail to open. In that case, a back-up protection procedure would need to be considered.

- Once the faulty line is isolated and all $C_{DCCBs}$ reclosed, the system enters the mode $M_8$. In this mode, the MMCs are still charged to their rated value as the SM capacitors are neither charged nor discharged while the converter was blocked. However, because of the DC voltage collapse due to the short circuit, all the healthy cables are discharged. Because it is assumed that the short-circuit fault has been effectively removed at this stage, $M_8$ does not have any internal behavior.
- Then, the voltage restoration mode $M_9$ is entered, in order to deblock the MMC controllers and bring back the cable's voltage to their rated value. During this mode, one MMC is selected to regulate the DC voltage while the rest of them control the power. Finally, at the end of this mode, the mode $M_3$ will again be reached. Then, in order to return to the "marked" mode $M_5$, all the converters will be configured to realize the power references provided by the control center, as described in $M_4$.

### 6.3.2 Component models

According to the previous description of the modes, the set $\mathcal{C}^{M_j}$ of the components in a station is summarized in Table 6.1 for each mode. As a reminder, because the mode-switching events are not included in the components alphabet, the set $\mathcal{C}^{M_j}$ is equivalent to the set $\mathcal{C}_{\circlearrowleft}^{M_j}$ of components that determine the intramodal behavior of a mode.

Concerning the models of the components in a station for each mode, they are defined according to Definition 4.3 and are illustrated in Figure 6.2. The particular component models for the mode $M_2$ are not shown in Figure 6.2, as they correspond to the ones given in Figure 4.17 for the start-up of a point-to-point link. However, as opposed to the point-to-point architecture, the DCCBs need to be differentiated in an MTDC grid. Therefore, the CDCCB, LDCCB1 and

| Mode $M_j \in \mathcal{M}$ | $\mathcal{C}^{M_j} = \mathcal{C}_{\circlearrowleft}^{M_j}$ |
|---|---|
| $M_1$ (OS000) | $\emptyset$ |
| $M_2$ (OM1) | $\{MMC, ACCB, CDDCB, LDCCBij, CABLEij\}$ |
| $M_3$ (OS110) | $\emptyset$ |
| $M_4$ (OM2) | $\{MMC, CABLEij\}$ |
| $M_5$ (OS111) | $\{CABLEij\}$ |
| $M_6$ (OS102) | $\emptyset$ |
| $M_7$ (OM3) | $\{MMC, CDDCB, LDCCBij, CABLEij\}$ |
| $M_8$ (OS100) | $\emptyset$ |
| $M_9$ (OM4) | $\{MMC, CABLEij\}$ |

Table 6.1 – Set $\mathcal{C}^{M_j}$ in $M_j$ for the station $i$ ($i \in \{1,2,3\}, j \in \{1,2\}$)

LDCCB2 models are all instantiated from the generic circuit breaker model (see Figure 4.14) but their events are renamed accordingly.

### 6.3.3 Validation of the consistency between models

This section is focused on the consistency of the models built with respect to Definition 5.1.

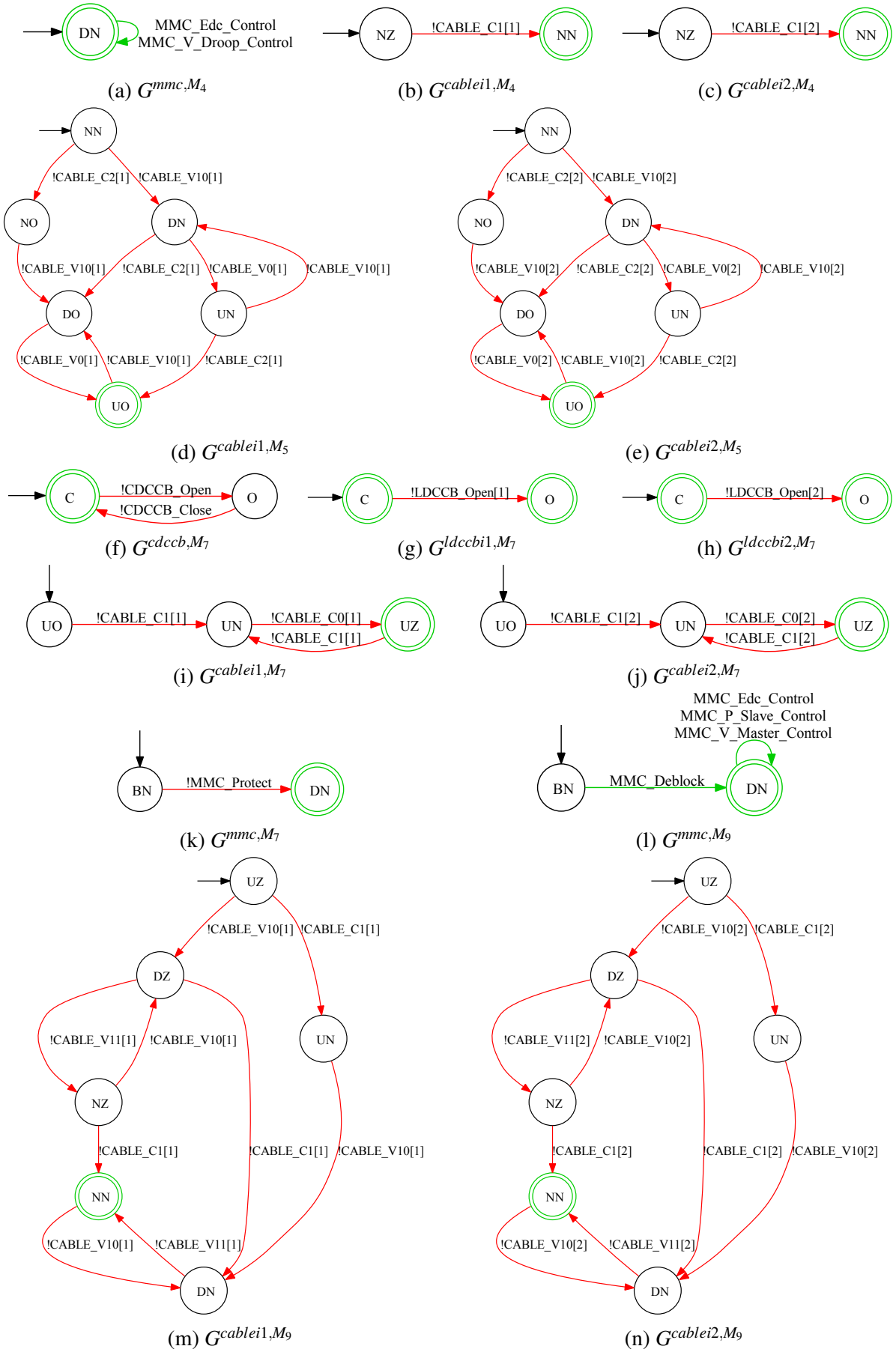
Firstly, Condition 1 is verified to be fulfilled for all the modes that do not present an internal behavior $(M_1, M_3, M_6, M_8)$, as shown in Table 6.1.

Then, Condition 2 is verified to be respected by all the component models in each mode. For instance, the models $G^{mmc,M_4}$, $G^{cable1,M_4}$ and $G^{cable2,M_4}$ are consistent with the modes automaton $G^{\mathcal{M}}$, given that the initial state of the models (DN, NZ, NZ) corresponds to the state identical to the configuration of the system in $M_2$ (see Figure 4.17), that is, all the cables in the DC grid are charged and the current in the system is zero. Also, the MMC is similarly charged and its controller deblocked. The circuit breakers have been already closed during the mode $M_2$ corresponding to the start-up and so their state is not modified in $M_4$. Furthermore, the product of the marked states (DN, NN, NN) corresponds to a configuration of the system where all the cables and MMCs are charged, the current circulating through the cables is non-zero and the converter controller is deblocked.

Similarly, the models $G^{cable1,M_5}$ and $G^{cable2,M_5}$ are consistent with $G^{\mathcal{M}}$. Indeed, the product of the initial states corresponds to a state where all the cables are charged and the DC current is non-zero (NN, NN), as the marked states in $M_4$, while the product of the marked states corresponds to the situation originated by a short-circuit fault, i.e. zero DC voltage and overcurrent (UO, UO), as in mode $M_6$. The state of the MMC and the CBs is not modified in $M_5$.

Next, the models $G^{cdccb,M_7}$, $G^{ldccb1,M_7}$, $G^{ldccb2,M_7}$ and $G^{mmc,M_7}$ are also validated. The models in $M_7$ are thus consistent with the adjacent modes $M_6$ and $M_8$ as the product of the initial states (DN, C, C, C, UO, UO) corresponds to a system where all the CBs are closed, the MMC is charged and deblocked and the state of the cables is not modified from $M_5$; while the product of the marked states ((BN, C, C, C, UZ, UZ), (BN, C, O, C, UZ, UZ) or (BN, C, C, O, UZ, UZ)) corresponds to a situation where the MMC has been autonomously blocked, the CDCCB is closed and the line breakers are either open or closed depending on the location of the short-circuit fault. The DC current in the cables falls to zero once the AC grid is separated from the fault by the DCCBs.

Finally, the models $G^{mmc,M_9}$, $G^{cable1,M_9}$ and $G^{cable2,M_9}$ are consistent with $G^{\mathcal{M}}$ as the product of the initial states (BN, UZ, UZ) correspond to a grid where all the cables are discharged and the MMC is blocked (as in $M_7$). The state of the CBs is not modified as the faulty cable remains isolated during this mode. The product of the marked states (DN, NZ, NZ), on the other hand, correspond to a grid where all the cables are charged, the MMC is charged and deblocked and the DC current is zero, as in $M_4$.




(a) $G^{mmc,M_4}$ (b) $G^{cablei1,M_4}$ (c) $G^{cablei2,M_4}$

(d) $G^{cablei1,M_5}$ (e) $G^{cablei2,M_5}$

(f) $G^{cdccb,M_7}$ (g) $G^{ldccbi1,M_7}$ (h) $G^{ldccbi2,M_7}$

(i) $G^{cablei1,M_7}$ (j) $G^{cablei2,M_7}$

(k) $G^{mmc,M_7}$ (l) $G^{mmc,M_9}$

(m) $G^{cablei1,M_9}$ (n) $G^{cablei2,M_9}$

Figure 6.2 – Particular component models in the station $i$ ($i \in \{1,2,3\}$) for each mode

## 6.4 Intramodal study

The intramodal study is the second step of the proposed approach, illustrated in Figure 5.1. This step consists of studying the internal behavior of each mode without taking into account the commutative behavior of the system. In this step, the controlled plants of the stations and the grid are built according to the compositional approach illustrated in Figure 4.2. Because of the high number of models built for this case study, only those that are considered to be the most representative of the method are illustrated in the following, at the expense of completeness.

### 6.4.1 Controlled plant of the station

#### Plant models construction

For each of the modes contained in $\mathcal{M}$ that have an internal behavior, we construct the uncontrolled plant of the station $G_{in}^{st,M_j}$ from parallel composition of the particular component models $G^{C_i,M_j}$, the physical constraints $G_{pc_k}^{C_i,M_j}$ related to the components and the physical constraints $G_{pc_k}^{st,M_j}$ related to the concurrent operation of the components in a station. The components taken for the construction of the plants are the components included in $\mathcal{C}_{\circlearrowleft}^{M_j}$ (Table 6.1).

Some of the physical constraints models are given in Figure 6.3. For instance, the automata in figures 6.3a to 6.3c model how the behavior of the circuit breakers is constrained during the start-up. Because no fault occurs during this mode, the circuit breaker is closed immediately after a closing request occurs. The automaton in Figure 6.3d, on the other hand, models the fact that the configuration of the MMC controller can either be chosen before the MMC voltage reaches its nominal value in case the converter works as a source, or after the SMs are completely energized if the MMC works in the remote role (see Figure 4.16a). The automaton in Figure 6.3e models the physical impossibility for the current to rise to the desired value if the MMC controller has not been previously configured in mode $M_4$. Concerning the mode $M_7$, the automata in Figure 6.3g and Figure 6.3f model the fact that the short-circuit fault current is effectively extinguished once the CDCCB is opened and that this occurs after the blocking of the MMC in terms of time. Furthermore, the automaton in Figure 6.3h models the different possible scenarios for a station once the fault is identified: if no cable connected to the station is impacted, the CDCCB is reclosed again, otherwise, the LDCCB associated to the faulty cable has previously been opened (only one cable is considered to be impacted within the grid in our example). Finally, the automata in Figure 6.3i and Figure 6.3j model a constraint similar to the one in $G_{pc_{1i}}^{st,M_4}$. However, because a short-circuit fault might have occurred before entering $M_4$, the automata take into account the scenarios where one cable is out of service.

The size of the obtained models is given in Table 6.2 for each mode with an internal behavior.

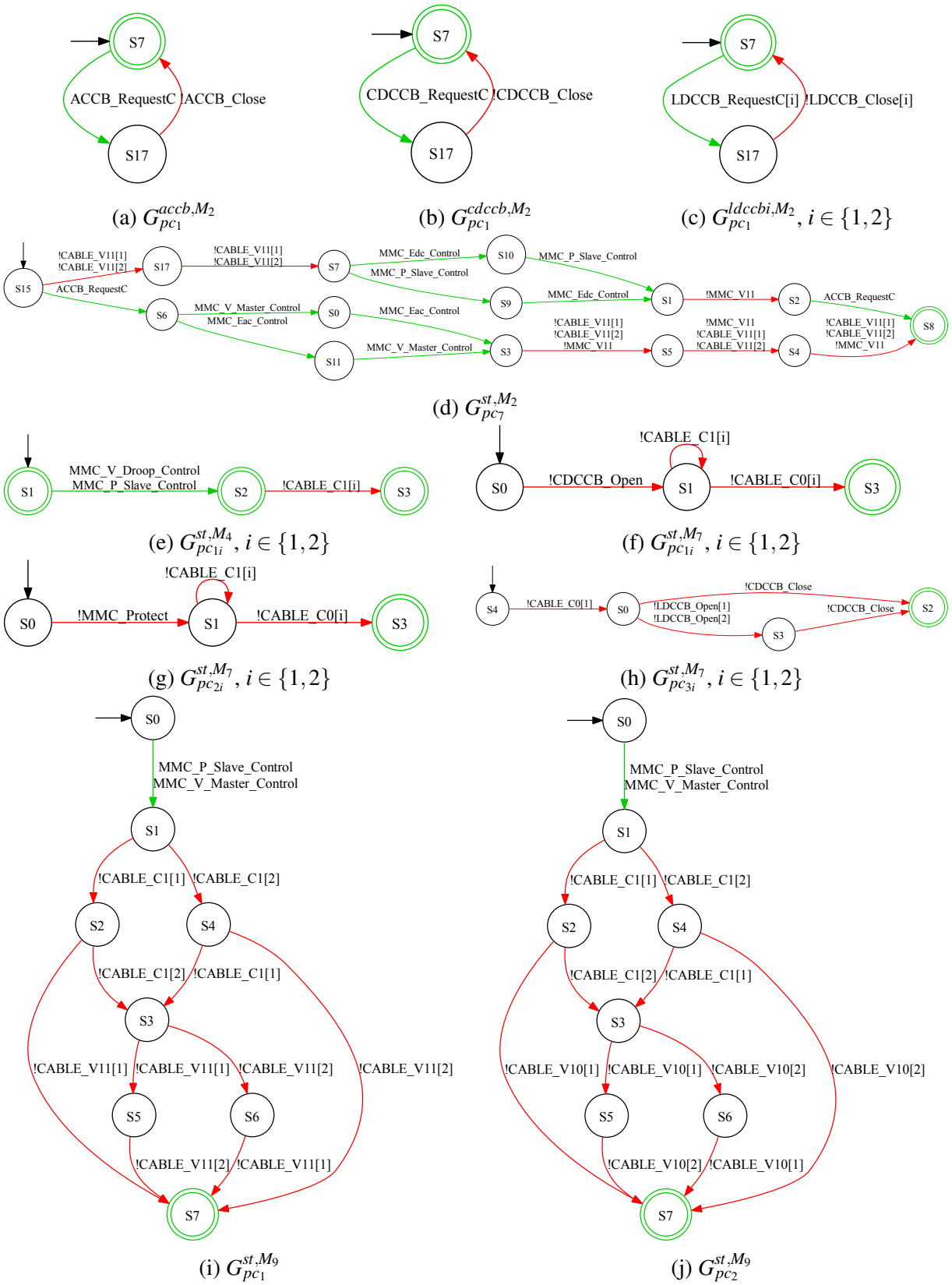


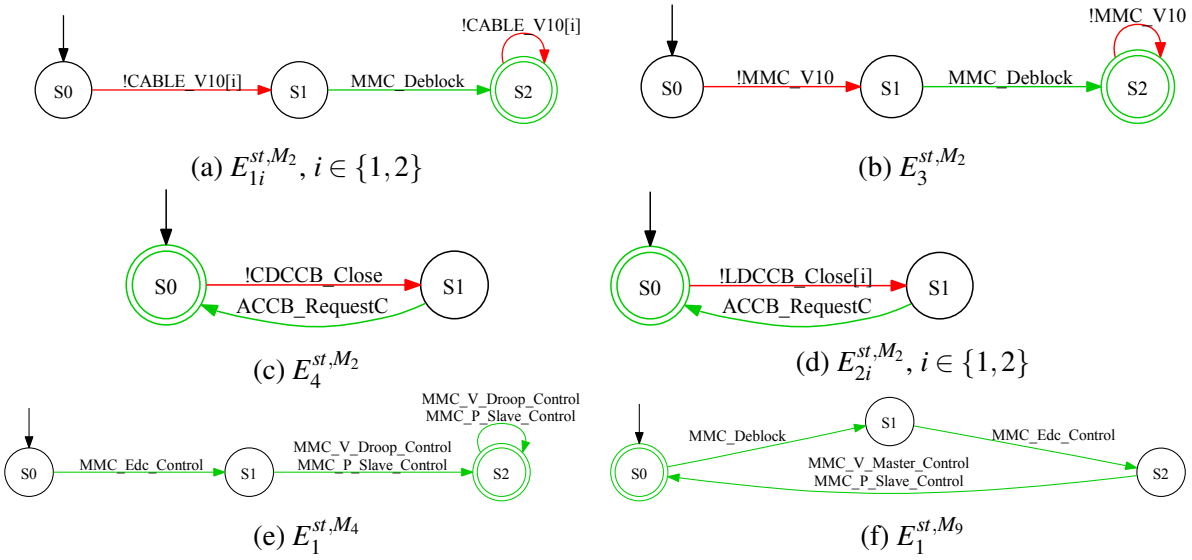

Figure 6.3 – Physical constraints models

| $G^{st,M_j} = (\|_{C_i \in C^{M_j}} G^{C_i,M_j}) \parallel (\|_{k \in \mathbb{N}} G^{C_i,M_j}_{pc_k}) \parallel (\|_{k \in \mathbb{N}} G^{st,M_j}_{pc_k})$ | No. of states | No. of transitions |
|---|---|---|
| $G^{st,M_2}$ | 210 | 496 |
| $G^{st,M_4}$ | 5 | 10 |
| $G^{st,M_5}$ | 36 | 84 |
| $G^{st,M_7}$ | 17 | 21 |
| $G^{st,M_9}$ | 14 | 34 |

Table 6.2 – Size of the uncontrolled plants of the station $G^{st,M_j}$

### Specification models construction

The specifications at the station level internal to each mode $M_j$ are modeled. Because the modes $M_5$ and $M_7$ describe the behavior of the grid during the short-circuit fault, which is uncontrollable to the supervisor as it is managed autonomously by the protection relays, no specifications are needed for these two modes. For the rest of modes with an internal behavior, some specifications are presented in Figure 6.4. Thus, the automata in Figure 6.4a and Figure 6.4b model the fact that all the cables connected to the station and the MMC arms need to be energized at least to the controllability threshold before the converter controller is deblocked. The automata in Figure 6.4c and Figure 6.4d model the fact that the ACCB is required to be closed after all the DCCBs of the station are closed so that the DC grid can be also charged during the start-up. The automaton in Figure 6.4e models the configuration of the MMC controller during the power ramp. Finally, the automaton in Figure 6.4f models the configuration of the MMC controller during the voltage restoration.




(a) $E^{st,M_2}_{1i}$, $i \in \{1,2\}$ (b) $E^{st,M_2}_{3}$

(c) $E^{st,M_2}_{4}$ (d) $E^{st,M_2}_{2i}$, $i \in \{1,2\}$

(e) $E^{st,M_4}_{1}$ (f) $E^{st,M_9}_{1}$

Figure 6.4 – Station specification models

| $E_{in}^{st,M_j} = \|_{l\in\mathbb{N}} E_{in,l}^{st,M_j}$ | No. of states | No. of transitions |
|---|---|---|
| $E_{in}^{st,M_2}$ | 72 | 245 |
| $E_{in}^{st,M_4}$ | 3 | 5 |
| $E_{in}^{st,M_9}$ | 3 | 4 |

Table 6.3 – Size of the specification models of the station $E_{in}^{st,M_j}$

The number of states and transitions of the obtained specification models of a station are summarized in Table 6.3 for each mode with an internal behavior.

### Controlled plant synthesis

From the uncontrolled plants and the specification models in each mode, we are able to build the controlled plants of the station. The construction of the models $H_{in}^{st,M_j}$ is described by Definition 4.4 (page 88). As explained above, the models $H_{in}^{st,M_5}$ and $H_{in}^{st,M_7}$ correspond to the uncontrolled plants in those modes, as they are uncontrollable to the supervisory control. The internal behavior of these modes is still indispensable in order for the ASC to be able to identify the mode commutation.

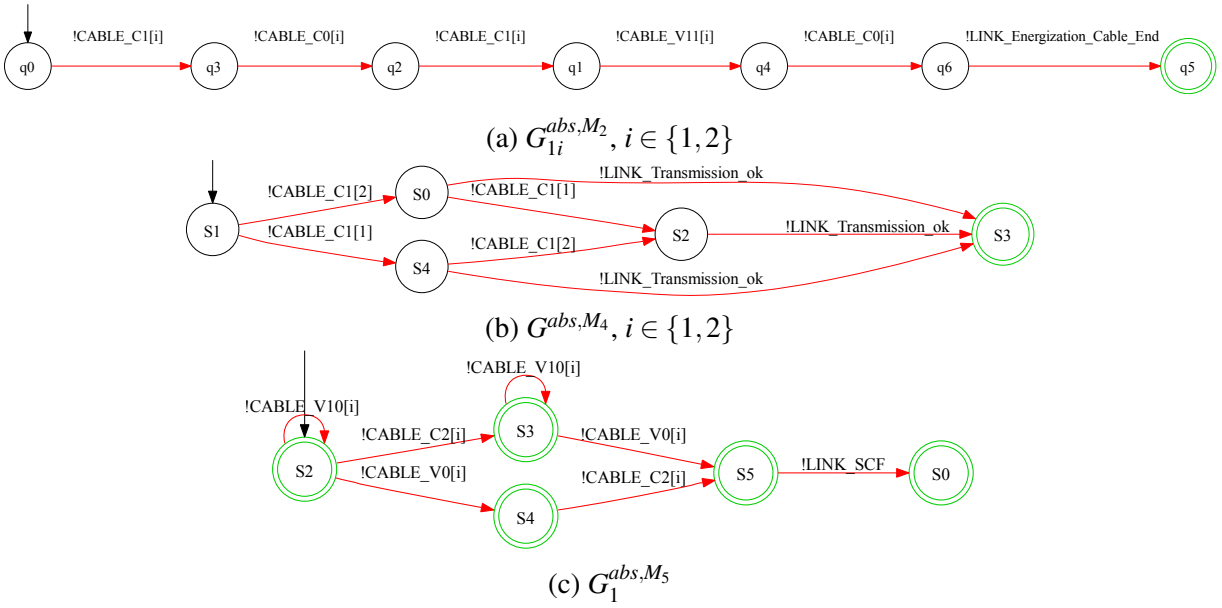
The size of the obtained models is summarized in Table 6.4.

| $H_{in}^{st,M_j}$ | No. of states | No. of transitions |
|---|---|---|
| $H_{in}^{st,M_2}$ | 99 | 169 |
| $H_{in}^{st,M_4}$ | 6 | 6 |
| $H_{in}^{st,M_5}$ | 36 | 84 |
| $H_{in}^{st,M_7}$ | 17 | 21 |
| $H_{in}^{st,M_9}$ | 15 | 22 |

Table 6.4 – Size of the controlled plants of the station $H_{in}^{st,M_j}$

### Abstraction of the controlled plant

The abstractable languages are then identified, according to Definition 4.5 (page 90). Once they are identified, the abstraction events are introduced in the controlled plant of the station by means of automata. For example, the automaton in Figure 6.5a introduces the event marking the end of the cables energization once both cables connected to the station have reached a steady state (the current stabilizes at zero after a charging phase) and their voltage reaches the nominal value. The automaton in Figure 6.5b introduces the event marking the effective power transmission once the station regulates the DC current to the desired value during $M_4$. As for




(a) $G_{1i}^{abs,M_2}$, $i \in \{1,2\}$

(b) $G^{abs,M_4}$, $i \in \{1,2\}$

(c) $G_1^{abs,M_5}$

Figure 6.5 – Automata introducing the abstraction events

the automaton in Figure 6.5c, it introduces the event marking the occurrence of a short-circuit fault, which is defined by a collapse of the DC voltage and the appearance of an overcurrent.

The abstractable languages are removed from the model as a result of the projection, as described in Section 4.3.2 (page 90). The size of the abstracted models is summarized in Table 6.5. In this thesis, because we consider the stations in the grid to be equivalent, the abstracted model of the stations in each mode is considered to be generic for all the stations.

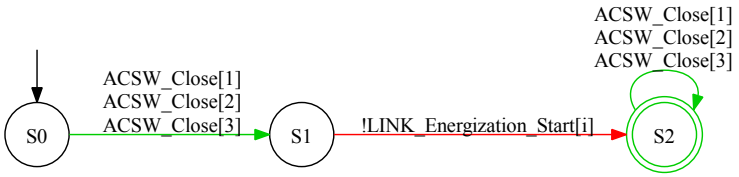
| $\tilde{H}_{in}^{st,M_j}$ | No. of states | No. of transitions |
|---|---|---|
| $\tilde{H}_{in}^{st,M_2}$ | 18 | 20 |
| $\tilde{H}_{in}^{st,M_4}$ | 4 | 4 |
| $\tilde{H}_{in}^{st,M_5}$ | 2 | 1 |
| $\tilde{H}_{in}^{st,M_7}$ | 4 | 5 |
| $\tilde{H}_{in}^{st,M_9}$ | 12 | 16 |

Table 6.5 – Size of the abstracted models $\tilde{H}_{in}^{st,M_j}$

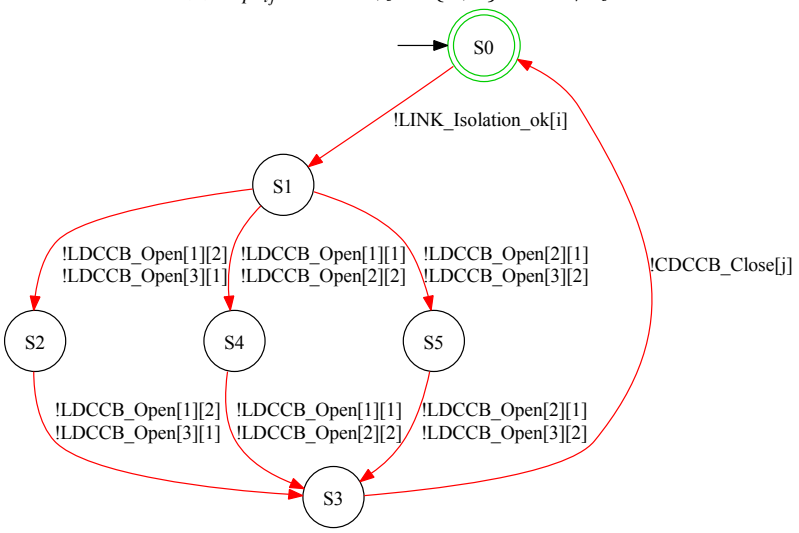
### 6.4.2 Controlled plant of the grid

#### Plant models construction

The abstracted model of the controlled plant of the stations is then instantiated as many times as there are stations in the grid and its events are renamed so as to reflect by means of an index to




(a) $G_{pc_{ij}}^{gr,M_2}$, with $i, j \in \{1,2\}$ and $i \neq j$




(b) $G_{pc_{ij}}^{gr,M_7}$, $i, j \in \{1,2\}$

Figure 6.6 – Physical constraint grid models

which station they correspond. Therefore, there do not exist shared events between the reduced models of the stations and so the only method that allows to further simplify the set of plant automata is the monolithic synthesis.

Before this, it might be necessary to model the physical constraints imposed by the interconnection of the different stations for each mode. Each individual physical constraint is modeled in the form of an automaton, denoted by $G_{pc_k}^{gr,M_j}$. As an example, the automaton in Figure 6.6a models the fact that a grid cannot start to be charged until one of the station in the MTDC system is connected to the adjacent AC grid, which provides the power necessary for the start-up. As for the automaton in Figure 6.6b, it models the three different cables where the fault can be identified by the protection algorithm in $M_7$. Typically, the identification of the faulty cable takes place after the opening of the CDCCBs and before their reclosing. The internal uncontrolled plant $G_{in}^{gr,M_j}$ of the grid in the mode $M_j$ is then defined to be $G_{in}^{gr,M_j} = (\|_{k\in\mathbb{N}}\, G_{pc_k}^{gr,M_j}) \parallel (\|_{i\in\mathbb{N}}\, \tilde{H}_{in,i}^{st,M_j})$ and the synthesis triple is modified such that $\mathcal{G}^{M_j} = \{G_{in}^{gr,M_j}\}$. The number of states and transitions of the obtained uncontrolled plants of the grid are summarized in Table 6.6.

### Specification models construction

In addition to the plant model, it is necessary to have a model of the specifications to perform a synthesis and obtain the controlled plant of the grid. The SCT allows to build this model of

| $G_{in}^{gr,M_j} = (\|_{k\in\mathbb{N}} G_{pc_k}^{gr,M_j}) \parallel (\|_{i\in\mathbb{N}} \tilde{H}_{in,i}^{st,M_j})$ | No. of states | No. of transitions |
|---|---|---|
| $G_{in}^{gr,M_2}$ | 1065 | 3129 |
| $G_{in}^{gr,M_4}$ | 34 | 93 |
| $G_{in}^{gr,M_5}$ | 8 | 12 |
| $G_{in}^{gr,M_7}$ | 22 | 36 |
| $G_{in}^{gr,M_9}$ | 88 | 216 |

Table 6.6 – Size of the uncontrolled plants of the grid $G_{in}^{gr,M_j}$

the specifications by the parallel composition of the models representing each individual specification. As an example, the automaton in Figure 6.7 models the fact that it is desired to connect all the stations to the MTDC grid before connecting to the adjacent AC grids. Otherwise, some cables and/or stations would remain uncharged during the start-up.

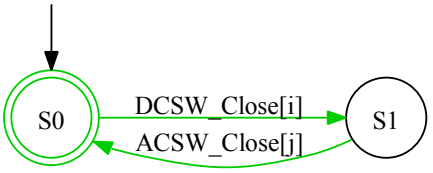



Figure 6.7 – $E_{in,ij}^{gr,M_2}$ $(i,j \in \{1,2\})$

The number of states and transitions of the obtained specification models $E_{in}^{gr,M_j} = \|_{l\in\mathbb{N}} E_{in,l}^{gr,M_j}$ are summarized in Table 6.7:

| $E_{in}^{gr,M_j} = \|_{l\in\mathbb{N}} E_{in,l}^{gr,M_j}$ | No. of states | No. of transitions |
|---|---|---|
| $E_{in}^{gr,M_2}$ | 826 | 2919 |
| $E_{in}^{gr,M_4}$ | 30 | 60 |
| $E_{in}^{gr,M_9}$ | 146 | 369 |

Table 6.7 – Size of the grid specification models $E_{in}^{gr,M_j}$

**Controlled plant synthesis**

Then, once the plant models and specifications during the considered procedure are obtained, we are able to build the internal controlled plant $H_{in}^{gr,M_j}$ of the grid for each mode $M_j$. At the end of this stage, the synthesis result for each mode is obtained by composing the controlled plant of the grid for a given mode with the controlled plants of each station. The size of the obtained models is given in Table 6.8.

| $H_{in}^{gr,M_j}$ | No. of states | No. of transitions |
|---|---|---|
| $H_{in}^{gr,M_2}$ | 138 | 198 |
| $H_{in}^{gr,M_4}$ | 25 | 45 |
| $H_{in}^{gr,M_5}$ | 8 | 12 |
| $H_{in}^{gr,M_7}$ | 22 | 36 |
| $H_{in}^{gr,M_9}$ | 44 | 60 |

Table 6.8 – Size of the controlled plants of the grid $H_{in}^{gr,M_j}$

### 6.4.3 Validation of the intramodal study

Firstly, the internal controlled plants $H_{in,i}^{st,M_j}$ and $H_{in}^{gr,M_j}$ exist, which proves that it is possible to find a trajectory in each mode that meets all the specifications. Secondly, these models have been validated in collaboration with the control and protection experts with respect to the description of the modes. Otherwise, unidentified errors at this stage would carry them to the following models. In this example, after studying all the possible strings of events in the models of each mode, the obtained models do not have any apparent error. In consequence, we can move onto the next step which concerns the study of the intermodal commutations.

## 6.5 Intermodal study

The intramodal study focuses only on the components used to represent the internal behavior of the system in each mode. The intermodal study, third step of the approach illustrated by the Figure 5.1 (page 110), concerns the extension of the controlled plant models by means of plant and specification automata so as to take into account the commutative behavior of the system as defined by the modes automaton. In the next section, all the necessary specification models for the extension of the controlled plants are given.

### 6.5.1 Extension of the controlled plants

We now consider in each mode the commutative behavior described by the modes automaton $G^{\mathcal{M}}$ shown in Figure 5.4. Because the intramodal specifications are built exclusively on the event sets generated by the components used in the modes, it is necessary to define intermodal specifications in order to couple both the internal alphabets $\Sigma_{in,i}^{st,M_j}$ and $\Sigma_{in}^{gr,M_j}$ of the controlled plants in a mode and the set $\Sigma^{\mathcal{M}}$ of commutation events of $G^{\mathcal{M}}$, as $\Sigma^{\mathcal{M}} \cap ((\cup_{i=1}^{n} \Sigma_{in,i}^{st,M_j}) \cup \Sigma_{in}^{gr,M_j}) = \emptyset$. In consequence, the intermodal specification, denoted by $E_{\rightleftarrows}^{M_j}$, representing the mode-switching dynamics is modeled in this step. This model allows to place the language of

the internal controlled plants in the corresponding mode of the modes automaton. Formally, the specification model $E_{\rightleftarrows}^{M_j}$ is built from the parallel composition of elementary specification automata $E_{\rightleftarrows}^{l,M_j}$, such that $E_{\rightleftarrows}^{M_j} = \|_{l\in\mathbb{N}}\, E_{\rightleftarrows}^{l,M_j}$.

The intermodal specification and plant models are given in Figure 6.8 for each mode $M_j \in \mathcal{M}$.

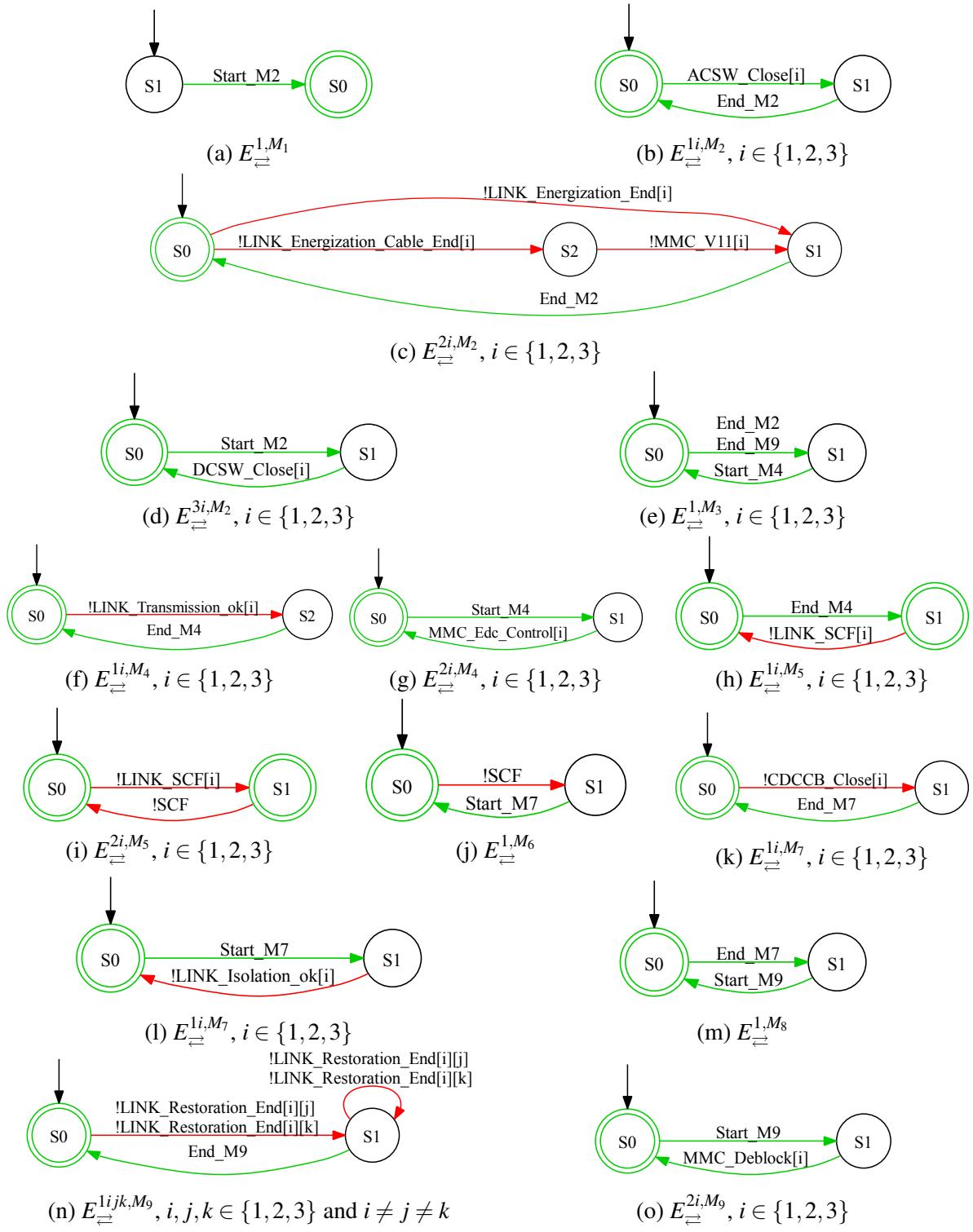



Figure 6.8 – Intermodal specification and plant models

Because the modes $M_1$, $M_3$, $M_6$ and $M_8$ do not have any internal behavior, only the commutative behavior of the mode is modeled by the respective specifications in Figure 6.8a, Figure 6.8e, Figure 6.8j and Figure 6.8m. That is, only those transitions labeled by an event in $\Sigma^{\mathcal{M}}$ that reaches or leaves the state representing the "empty" behavior of the mode are represented in the cited models. On the other hand, it is necessary for the rest of the modes to relate the events in the internal alphabet of the mode to those in the modes automaton. Thus, while the automata in figures 6.8d, 6.8g, 6.8h, 6.8l and 6.8o indicate that the first event of the internal controlled plants takes place after the commutative transition entering the mode has occurred, the automata in figures 6.8b, 6.8c, 6.8f, 6.8i, 6.8k and 6.8n model the fact that the commutation transition leaving the mode takes places once the internal controlled plant have reached their marked state, i.e. the internal sequence of events is completed, thus reaching the objective of the mode.

The abstracted plant of the station $\tilde{H}_{in,i}^{st,M_j}$ and the controlled plant of the grid $H_{in}^{gr,M_j}$ are then extended by synthesis to the models $H_{st,i}^{M_j}$ and $H_{gr}^{M_j}$, which consider the commutative behavior of the system, according to Definition 5.2. The size of the resulting models is given in Table 6.9. The automata obtained for the modes $M_1$, $M_3$, $M_6$ and $M_8$ correspond to $G^{\mathcal{M}}$, as they do not have an internal behavior and only the commutative behavior is represented.

| $H^{M_j}$ | No. of states | No. of transitions |
|---|---|---|
| $H_{st,i}^{M_1}$ | 9 | 9 |
| $H_{gr}^{M_1}$ | 9 | 9 |
| $H_{st,i}^{M_2}$ | 26 | 79 |
| $H_{gr}^{M_2}$ | 146 | 207 |
| $H_{st,i}^{M_3}$ | 9 | 9 |
| $H_{gr}^{M_3}$ | 9 | 9 |
| $H_{st,i}^{M_4}$ | 14 | 14 |
| $H_{gr}^{M_4}$ | 35 | 55 |
| $H_{st,i}^{M_5}$ | 4 | 3 |
| $H_{gr}^{M_5}$ | 10 | 14 |
| $H_{st,i}^{M_6}$ | 9 | 9 |
| $H_{gr}^{M_6}$ | 9 | 9 |
| $H_{st,i}^{M_7}$ | 15 | 16 |
| $H_{gr}^{M_7}$ | 33 | 47 |
| $H_{st,i}^{M_8}$ | 2 | 36 |
| $H_{gr}^{M_8}$ | 2 | 88 |
| $H_{st,i}^{M_9}$ | 20 | 22 |
| $H_{gr}^{M_9}$ | 55 | 71 |

Table 6.9 – Size of the extended controlled plants at the station and grid levels

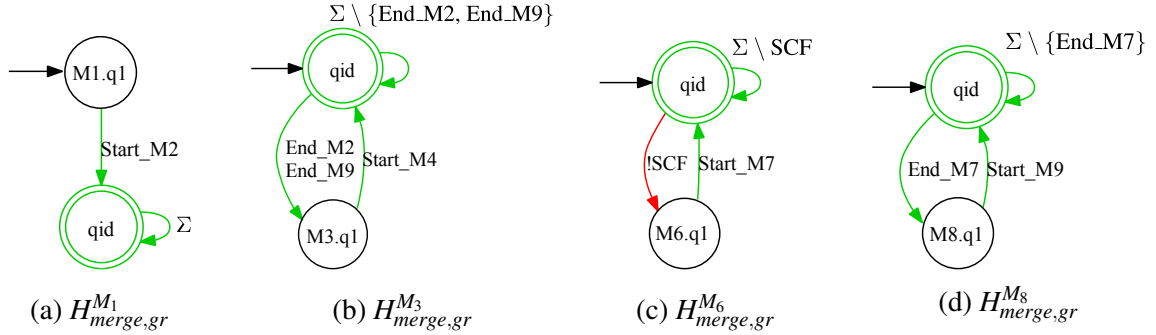
### 6.5.2 Validation of the intermodal study

After extending the controlled plants of the grid and the stations during the intermodal study, we have converged to a point where it is no longer necessary to modify the models because their behavior has been validated as they achieve the desired objectives in each considered mode. We are therefore able to ensure that switching from one mode to another is reliable and is consistent with the specifications and the internal behavior.

## 6.6 Merging of the non-significant states

An large part of the controlled plants behavior has been generated only to validate the commutations between the different models. However, the behaviors that are necessary for the supervisory control to work correctly include only the internal and the commutative behavior, which represent the behavior of the system in a mode. Therefore, the last step of the design method concerns the merging of those states that are non-significant for the supervisory control, which become a state where the mode is inactive. In consequence, all outgoing transitions in this state will activate the mode, and any transitions entering this state will disable it. All the events contained in $\Sigma = (\cup_{i=1}^{n} \Sigma^{C_i}) \bigcup \Sigma^{\mathcal{M}}$ that do not provoke an activation of the mode are added in a self-loop transition outgoing from the inactive state, which increases the number of transitions.

The controlled plants of the grid resulting from the merging of the non-significant states in each mode are given in figures 6.9, 6.10 and 6.11, except for $M_2$, $M_4$ and $M_9$ because of the large size of the models. To conclude, the global equivalent behavior given by the automaton $H$ resulting from the parallel composition of all the merged models, such that $H = (\|_{i,j\in\mathbb{N}} H^{M_j}_{merge,st,i}) \| (\|_{j\in\mathbb{N}} H^{M_j}_{merge,gr})$, is calculated. The global behavior described by the obtained model respects the conditions given in Definition 5.1 and corresponds to the desired discrete-event dynamics for the considered case study. Thus, the supervisory control is validated for its implementation. Table 6.10 summarizes the size for each mode of the model representing the behavior of the system in it.




(a) $H^{M_1}_{merge,gr}$ (b) $H^{M_3}_{merge,gr}$ (c) $H^{M_6}_{merge,gr}$ (d) $H^{M_8}_{merge,gr}$

Figure 6.9 – Merged models of the modes with no internal behavior

| Automaton | No. of states | No. of transitions | Automaton | No. of states | No. of transitions |
|---|---|---|---|---|---|
| $H^{M_1}_{merge,st}$ | 2 | 35 | $H^{M_6}_{merge,st}$ | 2 | 36 |
| $H^{M_1}_{merge,gr}$ | 2 | 87 | $H^{M_6}_{merge,gr}$ | 2 | 88 |
| $H^{M_2}_{merge,st}$ | 21 | 58 | $H^{M_7}_{merge,st}$ | 5 | 41 |
| $H^{M_2}_{merge,gr}$ | 137 | 286 | $H^{M_7}_{merge,gr}$ | 23 | 124 |
| $H^{M_3}_{merge,st}$ | 2 | 36 | $H^{M_8}_{merge,st}$ | 2 | 36 |
| $H^{M_3}_{merge,gr}$ | 2 | 88 | $H^{M_8}_{merge,gr}$ | 2 | 88 |
| $H^{M_4}_{merge,st}$ | 5 | 39 | $H^{M_9}_{merge,st}$ | 9 | 47 |
| $H^{M_4}_{merge,gr}$ | 26 | 133 | $H^{M_9}_{merge,gr}$ | 45 | 148 |
| $H^{M_5}_{merge,st}$ | 3 | 37 | $H$ | 289 | 459 |
| $H^{M_5}_{merge,gr}$ | 9 | 100 | | | |

Table 6.10 – Size of the merged models and their parallel composition

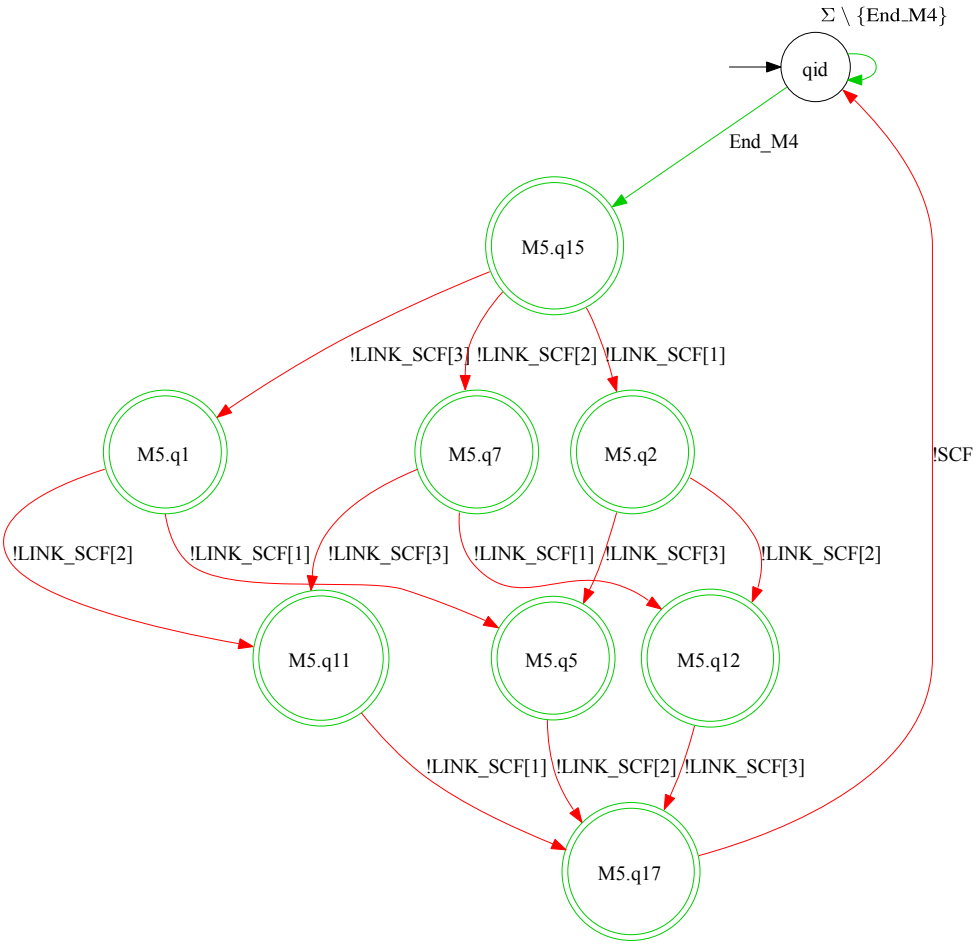



Figure 6.10 – $H^{M_5}_{merge,gr}$


Figure 6.11 – $H^{M_7}_{merge,gr}$

## 6.7 Simulation in a virtual mock-up

The three terminal MTDC grid represented in Figure 6.1 is modeled in the electromagnetic transient simulation software EMTP-RV thanks to its dedicated library of components. As the scope of this thesis is to supervise the MMC as a entire component and not each SM in its arms, the MMC model used for the simulation corresponds to a reduced-order averaged arm model, as presented in [Zam17]. Furthermore, according to [Zam17], this model replicates an averaged behavior of the arms SMs during a DC short-circuit fault. The MMC controllers [Shi17] and the protection algorithms associated to the protection relays [Lou17] have been also integrated. The parameters of the simulation model, given in Appendix E, correspond mostly to those of the INELFE (INterconexión Eléctrica Francia-España or Electricity Interconnection France-Spain) HVDC link between France and Spain and can be found in [Den13; Lou17; Shi17]. Then, following the implementation method proposed in sections 4.4 and 5.6, the merged grid automata of each mode have been implemented within the *Supervision* function of the proposed C code supervisory control program, while the merged station automata of each mode have been implemented within the *Logic Control* function. The values of the different thresholds that relate the digital signals corresponding to the component's voltage and current measurements in EMTP-RV with the response variables in the *Interface* function have been determined according to the regions of operation in Figure 4.4 and Figure 4.5 and the simulation parameters in Appendix E.

At this point, the source code file containing the supervisory control program is integrated in the simulation software EMTP-RV by means of a DLL, which has been built in Visual Studio with the specific set of files provided in EMTP-RV for the building of DLLs. The custom C code containing the supervisory control program can then be called from the mentioned files via a header file that we have previously written. Then, the digital signals associated to the components in the software are manually linked with the corresponding supervisory control's input or output and the DLL is built. The obtained DLL can finally be implemented in EMTP-RV for simulation purposes. For this, the simulation model has been thoroughly adapted so as to integrate in an homogeneous manner the supervisory control, the system measurements, the continuous-time controllers and the protection algorithms. Because the two poles of the MTDC grid are identical (except for the DC voltage polarity) and are independently operated, the supervisory control is required to be implemented two times (one supervisory control per pole). For this purpose, two copies of the same DLL can be used. Indeed, the internal variables of each copy remain local to the DLL as the inner functions have been coded with pointers in this PhD thesis. Otherwise, the use of global variables would oblige to name differently the variables in each copy, which would be very time consuming. Each of the implemented DLLs counts 45 inputs and 35 outputs.

An offline simulation is then performed so as to validate the intramodal and intermodal behavior of the implemented supervisory control, which is executed at each time-step (fixed at 10

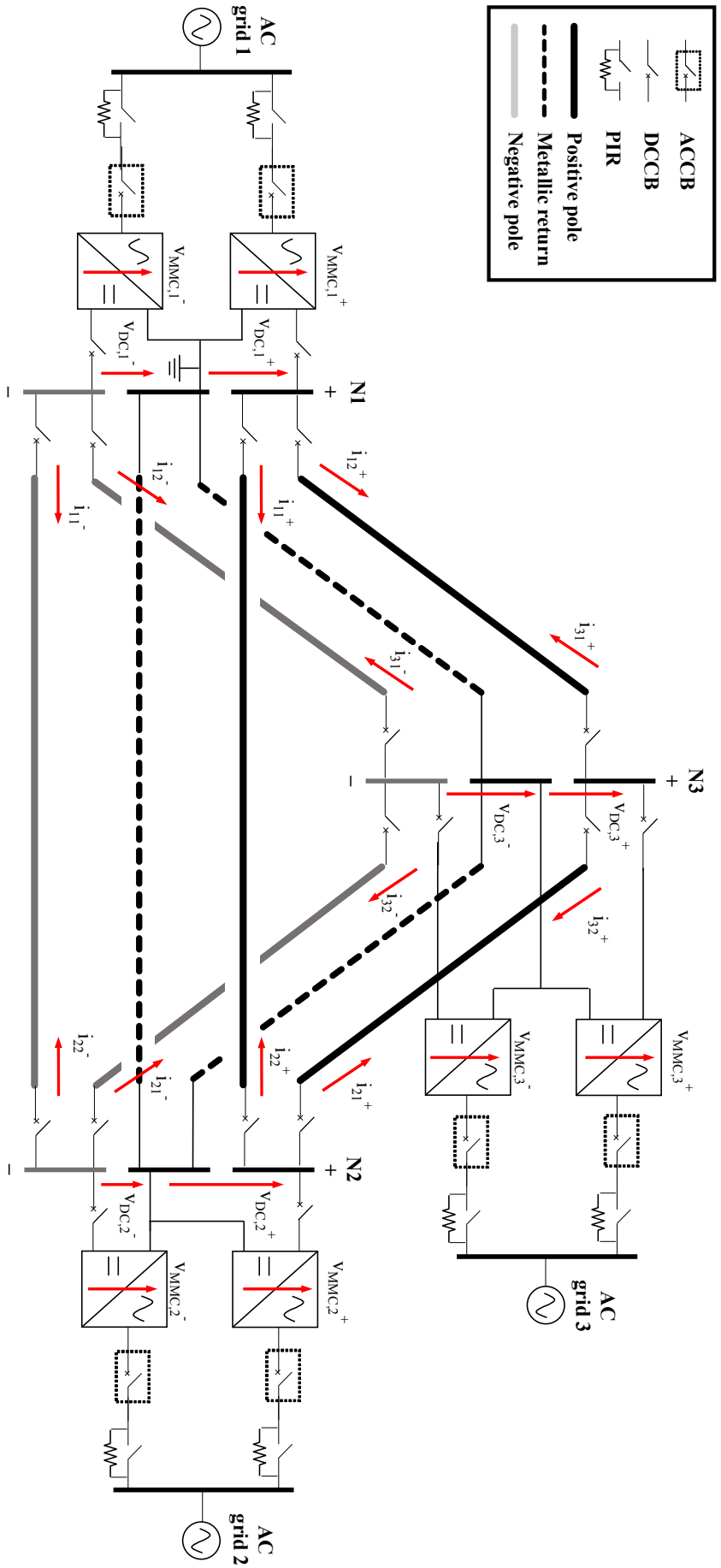



Figure 6.12 – Voltage in the MMCs

µs). Given the degree of detail of the EMTP-RV dedicated component models and the impossibility to test the obtained controllers in a real system, the performed simulation is considered as a valid method for the validation of the proposed ASC. Also, in order to show the independence of operation of the two poles of the MTDC grid, a short-circuit fault is introduced only in the positive pole, while the negative pole remains healthy during the whole simulation. In consequence, the supervisory control related to the positive pole will follow the mode-switching dynamics described by the modes automaton (cf. Figure 5.4), while the one related to the negative pole will only be concerned by $M_1$, $M_2$, $M_3$, $M_4$ and $M_5$, given that the event SCF never occurs. The obtained simulation results are shown in figures 6.13, 6.14, 6.15 and 6.16 for the two poles of the grid. Furthermore, the mode of the system at each time instant is indicated in the figures, except for $M_1$, $M_3$, $M_6$ and $M_8$ given that they do not have an internal behavior and so the time interval during which the system is in these modes is very short to be appreciated.

The measured variables are represented in Figure 6.12. For each cable, the DC current is measured at each of its ends. Thus, the amplitude of the current measured at each end is identical, however, the direction varies. If the current is transferred through the station from the DC grid to the AC grid, the current will be positive. In opposition, if the current is transferred through the station from the AC grid to the DC grid, the measured current will be negative. The DC voltage is measured for each station as the difference of potential between the busbar and the ground. Thus, the DC voltage at the negative pole will be negative, as opposed to the one in the positive pole. Finally, the MMC voltage is calculated as being the average of the voltages stored in the MMC arms, which are calculated by adding all the SM voltages in the same arm. Because the voltage in a SM is measured as the difference of potential across the SM capacitor (which cannot be negative), the MMC voltage is always positive, independently of the pole.

The voltage within the MMCs of both poles is shown in Figure 6.13. Firstly, the supervisory control enforces the corresponding commands during the start-up so as to bring the voltage in the MMCs to its rated value. During the controlled charging stage, the MMCs are charged one after another in order to minimize the fluctuations of the DC voltage (see Figure 6.14 from 1.1s to 1.25s). Then, as it can be observed, the voltage level in the MMCs remains constant during the whole simulation for the two poles as a result of the autonomous protection of the MMCs and the action of the supervisory control. Indeed, the latter effectively modifies the configuration of the MMCs controllers so as to keep the MMC within a controllable region of operation in order to restore the power balance in each pole when necessary.

Figure 6.14 shows the DC bus voltage evolution during the MTDC grid's operation. At the beginning of the simulation, both poles are in $M_1$, although the start-up mode $M_2$ is immediately activated by the supervisory control. The reason why the start-up charging only begins at 0.05s is due to the delay associated to the closing of the CBs, which is of 10 ms for the DCCBs and 40 ms for the ACCB. During the start-up, the ASC enforces in the grid the necessary commands according to the considered strategy (cf. Section 6.3.1) and the evolution of the grid components

until all the MMCs and cables are charged to the rated voltage, which occurs at around 1.25s. Hence, the mode $M_3$ is reached at this stage. Immediately after, the supervisory control activates the mode $M_4$ so as to inject the desired power in the grid. Once the DC currents (Figure 6.15) reach the desired value, the marked mode $M_5$ corresponding to the nominal operation of the grid is activated and the system remains in this mode as long as no short-circuit fault occurs. At this point, the simulation results differ for the two poles. While the negative pole remains in $M_5$ for the rest of the simulation, a short-circuit fault is introduced in the positive pole at 2s. As the grid is forced to enter $M_6$, the protection algorithms within the simulation model autonomously respond by executing the protection strategy described by the models integrated in the mode $M_7$ of the supervisory control. As the ASC is able to follow the execution of the protection strategy, it is able to rapidly activate the voltage restoration mode $M_9$ at the end of it ($\approx$ 2.1s), i.e. once the system enters $M_8$. Finally, the system is brought back again to the marked mode $M_5$ by the supervisory control once the power flow is restored to the desired value in $M_4$.

In Figure 6.15, the DC current flowing through each cable of the two poles is shown. From the simulation results it can be appreciated that the DC current naturally flows within the grid during the start-up (from 0s to 1.25s) until the grid reaches a steady-state voltage and all the MMCs and cables are charged. Furthermore, the uncontrolled charging (from 0s to 1.1s) can be clearly distinguished from the controlled charging stage (from 1.1s to 1.25s) by the DC currents allure. In Figure 6.15b the effect of the activation of $M_4$ by the supervisory control can be clearly seen as the DC currents are modified until the desired value is reached. Then, the short-circuit fault in the positive pole occurs at 2s. While the impact in the negative pole is negligible, the DC current in the positive pole cables rises abruptly to reach a range of 5 to 10 p.u. The protection algorithms in each station detect locally the fault once the current reaches 2 p.u. and command the CDCCBs to open so as to extinguish the DC current. Then, the faulty cable is correctly identified and the associated LDCCBs are opened. In Figure 6.15a, the faulty cable (Cable 31/Cable 12) can be clearly recognized given that the measurements at both ends of the cable have the same direction, as the DC currents head towards the fault from the AC grid. This is not the case for the healthy cables. Then, because the supervisory control is able to follow the internal behavior of $M_7$, it configures automatically the MMC controllers to restore the DC voltage and currents so as to bring back the system to $M_5$ immediately after the faulty cable is isolated at around 2.1s, except for the Cable 13, which is out of operation.

Finally, the DC power transmitted by each station is shown in Figure 6.16 for the two poles. As the DC power is the product of the DC voltage and the DC current, the power evolution follows the same evolution as the DC current. It should be noted, however, that while the DC currents in the negative pole are slightly affected by the voltage restoration of the positive pole at around 2.2s in Figure 6.15b, the DC power transmitted through the negative pole remains constant throughout the whole short-circuit fault. This is due to the fact that the MMC controller regulates the DC power entering and leaving the station and not how this power is distributed

in each cable. In consequence, while a perturbation in the AC grids due to the restoration of one pole does not impact the power transmitted through the other pole, it might impact how this power is distributed through the cables. However, the DC current in the cables remains within the non-zero and secure region of operation and so this slight disturbance does not have a major impact on the operation of the negative pole, as the ASC does never deactivate $M_5$.

In consequence, the performance of the automated supervisory control developed for the considered case study has proven to be satisfactory, as it has effectively managed the transitions between the modes of the system during the simulation of the grid operation and coordinated the component actions according to the internal behavior of each mode. The supervisory control's performance has been equally tested for a random sequence of external demands and the illegal transitions were correctly prohibited. Thus, the required specifications were respected.

## 6.8 Conclusion

In this chapter it has been proven that it is possible to design and implement a valid SCT-based supervisory control for a realistic operation cycle of an MTDC grid, which includes the start-up of the system, the injection of power within the grid, the maintenance of the power balance, the protection in case of a short-circuit fault and the subsequent restoring of the power balance in the grid. The difference between the objectives to be reached at each mode has facilitated the modeling of the specifications internal to each of them. Furthermore, the use of a compositional approach has allowed to minimize the information transmitted between the station and the grid levels to the essential, thus simplifying the resulting models without compromising the level of detail of the supervisory control. The respect of the specifications representing the objectives internal to each mode has been validated. Then, the internal models have been extended so as to take into account the commutative behavior described by the modes automaton. The controllability and consistency of the extended models has been validated. Next, the size of the extended models has been reduced so as to keep only the significant behavior in each mode. The merged models represent the expected and validated system behaviors and include all the admissible incoming and outgoing commutations within each mode. Finally, the models built in the Supremica software have been exported and manipulated in order to generate a C source code file according to the implementation method presented in this PhD thesis. The resulting code has been integrated within a simulation model in the EMTP-RV software. The performance of the implemented supervisory control has proven to be satisfactory and all the considered internal and commutative behaviors have been respected for each mode.

To conclude, the applicability of the design and implementation method of an automated supervisory control for HVDC systems proposed in this PhD thesis has been validated. Because of the use of SCT-based approaches, the designed models are safe by construction, i.e. they comply with all the specifications described, be they safety specifications, functional or related to

mode-switching. Furthermore, the models properties are not modified during the implementation. Thus, because the adequacy of the supervisory control largely depends on the construction of the component and specification models, particular attention must be paid by the designer and the system experts during the different validation steps.

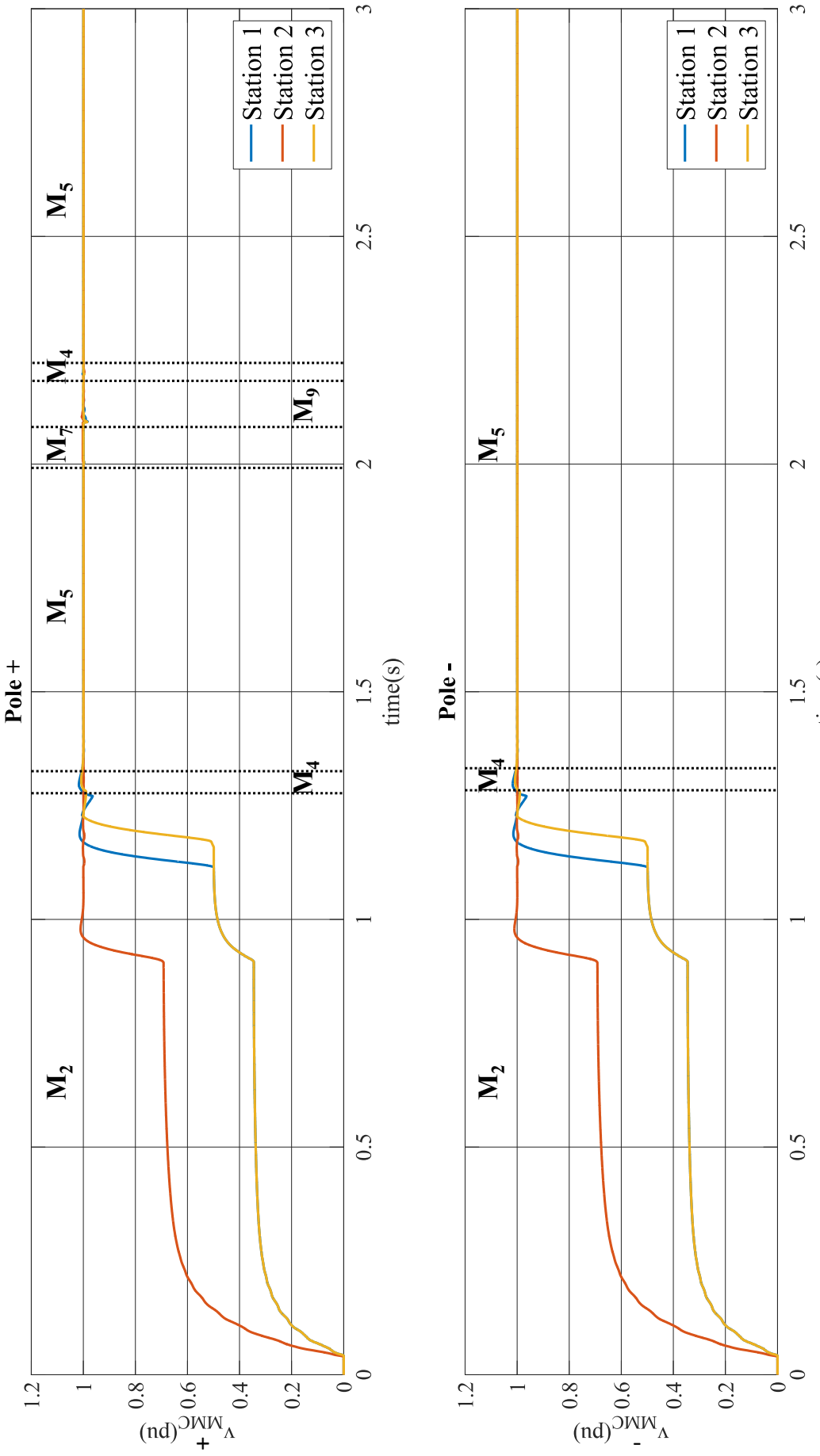



Figure 6.13 – Voltage in the MMCs

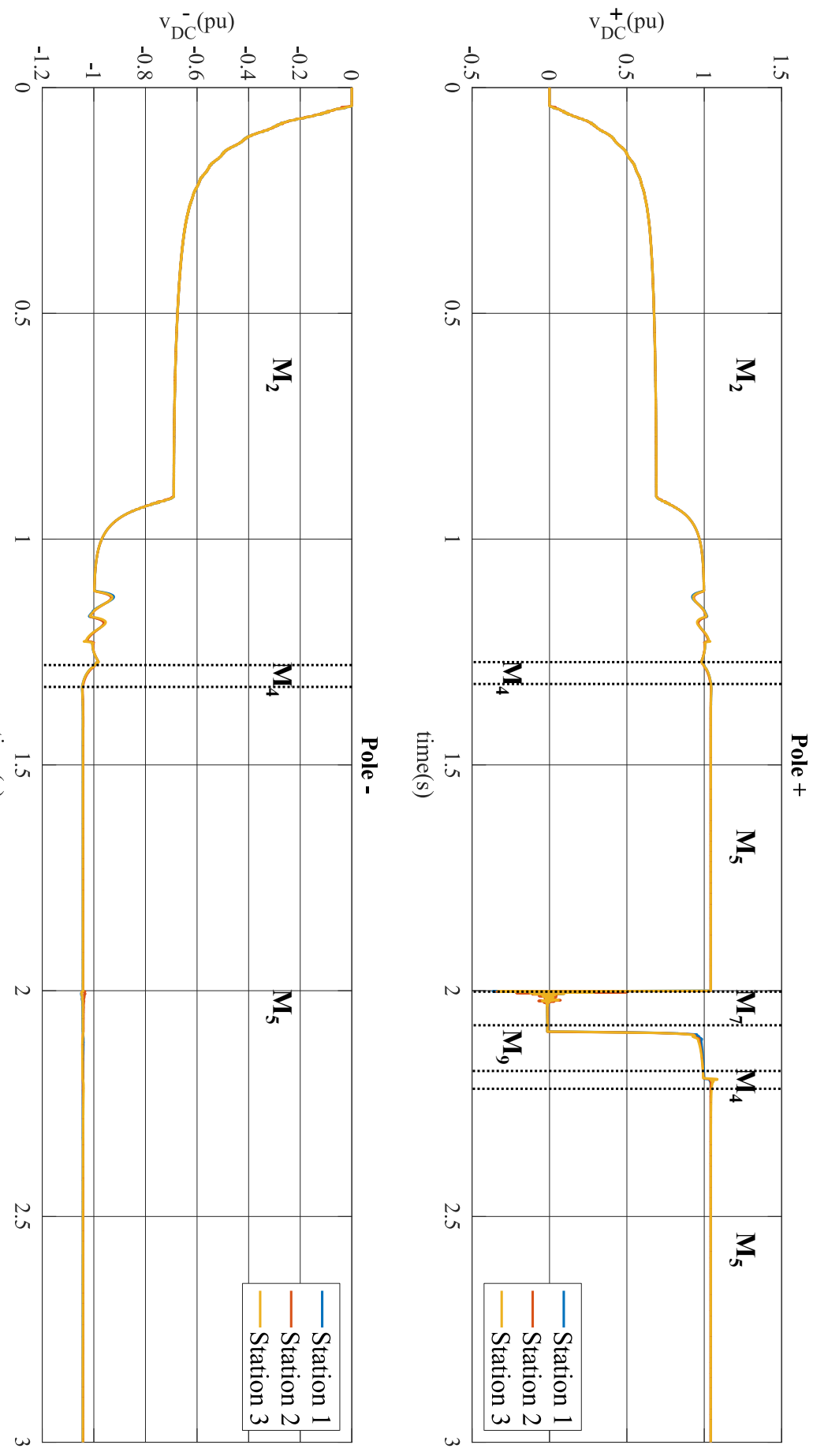
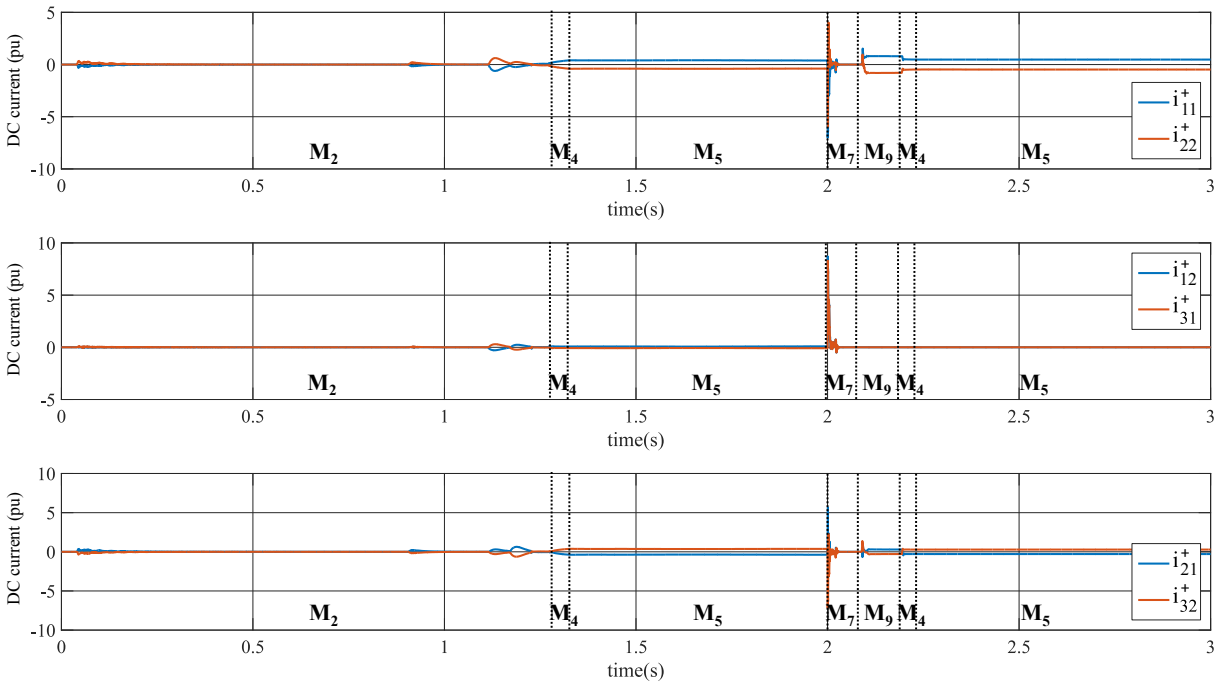
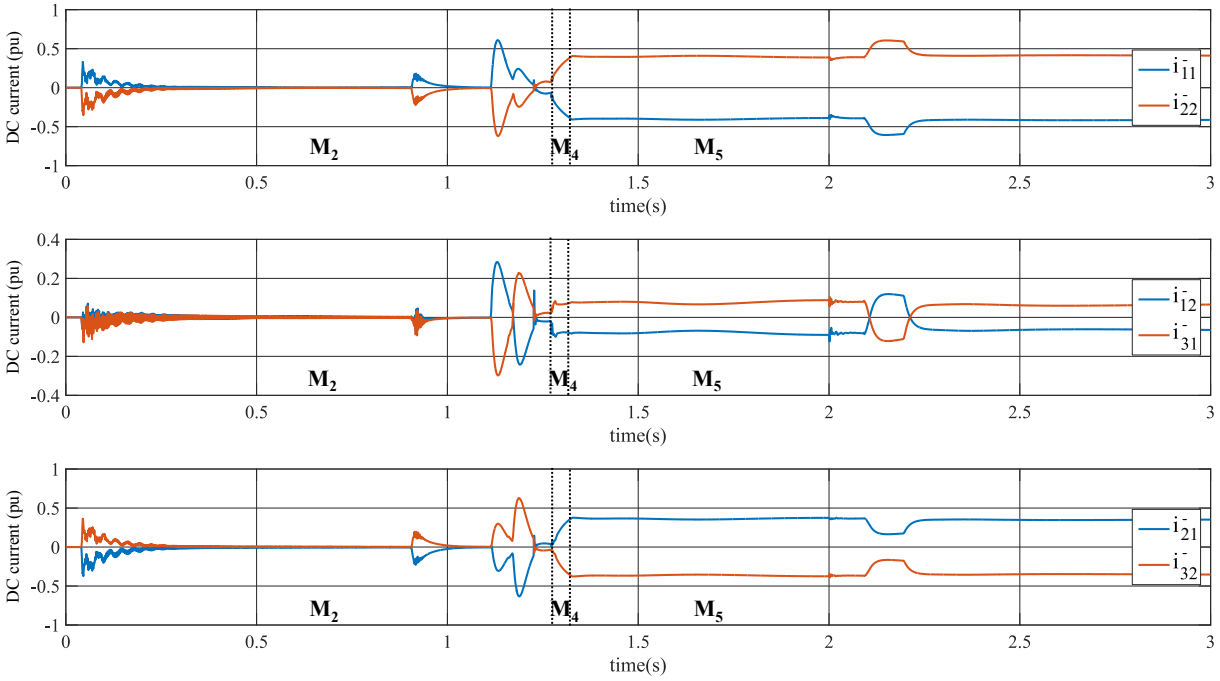



Figure 6.14 – DC bus voltage




(a) Positive pole




(b) Negative pole

Figure 6.15 – DC current in the cables

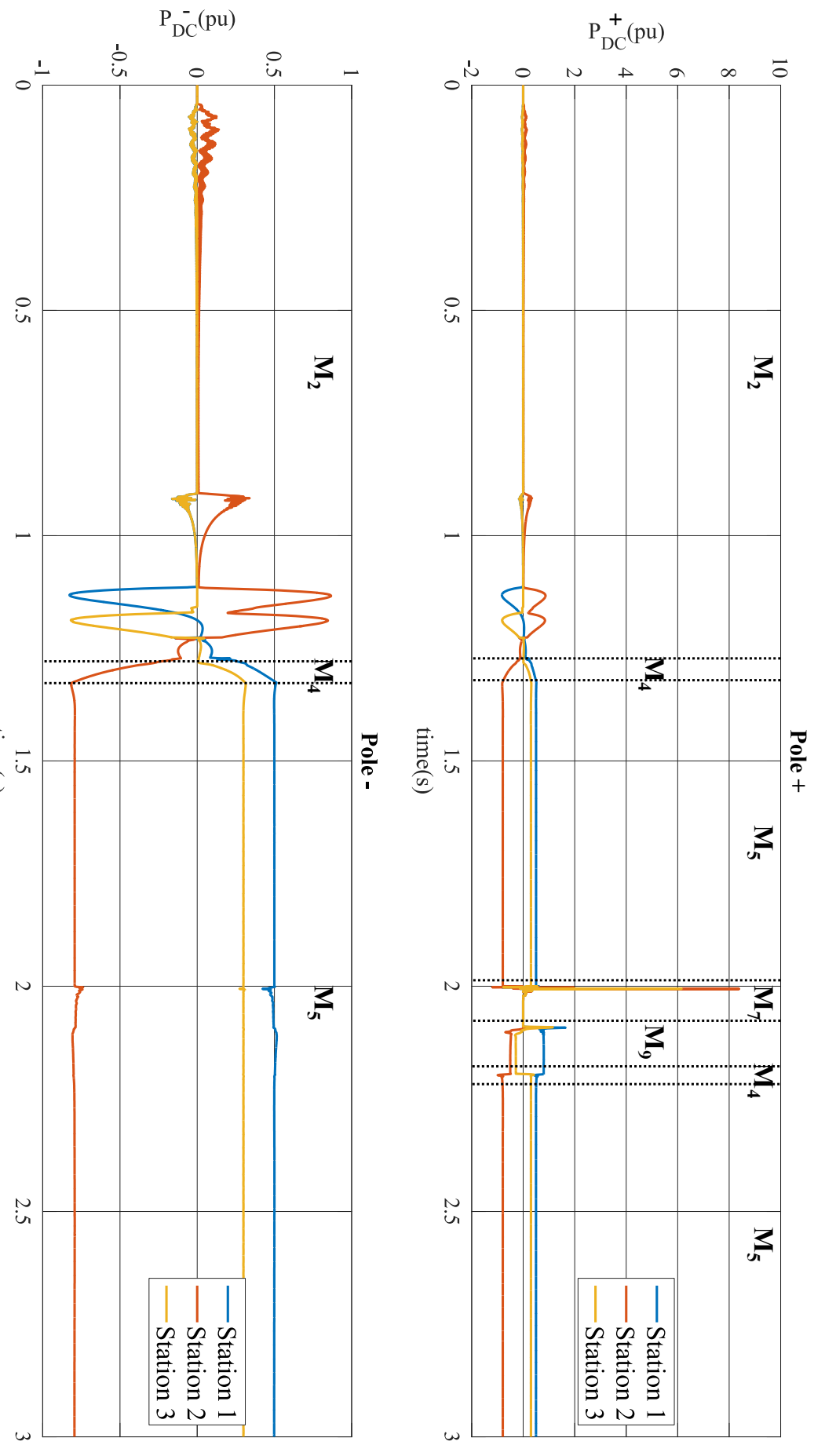



Figure 6.16 – DC bus power

# 7

# General conclusion

## Contents

## 7.1 Conclusions

In response to the continued increase in global energy consumption and the growing concerns on sustainable development, Renewable Energy Sources (RES), such as wind, sun, biomass, geothermal energy, etc. are considered to be promising alternatives to traditional energy sources and should replace conventional fossil fuel power plants in the coming decades. Nevertheless, due to their intermittent nature and their location, which is likely to be at a great distance from the consumption centers, the integration of such sources into the electrical grids has already put unprecedented stress on existing power transmission systems based on traditional Alternating Current (AC) technology. Therefore, a fundamental upgrade of the existing AC transmission system using the High-Voltage Direct Current (HVDC) technology is considered to be the most promising and technically feasible solution for the integration of RES into the grid. In line with the growing number of applications, the concept of connecting multiple HVDC stations to form a Multi-Terminal DC (MTDC) grid has emerged. It is expected to provide more flexibility and reliability in the power transmission by serving as an additional bulk power grid overlaid to the AC transmission system.

However, because of the large number of power electronics devices in an HVDC system and their fast dynamics, disturbances quickly propagate within an MTDC grid, which can eventually lead to a blackout of the grid and affect the entire power transmission system. Furthermore, a certain coordination between the components of the HVDC system is needed as negative interactions may arise between the multiple protection and control schemes triggered by a disturbance. In HVDC systems, however, human intervention is greatly limited during the real-time operation of the grid, as opposed to AC systems, by cause of the short-time range in which the different control and protection actions should be performed. In consequence, the development of an Automated Supervisory Control (ASC) that replaces the human operator for real-time monitoring and component coordination tasks is essential if an acceptable grid operation is to be expected. Such a supervisory control needs to be reactive to the grid's evolution so as to respect the required specifications in an automated manner, while the decision-making from the human operator is minimized as much as possible.

Thus, the major contributions of this thesis is to propose a formal methodology for the design and implementation of supervisory control that manages both the mode-switching dynamics of the system and the components' behavior in each mode. Given that on the one hand, such management is event-driven, and on the other hand, the physical evolution of the components in an HVDC system is determined by the change in the digital signals that command them, the proposed method is based on Discrete Event Systems (DES) modeling. Furthermore, given that a mode is a particular configuration of the system where a set of components must meet a set of specifications, the design method proposed in this thesis is based on formal approaches within the Supervisory Control Theory (SCT) framework, which constitute a powerful tool for

the construction of safe models and avoid steps of verification of the specifications afterwards on the built models.

Although the use of DES-based techniques has already been proposed in the literature to the operation of AC power systems, ultimately, the lack of a coherent and unambiguous control architecture that takes into account a realistic behavior of the system has made it difficult to apply the proposed solutions outside the considered case studies. Therefore, the formal approaches used in this thesis have been selected so as to adapt the ASC architecture to the nature of power transmission systems, which are geographically dispersed (converter stations located throughout the grid), hierarchically operated (local protection and control schemes interacting with a control center) and experiencing diverse situations during their operation (fault treatment, power balance restoration, etc.).

In consequence, the proposed design method merges vertical and modal decomposition approaches. Concerning the vertical decomposition, the compositional approach [Moh14] is retained as it allows to define several levels of control during the design phase and is not limited by a predefined structure. Thus, the flexible design eases the integration of new aspects (components, modes, specifications...) with the minimum impact on the existing structure, which is particularly interesting for MTDC systems given that these systems are not yet fully developed. As for the modal decomposition, this thesis follows the previous work concerning mode management [Far10], which resulted in a control architecture of concurrent models, with one model in each mode representing both the internal and commutative behavior of the latter, such that the uniqueness of active mode is respected and no contradiction exists in the control law produced.

Prior to the design of the supervisory control, an exhaustive methodology for the identification of the functional and monitoring requirements of an HVDC system, from the component level to the grid level, has been proposed. This methodology permits to collect the necessary information for the construction of a set of models that describe the components' generic behavior, and the identification of the mode-switching dynamics of the grid modeled by the modes automaton. In addition, the functional and monitoring analysis has been conceived as a common framework between the ASC designer and the HVDC experts so that the models built are consistent with the physical behavior of the system and the expectations from it.

The design of the supervisory control has then been decomposed into four distinct steps.

The first step deals with the modeling of the components in a station for any given mode, such that their behavior is considered without restrictions, that is, in its generic form. These generic model can then be later instantiated and adapted to the particular behavior of the component inside a particular mode. Similarly, the modes automaton is constructed in this step, from the results obtained during the functional and monitoring analysis. Contrarily to the work presented in [Far10], however, the commutation events in an HVDC system are not generated by a component alone, but are rather dependent on the reaching of a certain state by the whole

set of components in the system. In consequence, a set of conditions to be respected is given so as to verify that the modes automaton is consistent with the component models in each mode.

The second step, the intramodal study, concerns the construction and validation of the behavior internal to each mode, both at the station and the grid level, with respect to the operational requirements and regardless of the features of the other modes by means of the compositional approach [Moh14]. In this step, particular attention should be paid when modeling the specifications and physical constraints under the form of formal models. At first, a model of the controlled behavior of a station is then synthesized from the specifications and physical constraints. If possible, the resulting automaton is abstracted so as to keep only the essential information for its subsequent manipulation at the grid level. The global plant of the grid is then constructed by composition of the station models, for a grid of any size, and the model describing the controlled behavior of the grid is obtained by synthesis. Thus, this step is completed once the controlled behavior is validated at all the operation levels, that is, the station, which coordinates the components, and the grid, which coordinates the stations.

The third step, the intermodal study, extends the models validated in the second step by means of newly defined specifications in order to take into account the commutative behavior referred by the modes automaton while ensuring their controllability.

The fourth step then merges the non-significant states of the extended models into an inactive state, so that only the states associated to the considered mode are preserved. This state represents the mode inactivity and provides concurrent operation between models. In opposition to the work presented in [Far10], each transition activating a mode reaches a unique state (as the commutation events are not related to a single component) and so no information is lost during the mode commutation. Hence, it is not necessary to track the trajectory followed during the switching between the modes. Finally, conditions are given to verify that the global behavior obtained via the concurrent functioning of the models in each mode is controllable and non-blocking.

Once the supervisory control obtained from the design process is validated with the Supremica sofware, a method that implements the resulting models into different levels of control, based on the work presented in [Vie17], is proposed. Our method takes into account the hybrid nature of power transmission systems, thus integrating an interface that relates the continuous-time behavior of the physical components to their discrete-event model. Moreover, the method is conceived through generic expressions that can be later implemented in C code, thus allowing a highly reactive and customized control for the real-time operation of HVDC systems.

As for the practical aspects of the thesis, the proposed design and implementation method has been first illustrated with a case study considering the start-up of a point-to-point link [Rom17]. Moreover, a software tool that largely automates the proposed implementation method has been developed. This tool manipulates the models built in the Supremica software and generates as output a C source code file that can be implemented in the EMTP-RV software. The applica-

bility of the proposed design and implementation methodology has then been validated for a realistic operation cycle of a 3-terminal MTDC grid and the resulting simulation has proven to be satisfactory.

To conclude, three prerequisites are invariably needed if an ASC is to be developed: it is necessary to define the mode-switching dynamics of the system; it is necessary to identify the functions of each component and their behavior in each mode; and finally, for industrial issues, the supervisory control needs to be designed and implemented, preferably by means of a formal method that results in a supervisory control which respects the system specifications by construction. Thus, a generic methodology for the analysis of the functional and monitoring requirements of the system is provided so as to guarantee the coherence of the ASC with the physical system. We have also adopted a clear and definite approach to formally represent, methodically design and systematically verify the modes' internal and commutative behaviors so as to respect the operating specifications of the system during all the operating conditions at the levels of control corresponding to the configuration of the system. The modal and vertical decomposition of the system, along with the multi-model approach, makes it possible to limit the combinatorial explosion. Furthermore, the use of a formal method has allowed to verify that the global behavior of the resulting supervisory control is non-blocking and controllable, such that all the models are safe by construction and consistent with each other. Also, an implementation method that eases the test and validation of the designed supervisory control during a realistic simulation of an MTDC grid operation in a dedicated software has been proposed. In addition, the context in which the thesis was conducted (research program P1 within SuperGrid Institute) has allowed to work closely with power system experts, which has originated many exchanges between the two research fields, and we were able to convince these experts of the usefulness of formalizing their work for supervision purposes by DES and SCT-based approaches. Finally, the main points that needed to be adressed during this thesis were the modeling of the continuous-time behavior of the components as discrete-event dynamics, the integration of the resulting supervisory control within the control architecture of the grid and its implementation in a continuous-time simulation environment.

## 7.2 Perspectives

A perspective to our work would be to extend the modes considered during the intermodal study so as to cover the complete set of operating scenarios resulting from the monitoring analysis. Similarly, the number of mode commutations that deactivate the modes could be extended, and the case where the corresponding transitions are generated from any given state of the internal models could be considered. Another interesting perspective would be to establish a hierarchical partition of the modes, such that the component modes are differentiated from the station modes and the grid modes, to such a degree that the different levels of operation would

have different modes of operation. In this manner, the mode of a higher ranked operation level would be the result of the composition of the lower-level modes, thus highlighting the influence that a single component can have on the global behavior and increasing the knowledge of the system.

Also, although we stated in this thesis that a supervisory control for HVDC applications should ideally be designed by means of horizontal, vertical and modal decomposition approaches, horizontal decomposition was out of the scope of our work. Thus, industrial needs will certainly consider the possibility to decentralize the supervisory control and locate the decision-making program as close as possible to the sensors and actuators in each station. For this, a thorough analysis of the communication requirements in an HVDC system should be performed before any formal approach is integrated within the supervisory control design method.

Furthermore, the interface level of the implemented supervisory control could be largely improved. For instance, the continuous-time signals from the grid sensors are currently related to their corresponding discrete events via the detection of a predefined threshold's crossing. The implementation of automata could be considered so as to recognize the sequential evolution of a given variable or a given set of variables. In this manner, a more sophisticated interface could interact with complex observers so as to facilitate the integration of advanced control techniques, such as model predictive control [Mor11], into the supervision system.

From a practical perspective, future work could consider the extension of the proposed implementation method to other implementation languages other than C code. For instance, because the method is based on generic textual expressions, the former could be extended to Structured Text (ST) language [Int03], given that ST can be easily translated to C code and vice versa [Bas07]. Finally, although offline simulations are considered to be valid for the test of the ASC (as the latter is executed every 10 µs), the performance of the developed supervisory control during the operation of an MTDC grid could be evaluated in a real-time simulation software, such as HYPERSIM [Do99]. This program is very similar to EMTP-RV and so the C code obtained in this thesis could be equally implemented in HYPERSIM. The real-time simulation could be particularly interesting for the analysis of the communication requirements in an HVDC system in view of the decentralization of the ASC.

# Bibliography


[Abe17] M. Abedrabbo et al. “Impact of DC grid contingencies on AC system stability”. *13th IET International Conference on AC and DC Power Transmission (ACDC 2017)*, pp. 1–7, 2017.

[ADE81] ADEPA. “Guide d’étude des modes de marches et d’arrêts.” Technical report. France: ADEPA, 1981.

[Ahm12] N. Ahmed et al. “HVDC SuperGrids with modular multilevel converters - The power transmission backbone of the future”. *International Multi-Conference on Systems, Signals and Devices, SSD,* pp. 1–7, 2012.

[Ame15] A. Ametani, T. Ohno, and N. Nagaoka. *Cable system transients: theory, modeling and simulation*. John Wiley & Sons, 2015.

[ANS87] ANSI/IEEE. *C37.1-IEEE Standard Definition, Specification, and Analysis of Systems Used for Supervisory Control, Data Acquisition, and Automatic Control*. 1987.

[Arr07] J. Arrillaga, Y. H. Liu, and N. R. Watson. *Flexible power transmission: the HVDC options*. John Wiley & Sons, 2007.

[Aur17] G. Auran. “Full selective protection strategy for multi-terminal cable HVDC grids based on HB-MMC converters”. PhD thesis. Université Grenoble Alpes, Grenoble, France, 2017.

[Bal92] S. Balemi. “Control of discrete event systems: theory and application”. PhD thesis. Swiss Federal Institute of Technology, Zurich, Switzerland, 1992.

[Bar00] G. Barrett and S. Lafortune. “Decentralized supervisory control with communicating controllers”. *IEEE Transactions on Automatic Control,* vol. 45, no. 9, pp. 1620–1638, 2000.

[Bar04] K. Barnes, B. Johnson, and R. Nickelson. *Review Of Supervisory Control And Data Acquisition (SCADA) Systems*. Idaho National Engineering and Environmental Laboratory (INEEL), 2004.

[Bas07] F. Basile and P. Chiacchio. “On the implementation of supervised control of discrete event systems”. *IEEE Transactions on Control Systems Technology,* vol. 15, no. 4, pp. 725–739, 2007.

[Ben15] A. Benchaib. *Advanced control of AC/DC power networks: system of systems approach based on spatio-temporal scales*. John Wiley & Sons, 2015.

[Ber96] P. Bernus and L. Nemes. “A framework to define a generic enterprise reference architecture and methodology introduction: identification of what is a generic enterprise reference architecture and methodology”. *Computer Integrated Manufacturing Systems,* vol. 9, no. 3, pp. 179–191, 1996.

[Bis04] T. Biswas, A. Davari, and A. Feliachi. “Application of discrete event systems theory for modeling and analysis of a power transmission network”. *IEEE Power Systems Conference and Exposition,* vol. 2, pp. 1024–1029, 2000.

[Bru06] Brugg Kabel AG. “High voltage XLPE cable systems”. Technical User Guide. Switzerland: Brugg Kabel AG, 2006.

[Bui17] G. Buigues et al. “Present and future multiterminal HVDC systems : current status and forthcoming developments”. *Renewable Energy and Power Quality Journal,* vol. 1, no. 15, pp. 83–88, 2017.

[Can08] M. Cantarelli and J.-M. Roussel. “Reactive control system design using the supervisory control theory: evaluation of possibilities and limits”. *Proceedings of the 9th International Workshop on Discrete Event Systems, WODES,* pp. 200–205, 2008.

[Cas08] C. G. Cassandras and S. Lafortune. *Introduction to discrete event systems*. Springer, 2008.

[Cha95] F. Charbonnier, H. Alla, and R. David. “The supervised control of discrete event dynamic systems: a new approach”. *Proceedings of the 34th Conference on Decision and Control,* vol. 1, pp. 913–920, 1995.

[Cha99] F. Charbonnier, H. Alla, and R. David. “The supervised control of discrete event dynamic systems”. *IEEE Transactions on Control Systems Technology,* vol. 7, no. 2, pp. 175–187, 1999.

[Che93] K.-T. Cheng and A. S. Krishnakumar. “Automatic functional test generation using the extended finite state machine model”. *30th ACM/IEEE Design Automation Conference*, pp. 86–91, 1993.

[Cho17] Y. K. Choi. “Control device in HVDC system and operating method of thereof”. EP 3 200 306. European Patent Office, 2017.

[Cho59] N. Chomsky. “On certain formal properties of grammars”. *Information and Control,* vol. 2, no. 2, pp. 137–167, 1959.

[Cie88] R. Cieslak et al. “Supervisory control of discrete-event processes with partial observations”. *IEEE Transactions on Automatic Control,* vol. 33, no. 3, pp. 249–260, 1988.

[Das11] A. Das, H. Nademi, and L. Norum. “A method for charging and discharging capacitors in modular multilevel converter”. *37th Annual Conference on IEEE Industrial Electronics Society, IECON,* pp. 1058–1062, 2011.

[DeB13] S. DeBoeck et al. “Configurations and earthing of HVDC grids”. *Power and Energy Society General Meeting, PES,* pp. 1–5, 2013.

[Deb15] S. Debnath et al. “Operation, Control, and Applications of the Modular Multilevel Converter: A Review”. *IEEE Transactions on Power Electronics,* vol. 30, no. 1, pp. 37–53, 2015.

[Den13] S. Dennetiere et al. “Modeling of modular multilevel converters for the France-Spain link”. *IPST Proceedings,* pp. 1–7, 2013.

[Des13] J. Descloux. “Protection contre les courts-circuits des réseaux à courant continu de forte puissance”. PhD thesis. Université Grenoble Alpes, Grenoble, France, 2013.

[Die02] P. Dietrich et al. “Implementation considerations in supervisory control”. *Synthesis and control of discrete event systems*, pp. 185–201, 2002.

[Do99] V. Q. Do et al. “Hypersim, an integrated real-time simulator for power network and control systems”. *3rd International Conference on Digital Power System Simulators, IDCS’99,* Vasteras, Sweden, 1999.

[Don17] B. Donnot et al. “Introducing machine learning for power system operation support”. *10th Bulk Power Systems Dynamics and Control Symposium (IREP),* pp. 1–11, 2017.

[Dor13] J. Dorn and M. Pohl. “Transformation of the energy system in Germany - Enhancement of system stability by integration of innovative multilevel HVDC in the AC grid”. *Security in Critical Infrastructures Today, Proceedings of International ETG-Congress 2013; Symposium 1*, pp. 1–6, 2013.

[Dra14] T. Dragicevic et al. “Supervisory control of an adaptive-droop regulated DC microgrid with battery management capability”. *IEEE Transactions on Power Electronics,* vol. 29, no. 2, pp. 695–706, 2014.

[DyL67] T. Dy-Liacco. “The adaptive reliability control system”. *IEEE Transactions on Power Apparatus and Systems,* no. 5, pp. 517–531, 1967.

[Eco08] C. Economakos and G. Economakos. “FPGA implementation of PLC programs using automated high-level synthesis tools”. *IEEE International Symposium on Industrial Electronics,* pp. 1908–1913, 2008.

[Ene11] Energynautics GmbH. “European grid study 2030/2050”. Technical report. 2011.

[ENT13] ENTSO-E. “e-Highway2050: modular development plan of the pan-European transmission system 2050”. Brochure. 2013.

[Ess99] W. H. Esselman, D. J. Sobajic, and J. Maulbetsch. “Hybrid discrete and continuous control for power systems”. *Discrete Event Dynamic Systems,* vol. 9, no. 4, pp. 297–318, 1999.

[Eur10] European Comission. “Energy 2020: A strategy for competitive, sustainable and secure energy”. COM(2010) 639 final. 2010.

[Eur16] European Environmental Agency. “Renewable energy in Europe - recent growth and knock-on effects”. Technical report. 2016.

[Fab98] M. Fabian and A. Hellgren. “PLC-based implementation of supervisory control for discrete event systems”. *Proceedings of the 37th IEEE Conference on Decision and Control,* vol. 3, pp. 3305–3310, 1998.

[Far09] G. Faraut, L. Pietrac, 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.

[Far10] G. Faraut. “Commutations sûres de mode pour les Systèmes à Evénements Discrets”. PhD thesis. Institut National des Sciences Appliquées de Lyon, Villeurbanne, France, 2010.

[Fen06] L. Feng and W. M. Wonham. “Computationally efficient supervisor design: Control flow decomposition”. *Proceedings of the 8h International Workshop on Discrete Event Systems, WODES,* pp. 9–14, 2006.

[Fen08] L. Feng and W. M. Wonham. “Supervisory control architecture for discrete-event systems”. *IEEE Transactions on Automatic Control,* vol. 53, no. 6, pp. 1449–1461, 2008.

[Fin78] L. H. Fink and K. Carlsen. “Operating under stress and strain”. *IEEE Spectrum,* vol. 15, no. 3, pp. 48–53, 1978.

[Flo07] H. Flordal et al. “Compositional synthesis of maximally permissive supervisors using supervision equivalence”. *Discrete Event Dynamic Systems: Theory and Applications,* vol. 17, no. 4, pp. 475–504, 2007.

[Gao14] F. Gao et al. “Startup strategy of VSC-HVDC system based on modular multilevel converter”. *Energy Conversion Congress and Exposition, ECCE,* pp. 1946–1952, 2014.

[Gem08] B. Gemmell et al. “Prospects of multilevel VSC technologies for power transmission”. *Transmission and Distribution Conference and Exposition,* pp. 1–16, 2008.

[Gon18] J. C. González et al. “HVDC protection criteria for transient stability of AC systems with embedded HVDC links”. *14th IET International Conference Developments in Power System Protection, DPSP,* no. 15, pp. 956–960, 2018.

[Gou04] D. Gouyon, J.-F. Pétin, and A. Gouin. “Pragmatic approach for modular control synthesis and implementation”. *International Journal of Production Research,* vol. 42, no. 14, pp. 2839–2858, 2004.

[Har05] J. Hart. *Windows system programming*. Third edition. Addison-Wesley, 2005.

[Hau08] P. Haugland. “It’s time to connect: technical description of HVDC light technology”. Technical report. Sweden: ABB, 2008.

[Her10] D. van Hertem, M. Ghandhari, and M. Delimar. “Technical limitations towards a SuperGrid - A European prospective”. *IEEE International Energy Conference and Exhibition, EnergyCon,* pp. 302–309, 2010.

[Her16] D. van Hertem and O. Gomis-Bellmunt. *HVDC grids: for offshore and supergrid of the future*. John Wiley & Sons, 2016.

[His00] I. A. Hiskens and M. A. Pai. “Hybrid systems view of power system modelling”. *Proceedings of the IEEE International Symposium on Circuits and Systems, ISCAS,* vol. 2, pp. 228–231, 2000.

[Hop79] J. E. Hopcroft and J. D. Ullman. *Introduction to automata theory*. Addison-Wesley, 1979.

[Hwa12] U. Hwa Lee, S. Yuen Yuen, and Y.-S. Cho. “Energy Management System and control method using the same, for determining an operation mode of a high voltage direct current system”. US 8,169,106. U.S. Patent and Trademark Office, 2012.

[Int03] International Electrotechnical Commission. “Programmable controllers - Part 3: Programming languages”. International Standard IEC 61131-3. 2003.

[Jac10] B. Jacobson et al. “VSC-HVDC transmission with cascaded two-level converters”. *Cigré Session*, B4–B110, 2010.

[Jah94] F. Jahanian and A. Mok. “Modechart: a specification language for real-time systems”. *IEEE Transactions on Software Engineering,* vol. 20, no. 12, pp. 933–947, 1994.

[Jia00] S. Jiang and R. Kumar. “Decentralized control of discrete event systems with specializations to local control and concurrent systems”. *IEEE Transactions on Systems, Man, and Cybernetics, Part B,* vol. 30, no. 5, pp. 653–660, 2000.

[Kam04] O. Kamach. “Approche multi-modèle pour les systèmes à événements discrets: application à la gestion des modes de fonctionnement”. PhD thesis. Institut National des Sciences Appliquées de Lyon, 2004.

[Kam05] O. Kamach et al. “Supervisory uniqueness for operating mode systems”. *IFAC Proceedings Volumes,* vol. 38, no. 1, pp. 110–115, 2005.

[Kha07] M. S. Khan and M. R. Iravani. “Supervisory hybrid control of a micro grid system”. *IEEE Canada Electrical Power Conference, EPC,* pp. 20–24, 2007.

[Kle56] S. C. Kleene. “Representation of events in nerve nets and finite automata”. *Automata studies, Annals of Mathematical studies*. Princeton University Press, 1956.

[Kom08] J. Komenda et al. “Supervisory control of modular systems with global specification languages”. *Automatica,* vol. 44, no. 4, pp. 1127–1134, 2008.

[Kon15] E. Kontos et al. “Impact of HVDC transmission system topology on multiterminal DC network faults”. *IEEE Transactions on Power Delivery,* vol. 30, no. 2, pp. 844–852, 2015.

[Koo01] T. J. Koo, G. J. Pappas, and S. Sastry. “Mode switching synthesis for reachability specifications”. *International Workshop on Hybrid Systems: Computation and Control,* pp. 333–346, 2001.

[Kou00] X. D. Koutsoukos et al. “Supervisory control of hybrid systems”. *Proceedings of the IEEE,* vol. 88, no. 7, pp. 1026–1049, 2000.

[Kum91] R. Kumar, V. K. Garg, and S. I. Marcus. “On controllability and normality of DEDS”. *Systems and Control Letters,* vol. 17, pp. 157–168, 1991.

[Kun04] P. Kundur et al. “Definition and classification of power system stability”. *IEEE Transactions on Power Systems,* vol. 19, no. 2, pp. 1387–1401, 2004.

[Kun94] P. Kundur. *Power system stability and control*. New York: McGraw-Hill, 1994.

[Lau97] J. K. Lauzon, B. Mills, and S. C. Benhabib. “An implementation methodology for the supervisory control of flexible manufacturing workcells”. *Journal of Manufacturing Systems,* vol. 16, no. 2, p. 91, 1997.

[Lea12] A. B. Leal, D. L. L. da Cruz, and M. d. S. Hounsell. *PLC-based implementation of local modular supervisory control for manufacturing systems*. InTech, 2012.

[Led09] R. J. Leduc, P. Dai, and R. Song. “Synthesis method for hierarchical interface-based supervisory control”. *IEEE Transactions on Automatic Control,* vol. 54, no. 7, pp. 1548–1560, 2009.

[Leó16] W. R. León García et al. “Full-selective protection strategy for MTDC grids based on R-type superconducting FCLs and mechanical DC circuit breakers”. *5th IET International Conference on Renewable Power Generation, RPG,* pp. 43–50, 2016.

[Les03] A. Lesnicar and R. Marquardt. “An innovative modular multilevel converter topology suitable for a wide power range”. *IEEE Bologna PowerTech - Conference Proceedings,* vol. 3, pp. 272–277, 2003.

[Let14] W. Leterme et al. “Overview of grounding and configuration options for meshed HVDC grids”. *IEEE Transactions on Power Delivery,* pp. 1–9, 2014.

[Let15] W. Leterme and D. van Hertem. “Classification of fault clearing strategies for HVDC grids”. *Cigré International Symposium - Across Borders - HVDC Systems and Market Integration*, pp. 143–153, 2015.

[Li15] B. Li et al. “Closed-loop precharge control of modular multilevel converters during start-up processes”. *IEEE Transactions on Power Electronics,* vol. 30, no. 2, pp. 524–531, 2015.

[Lin88] F. Lin and W. M. Wonham. “Decentralized supervisory control of discrete-event systems”. *Information Sciences,* vol. 44, no. 3, pp. 199–224, 1988.

[Lop12] Y. K. Lopes et al. “Local modular supervisory implementation in microcontroller”. *Proceedings of the 9th International Conference of Modeling, Optimization and Simulation, MOSIM,* pp. 1–9, 2012.

[Lou17] D. S. Loume et al. “A multi-vendor protection strategy for HVDC grids based on low-speed DC circuit breakers”. *13th IET International Conference on AC and DC Power Transmission AC/DC,* pp. 1–6, 2017.

[Mah07] J. Mahseredjian et al. “On a new approach for the simulation of transients in power systems”. *Electric Power Systems Research,* vol. 77, no. 11, pp. 1514–1520, 2007.

[Mal02] P. Malik. “Generating controllers from discrete-event models”. *Proceedings MOdelling and VErification of Parallel processes, MOVEP,* pp. 337–342, 2002.

[Mal17] R. Malik et al. “Supremica - An efficient tool for large-scale discrete event systems”. *IFAC-PapersOnLine,* vol. 50, no. 1, pp. 5794–5799, 2017.

[Mar09] I. Martínez de Alegría et al. “Transmission alternatives for offshore electrical power”. *Renewable and Sustainable Energy Reviews,* vol. 13, no. 5, pp. 1027–1038, 2009.

[Mar98] F. Maraninchi and Y. Remond. “Mode-automata: about modes and states for reactive systems”. *European Symposium On Programming*, pp. 185–199, 1998.

[Mel04] A. P. Meliopoulos. *Power System Modeling, Analysis and Control*. Marcel Dekker, 2004.

[Mil89] R. Milner. *Communication and concurrency. Series in Computer Science*. Prentice-Hall, 1989.

[Moh11] S. Mohajerani et al. “On the use of observation equivalence in synthesis abstraction”. *3rd International Workshop on Dependable Control of Discrete Systems, DCDS’11 - Conference Proceedings*, pp. 84–89, 2011.

[Moh12a] S. Mohajerani. “On compositional supervisor synthesis for discrete event systems”. Licentiate thesis. Chalmers University of Technology, Gothenburg, Sweden, 2012.

[Moh12b] S. Mohajerani, R. Malik, and M. Fabian. "An algorithm for weak synthesis observation equivalence for compositional supervisor synthesis". *IFAC Proceedings Volumes (IFAC-PapersOnline)*, pp. 239–244, 2012.

[Moh14] S. Mohajerani, R. Malik, and M. Fabian. "A framework for compositional synthesis of modular nonblocking supervisors". *IEEE Transactions on Automatic Control,* vol. 59, no. 1, pp. 150–162, 2014.

[Mor11] M. Moradzadeh, L. Bhojwani, and R. Boel. "Coordinated Voltage Control via Distributed Model Predictive Control". *Chinese Control and Decision Conference*, pp. 1612–1618, 2011.

[Nat16] National Development and Reform Commission (NDRC). *13th Renewable Energy Development Five Year Plan (2016-2020)*. 2016.

[Nie95] E. Niel et al. "Operational-safety supervisory control: an approach to supervisor activation". *IEEE Symposium on Emerging Technologies and Factory Automation, ETFA'95,* vol. 2, pp. 553–561, 1995.

[Nou96] M. Nourelfath and E. Niel. "On the use of macroactions in Supervisory Control Theory for monitoring purpose". *Proceedings of the IEEE Conference on Emerging Technologies and Factory Automation, EFTA'96*, pp. 157–162, 1996.

[Oli07] T. E. Oliphant. "Python for scientific computing". *Computing in Science & Engineering,* vol. 9, no. 3, pp. 1–11, 2007.

[Ove00] A. Overkamp and J. H. van Schuppen. "Maximal solutions in decentralized supervisory control". *SIAM Journal on Control and Optimization,* vol. 39, no. 2, pp. 492–511, 2000.

[Pav00] M. Pavella, D. Ernst, and D. Ruiz-Vega. *Transient stability of power systems: a unified approach to assessment and control*. Kluwer Academic Publishers, 2000.

[Per12] J. Peralta et al. "Detailed and averaged models for a 401-level MMC-HVDC system". *IEEE Transactions on Power Delivery,* vol. 27, no. 3, pp. 1501–1508, 2012.

[PRO17] PROMOTioN project. "Preliminary analysis of key technical, financial, economic, legal, regulatory and market barriers and related portfolio of solutions". *Deliverable 12.1*, pp. 1–59, 2017.

[Pro95] J. Prosser et al. "Supervisory control of electric power transmission networks". *IEEE Transactions on Power Systems,* vol. 10, no. 2, pp. 1104–1110, 1995.

[Que02] M. de Queiroz and J. Cury. "Synthesis and implementation of local modular supervisory control for a manufacturing cell". *Proceedings of the 6th International Workshop on Discrete Event Systems, WODES*, pp. 377–382, 2002.

[Ram02] A. Ramirez-Serrano et al. "A hybrid PC/PLC architecture for manufacturing-system control theory and implementation". *Journal of Intelligent Manufacturing,* vol. 13, no. 4, pp. 261–281, 2002.

[Ram87] 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.

[Ram89] P. J. Ramadge and W. M. Wonham. "The control of discrete event systems". *Proceedings of the IEEE,* vol. 77, no. 1, pp. 81–98, 1989.

[Rao15] H. Rao. "Architecture of Nan'ao multi-terminal VSC-HVDC systems and its multi-functional control". *CSEE Journal of Power and Energy Systems,* vol. 1, no. 1, pp. 9–18, 2015.

[Rau14] P. Rault. "Dynamic modeling and control of multi-terminal HVDC grids". PhD thesis. Ecole Centrale de Lille, Lille, France, 2014.

[Ren16] R. H. Renner and D. Van Hertem. "Potential of using DC voltage restoration reserve for HVDC grids". *Electric Power Systems Research,* vol. 134, pp. 167–175, 2016.

[Rom17] M. Romero Rodríguez et al. "Supervisory Control for High-Voltage Direct Current Transmission Systems". *IFAC Proceedings Volumes (IFAC-PapersOnline),* vol. 50, no. 1, pp. 12326–12332, 2017.

[Rom19] M. Romero-Rodríguez et al. "An implementation method for the supervisory control of time-driven systems applied to high-voltage direct current transmission grids". *Control Engineering Practice,* vol. 82, no. 1, pp. 97–107, 2019.

[Rud92] K. Rudie and W. M. Wonham. "Think globally, act locally: decentralized supervisory control". *IEEE Transactions on Automatic Control,* vol. 37, no. 11, pp. 1692–1708, 1992.

[Sch04] J. H. van Schuppen. "Decentralized control with communication between controllers". *Unsolved problems in mathematical systems and control theory*, pp. 144–150, 2004.

[Sch08] K. Schmidt, T. Moor, and S. Perk. "Nonblocking hierarchical control of decentralized discrete event systems". *IEEE Transactions on Automatic Control,* vol. 53, no. 10, pp. 2252–2265, 2008.

[Sha05] M. Shahidehpour, W. F. Tinney, and Y. Fu. "Impact of security on power systems operation". *Proceedings of the IEEE,* vol. 93, no. 11, pp. 2013–2025, 2005.

[Shi17] K. Shinoda. "Control and energy management of MMC-based multi-terminal HVDC grids". PhD thesis. Ecole Centrale de Lille, Lille, France, 2017.

[Sip06] M. Sipser. *Introduction to the theory of computation*. Boston: Thomson Course Technology, 2006.

[Su10] R. Su, J. H. van Schuppen, and J. E. Rooda. "Aggregative synthesis of distributed supervisors based on automaton abstraction". *IEEE Transactions on Automatic Control,* vol. 55, no. 7, pp. 1627–1640, 2010.

[Tak98] S. Takai. "On the language generated under fully decentralized supervision". *IEEE Transactions on Automatic Control,* vol. 43, no. 9, pp. 1253–1256, 1998.

[Tie16] P. Tielens and D. Van Hertem. "The relevance of inertia in power systems". *Renewable and Sustainable Energy Reviews,* vol. 55, pp. 999–1009, 2016.

[Ulb14] A. Ulbig, T. S. Borsche, and G. Andersson. "Impact of low rotational inertia on power system stability and operation". *IFAC-PapersOnLine,* vol. 19, pp. 7290–7297, 2014.

[Uni04] Union for the Co-ordination of Transmission of Electricity. *Operation handbook*. 2004.

[Vás16] J. W. Vásquez et al. "Alarm management based on diagnosis". *IFAC-PapersOnLine,* vol. 49, no. 5, pp. 126–131, 2016.

[Vie17] A. D. Vieira et al. "A method for PLC implementation of supervisory control of discrete event systems". *IEEE Transactions on Control Systems Technology,* vol. 25, no. 1, pp. 175–191, 2017.

[Weh89] L. Wehenkel, T. Van Cutsem, and M. Ribbens-Pavella. "An artificial intelligence framework for online transient stability assessment of power systems". *IEEE transactions on Power Systems,* vol. 4, no. 2, pp. 789–800, 1989.

[Wen07] Q. Wen et al. "Fault-tolerant supervisory control of discrete event systems: Formulation and existence results". *IFAC Proceedings Volumes,* vol. 40, no. 6, pp. 175–180, 2007.

[Won04] K. C. Wong and W. M. Wonham. "On the computation of observers in discrete-event systems". *Discrete Event Dynamic Systems,* vol. 14, no. 1, pp. 55–107, 2004.

[Won17] W. M. Wonham and K. Cai. *Supervisory Control of Discrete-Event Systems*. Course notes, Dept. of Electrical & Computer Engineering, University of Toronto, 2017.

[Won96] K. C. Wong and W. M. Wonham. "Hierarchical control of discrete-event systems". *Discrete Event Dynamic Systems,* vol. 6, no. 3, pp. 241–273, 1996.

[Wu18] Z. Wu et al. "State-of-the-art review on frequency response of wind power plants in power systems". *Journal of Modern Power Systems and Clean Energy,* vol. 6, no. 1, pp. 1–16, 2018.

[Yoo02] T.-S. Yoo and S. Lafortune. “A general architecture for decentralized supervisory control of Discrete Event Systems”. *Discrete Event Dynamic Systems: Theory and Applications,* vol. 12, no. 3, pp. 335–377, 2002.

[Yu13] Y. Yu et al. “Pre-charging control strategies of modular multilevel Converter”. *International Conference on Electrical Machines and Systems, ICEMS*, pp. 1842–1845, 2013.

[Zam17] A. Zama. “Modélisation et commande des convertisseurs modulaires multiniveaux (MMCs) destinés aux réseaux HVDC”. PhD thesis. Université Grenoble Alpes, Grenoble, France, 2017.

[Zay17] J. Zaytoon and B. Riera. “Synthesis and implementation of logic controllers - A review”. *Annual Reviews in Control,* vol. 43, pp. 152–168, 2017.

[Zha06] H. Zhao, Z. Mi, and H. Ren. “Modeling and analysis of power system events”. *Proceedings of the Chinese Society of Electrical Engineering,* vol. 26, no. 22, pp. 11–16, 2006.

[Zha10] P. Zhang, F. Li, and N. Bhatt. “Next-generation monitoring, analysis, and control for the future smart control center”. *IEEE Transactions on Smart Grid,* vol. 1, no. 2, pp. 186–192, 2010.

[Zho90] H. Zhong and W. M. Wonham. “On the consistency of hierarchical supervision in discrete-event systems”. *IEEE Transactions on Automatic Control,* vol. 35, no. 10, pp. 1125–1134, 1990.

# A

# Operations on languages

***Definition A.1 (Concatenation)***
*For two languages $L_a$ and $L_b$ such that $L_a, L_b \subseteq \Sigma^*$, their concatenation, denoted by $L_a L_b$, is defined as:*

$$L_a L_b \coloneqq \{s \in \Sigma^* | (s = s_a s_b) \text{ and } (s_a \in L_a) \text{ and } (s_b \in L_b)\}. \quad \blacklozenge$$

In other words, all the strings in $L_a L_b$ can be written as a string contained in $L_a$ immediately followed by a string contained in $L_b$.

***Definition A.2 (Prefix-closure)***
*For a language $L$ such that $L \subseteq \Sigma^*$:*

$$\overline{L} \coloneqq \{s \in \Sigma^* | (\exists t \in \Sigma^*)[st \in L]\}. \quad \blacklozenge$$

In words, the prefix-closure of $L$, denoted by $\overline{L}$, is the language formed by all the prefixes of all the strings in $L$. A language $L$ is said to be *prefix-closed* if $L = \overline{L}$, that is, if any prefix of any string is also an element of $L$. Otherwise, $L \subset \overline{L}$.

***Definition A.3 (Kleene-closure)***
*For a language $L$ such that $L \subseteq \Sigma^*$:*

$$L^0 := \{\varepsilon\},\ L^{i+1} := LL^i,\ L^* := \bigcup_{i\geq 0} L^i. \qquad \blacklozenge$$

In words, the Kleene-closure operator (*), when applied to a language $L$, gives the infinite set resulting from the union of $L$ concatenated with itself at each occurrence. Notice that the Kleene-closure is an idempotent unary operator: $(L^*)^* = L^*$.

***Definition A.4 (Projection)***
*For two alphabets $\Sigma_1$ and $\Sigma_2$ such that $\Sigma_2 \subseteq \Sigma_1$, the projection $P_{1,2}$ applied to strings is a function such that:*

$$P_{1,2} : \Sigma_1^* \to \Sigma_2^*$$

*where*

$$P_{1,2}(\varepsilon) := \varepsilon,$$
$$P_{1,2}(\sigma) := \begin{cases} \sigma & \text{if } \sigma \in \Sigma_2, \\ \varepsilon & \text{if } \sigma \in \Sigma_1 \setminus \Sigma_2, \end{cases}$$
$$P_{1,2}(s\sigma) := P_{1,2}(s)P_{1,2}(\sigma) \text{ for } s \in \Sigma_1^*,\ \sigma \in \Sigma_1.$$

*Then, the projection operation applied to a language L such that $L \in \Sigma_1^*$ is defined as follows:*

$$P_{1,2}(L) := \{t \in \Sigma_2^* | (\exists s \in L)[P_{1,2}(s) = t]\}. \qquad \blacklozenge$$

In words, the projection operation takes a string $s$ formed over the larger alphabet $\Sigma_1$ and removes from it the symbols that are not included in the smaller alphabet $\Sigma_2$. The projection is then extended to a language by applying it to all the strings in the language.

***Definition A.5 (Inverse projection)***
*For two alphabets $\Sigma_1$ and $\Sigma_2$ such that $\Sigma_2 \subseteq \Sigma_1$, the inverse projection $P_{2,1}^{-1}$ is a function applied to strings and defined as follows:*

$$P_{2,1}^{-1} : \Sigma_2^* \to\ 2^{\Sigma_1^*}$$

*where*

$$P_{2,1}^{-1}(t) := \{s \in \Sigma_1^* | P_{1,2}(s) = t\}$$

*When applied to languages, and considering a language L such that $L \in \Sigma_2^*$, the inverse projection of a language is defined by:*

$$P_{2,1}^{-1}(L) := \{s \in \Sigma_1^* | (\exists t \in L)[P_{1,2}(s) = t]\}. \qquad \blacklozenge$$

In words, considering a string in the smaller alphabet $\Sigma_2$, the inverse projection $P_{2,1}^{-1}$ returns the set of all strings from the larger alphabet $\Sigma_1$ that project to the given string. The inverse projection is then extended to a language by applying it to all the strings in the language.

# B

# Operations on automata

***Definition B.1 (Accessible part of an automaton)***
*Considering an automaton G such that* $G = (Q, \Sigma, \delta, q_0, Q_m)$*, its accessible part* $Ac(G)$ *is defined as* $Ac(G) = (Q_{ac}, \Sigma, \delta_{ac}, q_0, Q_{m,ac})$*, where:*

- $Q_{ac} = \{q \in Q | (\exists s \in \Sigma^*)[\delta(q_0, s) = q]\}$
- $Q_{m,ac} = Q_m \cap Q_{ac}$
- $\delta_{ac} = \delta|_{Q_{ac} \times \Sigma \to Q_{ac}}$ *(notation "|" means "restricted to")* ♦

In an automaton $G$, an *accessible* state is a state reachable from the initial state $q_0$, that is, a state $q$ for which a string $s \in \Sigma^*$ such that $\delta(q_0, s) = q$ exists when extending the transition function to $\Sigma^*$. It is therefore possible to extract from the automaton $G$ the automaton $Ac(G)$ containing only the accessible states of $G$. This operation does not modify the languages $L(G)$ and $L_m(G)$ because the strings of these languages are generated from the initial state. However, all the states that are not accessible from the initial state are deleted, thus reducing the size of the automaton.

***Definition B.2 (Co-accessible part of an automaton)***
*Considering an automaton G such that* $G = (Q, \Sigma, \delta, q_0, Q_m)$*, its co-accessible part* $CoAc(G)$ *is defined as* $CoAc(G) = (Q_{coac}, \Sigma, \delta_{coac}, q_{0,coac}, Q_m)$*, where:*

- $Q_{coac} = \{q \in Q | (\exists s \in \Sigma^*)[\delta(q, s) \in Q_m]\}$

- $q_{0,coac} = \begin{cases} q_0 \text{ if} & q_0 \in Q_{coac} \\ \text{undefined} & \text{otherwise} \end{cases}$
- $\delta_{coac} = \delta|_{Q_{coac} \times \Sigma \to Q_{coac}}$ ♦

The co-accessible states $q$ are in turn all the states of $G$ allowing to reach a marked state, that is, all the states $q$ for which there exists a string $s \in \Sigma^*$ such that $\delta(q,s) \in Q_m$ exists. It is therefore possible to extract from $G$ the automaton $CoAc(G)$ containing only the co-accessible states of $G$. This operation can abridge the generated language $L(G)$ by deleting some states accessible from the initial state. The marked language, however, remains unaltered (as long as the initial state exists). So, in the case $G = CoAc(G)$, the following equality is respected: $L(G) = \overline{L_m(G)}$.

***Definition B.3 (Trim operation)***

$$Trim(G) = Ac[CoAc(G)] = CoAc[Ac(G)]$$

An automaton in which all states are both accessible and co-accessible is called a trim automaton. Because a string outgoing from a non co-accessible state cannot reach a marked state and there is no string in $L(G)$ that reaches a non accessible state, a trim automaton is always non-blocking (Definition 3.3).

***Definition B.4 (Product)***
*Considering two automata $G_1 = (Q_1, \Sigma_1, \delta_1, q_{01}, Q_{m1})$ and $G_2 = (Q_2, \Sigma_2, \delta_2, q_{02}, Q_{m2})$, the automaton $G = (Q, \Sigma, \delta, q_0, Q_m)$ is obtained by the product of $G_1$ and $G_2$ such that:*

$$G = G_1 \times G_2 = Ac(Q_1 \times Q_2, \Sigma_1 \cap \Sigma_2, \delta, (q_{01}, q_{02}), Q_{m1} \times Q_{m2}),$$

*where*

- $Q = Q_1 \times Q_2$
- $\Sigma = \Sigma_1 \cap \Sigma_2$
- $\delta((q_1,q_2),\sigma) = \begin{cases} (\delta_1(q_1,\sigma), \delta_2(q_2,\sigma)) & \text{if } \delta_1(q_1,\sigma) \text{ and } \delta_2(q_2,\sigma) \text{ are defined} \\ \text{undefined} & \text{otherwise} \end{cases}$
- $q_0 = (q_{01}, q_{02})$
- $Q_m = Q_{m1} \times Q_{m2}$ ♦

The product operation is commutative ($G_1 \times G_2 = G_2 \times G_1$) and associative [$G_1 \times (G_2 \times G_3) = (G_1 \times G_2) \times G_3 = G_1 \times G_2 \times G_3$]. The language resulting from the product is:

- $L(G_1 \times G_2) = L(G_1) \cap L(G_2)$ ;

- $L_m(G_1 \times G_2) = L_m(G_1) \cap L_m(G_2)$.

Therefore, the product makes it possible to obtain exclusively the strict synchronous behavior of two automata, that is, the behavior described by the events common to the two automata.

***Definition B.5 (Parallel composition)***
*Considering two automata* $G_1 = (Q_1, \Sigma_1, \delta_1, q_{01}, Q_{m1})$ *and* $G_2 = (Q_2, \Sigma_2, \delta_2, q_{02}, Q_{m2})$*, the automaton* $G = (Q, \Sigma, \delta, q_0, Q_m)$ *is obtained by the parallel composition of* $G_1$ *and* $G_2$ *such that:*

$$G = G_1||G_2 = Ac(Q_1 \times Q_2, \Sigma_1 \cup \Sigma_2, \delta, (q_{01}, q_{02}), Q_{m1} \times Q_{m2}),$$

*where*

- $Q = Q_1 \times Q_2$
- $\Sigma = \Sigma_1 \cup \Sigma_2$
- $\delta((q_1, q_2), \sigma) = \begin{cases} (\delta_1(q_1, \sigma), \delta_2(q_2, \sigma)) & \text{if } \delta_1(q_1, \sigma) \text{ and } \delta_2(q_2, \sigma) \text{ are defined} \\ (\delta_1(q_1, \sigma), q_2) & \text{if } \delta_1(q_1, \sigma) \text{ is defined and } \sigma \notin \Sigma_2 \\ (q_1, \delta_2(q_2, \sigma)) & \text{if } \delta_2(q_2, \sigma) \text{ is defined and } \sigma \notin \Sigma_1 \\ \text{undefined} & \text{otherwise} \end{cases}$
- $q_0 = (q_{01}, q_{02})$
- $Q_m = Q_{m1} \times Q_{m2}$ ♦

The parallel composition operation is commutative ($G_1||G_2 = G_2||G_1$), as long as the reordering of the name of the states in the composed automaton is neglected. This is acceptable since the name of the states is only indicative and does not provide any information for the operation of automata. The parallel composition is also associative [$G_1||G_2||G_3 = (G_1||G_2)||G_3 = G_1||(G_2||G_3)$]. Moreover, if $P_1$ and $P_2$ are respectively considered as the projections of $\Sigma$ to $\Sigma_1$ and $\Sigma_2$, the languages resulting of the parallel composition are:

- $L(G_1||G_2) = P_1^{-1}[L(G_1)] \cap P_2^{-1}[L(G_2)]$ ;
- $L_m(G_1||G_2) = P_1^{-1}[L_m(G_1)] \cap P_2^{-1}[L_m(G_2)]$.

Therefore, in the parallel composition, the events that are not common to both $\Sigma_1$ and $\Sigma_2$ can occur in $G$ as long as they are possible in $G_1$ or $G_2$, whereas common events cannot occur unless both automata execute them simultaneously. Thus, this operation can be seen as a synchronization of automata on common events.

It should be noted that if $\Sigma_1 = \Sigma_2$, then $G_1 \times G_2 = G_1||G_2$, since all transitions are forced to be synchronized.

# C

# Compositional approach: example

This appendix illustrates the compositional approach by means of the example presented in [Moh14]. The manufacturing system shown in Figure C.1 comprises two machines $M_1, M_2$ linked by two buffers $B_1, B_2$ that can store one workpiece each. Furthermore, using switches $W_1$ and $W_2$, the user can suspend (events $sus_1, sus_2$) and resume (events $res_1, res_2$) production of $M_1$ or $M_2$, respectively. Then, the automata models of the system is given in Figure C.2. All events are observable, and uncontrollable events are prefixed by an exclamation mark (!).

For instance, event $s_1$ represents the fact that the machine $M_1$ takes a workpiece from outside the system, event $!o$ models the processing of the workpiece by $M_1$, event $!f_1$ models the delivery of the workpiece to the buffer $B_1$ and event $s_3$ models the fact that $M_1$ takes a work-

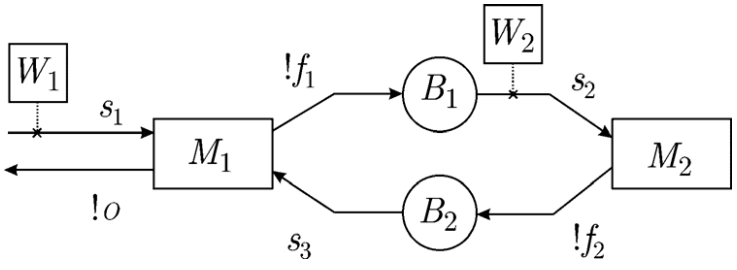



Figure C.1 – Manufacturing system overview [Moh14]

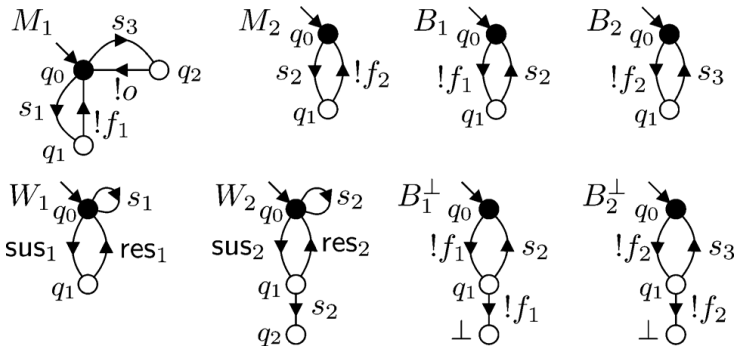



Figure C.2 – Automata models of the manufacturing system [Moh14]

piece from $B_2$. As for $M_2$, event $s_2$ models the fact that the machine takes workpieces from $B_1$ and event $!f_2$ models the delivery of the workpieces to $B_2$ after its processing. Then, $W_2$ models a requirement for the synthesized supervisor to prevent starting of $M_2$ in suspend mode, while $W_1$ models a plant where it is physically impossible to start $M_1$ in suspend mode. The models mentioned above are plants. Finally, the automata of the two buffers $B_1, B_2$, which are specification, represent the fact that only one workpiece can be stored in the buffer at a time, in order to avoid buffer overflow and under flow. In order to optimize the compositional synthesis algorithm developed by the authors of [Moh14], the specifications $B_1, B_2$ are transformed into the plant automata $B_1^{\perp}$ and $B_2^{\perp}$ by adding a transition for the uncontrollable events $!f_1, !f_2$ to a new blocking state ($\perp$). Then, the first step of the compositional approach is to simplify the initial models, defined by the set of plant automata $\mathcal{G}_0 = \{W_1, W_2, M_1, M_2, B_1^{\perp}, B_2^{\perp}\}$.

In consequence, because the controllable events $sus_1$ and $res_1$ are local to automaton $W_1$, the states $q_0$ and $q_1$ in $W_1$ can be merged, as the disabling of $sus_1$ and $res_1$ implies that either none or both of the states will always be removed. The resulting automaton $\tilde{W}_1$, given in Figure C.3, is thus synthesis equivalent to $W_1$. Because automaton $\tilde{W}_1$ is a selfloop-only automaton that always enables all its events, it can be disregarded in the synthesis [Moh14].

The events $sus_2$ and $res_2$ in automaton $W_2$ are abstracted similarly. However, the abstraction results in the non-deterministic automaton $\tilde{W}_2$ shown in Figure C.3. Thus, in order for a super-

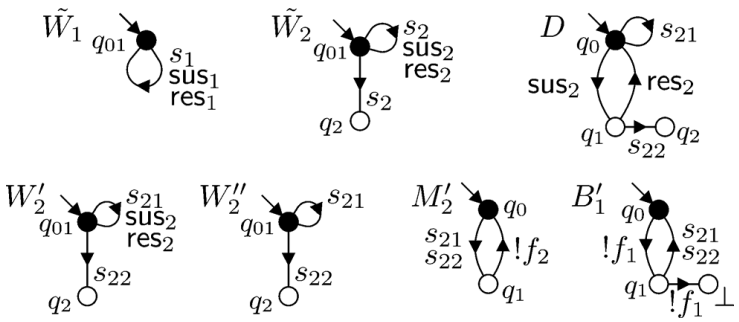



Figure C.3 – Abstraction results for switches in the manufacturing system example [Moh14]

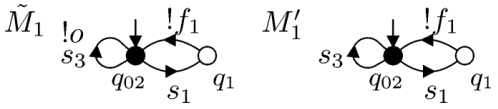



Figure C.4 – Abstracted automata of $M_1$ [Moh14]

visor to be able to discriminate whether to enable controllable event $s_2$ or not, based on the state of $\tilde{W}_2$; event $s_2$ in $\tilde{W}_2$ is renamed and replaced by two new events $s_{21}$ and $s_{22}$. The resulting deterministic automaton, given in Figure C.3, is denoted by $W'_2$. Then, because the events $sus_2$ and $res_2$ do not force a change in the state of the model, they are equally abstracted from $\tilde{W}_2$ [Moh14], which results in the automaton $W''_2$ of Figure C.3.

The automaton $D$ (named distinguisher), on the other hand, is a copy of the automaton $W_2$ such that the event $s_2$ has also been renamed in order to discriminate the transition related to $s_{21}$ from the one related to $s_{22}$. This automaton is stored in the set $\mathcal{S}$ of collected supervisors. Finally, the automata $M_2$ and $B_1^{\perp}$ need to be equally renamed so as to recognize the events $s_{21}$ and $s_{22}$. The renamed models are respectively denoted by $M'_2$ and $B'_1$ (Figure C.3).

Finally, because $\tilde{W}_1$ is no longer considered in the synthesis, the events $!o$ and $s_1$ are local to $M_1$ and so the states $q_0$ and $q_2$. However, $q_0$ and $q_1$ cannot be merged since $q_1$ can be a blocking state if $!f_1$ is disabled by other components [Moh14]. Since event $!o$ now only appears in a selfloop transition it can be removed from the automaton, resulting in the simplified model $M'_1$ given in Figure C.4.

At this point, each of the automata of the system has been simplified individually by means of the abstraction methods described in [Moh14]. Thus, the set of plant automata considered for the compositional synthesis is now defined to be $\mathcal{G} = \{W''_2, M'_1, M'_2, B'_1, B_2^{\perp}\}$. However, since none of these automata can be simplified further individually, they are selectively composed so as to be able to apply the abstraction methods onto the resulting automata. For instance, the automata $M'_1$ and $B'_1$ are composed, which causes $!f_1$ to become an event local to the resulting automaton $M'_1 \parallel B'_1$ (Figure C.5). Furthermore, $!f_1$ leads to the blocking state $\perp$. Then, in order to avoid the state $\perp$, and since $!f_1$ is uncontrollable, the controllable event $s_1$ is disabled (and the transition from $q_3$ to $q_1$ removed). This results in the supervisor $M'_1 \parallel B'_1$, shown in Figure C.5, which locally disables the event $s_1$ so as to avoid the blocking of the machine $M_1$. Hence, the

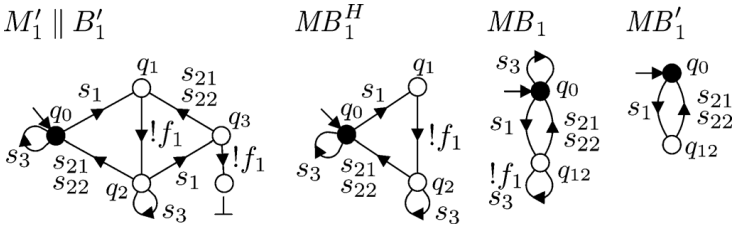



Figure C.5 – $M'_1 \parallel B'_1$ and its abstraction result [Moh14]

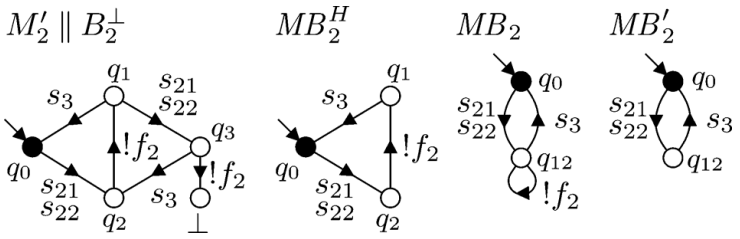



Figure C.6 – $M_2' \parallel B_2^{\perp}$ and its abstraction result [Moh14]

abstracted automaton $MB_1^H$ is added to the set $\mathcal{S}$ of collected supervisors.

However, although $MB_1^H$ acts as a supervisor with respect to $M_1$, it can still be regarded for the compositional synthesis as a plant automaton that needs to be simplified, given that it may restrict the behavior of the other models. Thus, states $q_1$ and $q_2$ are merged as a result of the abstraction of the transition associated to the local event $!f_1$; while the events $s_3$ and $!f_1$ are finally removed given that they only appears on selfloops in the entire model [Moh14]. The resulting abstracted plant automaton, denoted by $MB_1'$, is shown in Figure C.5. A similar procedure is applied to $M_2' \parallel B_2^{\perp}$, as illustrated in Figure C.6. Thus, the automaton $MB_2^H$, which is a non-blocking local supervisor of $M_2$, is included in the set $\mathcal{S}$ of collected supervisors. In addition, the model is further simplified, this time as a plant automaton, into $MB_2'$.

At this point of the approach, the set of uncontrolled plant models is $\mathcal{G} = \{W_2'', MB_1', MB_2'\}$ and the set of collected supervisors includes $\mathcal{S} = \{D, MB_1^H, MB_2^H\}$. Since no further individual abstraction is possible for each of the plant automata $(W_2'', MB_1', MB_2')$, a new automaton $\parallel \tilde{\mathcal{G}}$, shown in Figure C.7, is calculated from their composition. This automaton can only be simplified by monolithic synthesis, since there are no other models that allow to discriminate between the local and shared events. Hence, the associated supervisor $S_1$, given in Figure C.7, is added to the set of modular non-blocking supervisors. This set includes the result of the compositional synthesis for the original plants $(M_1, M_2, W_1, W_2)$ and is defined to be $\mathcal{S} = \{D, MB_1^H, MB_2^H, S_1\}$. Hence, the composition of the supervisors $MB_1^H, MB_2^H$ and $S_1$, along with the translation of the original plants alphabet to the supervisors alphabet provided by the distinguisher $D$; results in the least restrictive, controllable and non-blocking supervisor for the system, which produces the same controlled behavior as would a monolithic supervisor [Moh14].

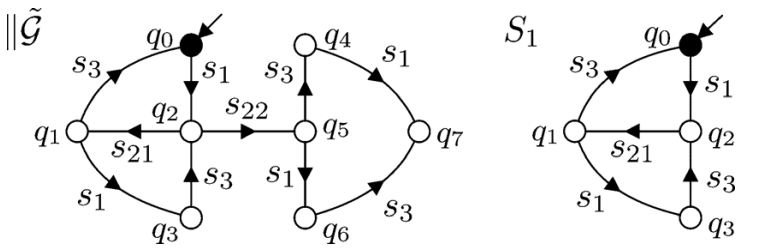



Figure C.7 – Final abstracted system and the calculated supervisor for $\parallel \mathcal{G}$ [Moh14]

To conclude, the compositional approach searches to simplify individually each of the plant automata of considered system by means of formal abstraction methods, and renaming if needed. Once this is no longer possible, the automata that share a certain behavior are composed so as to obtain a new model that could be further simplified. The abstraction-renaming-composition procedure is realized until only one plant model remains. Then, a final supervisor is calculated on the remaining plant. In addition, at any given stage of the compositional synthesis, intermediate supervisors that restrict the behavior of a given set of components are retained whenever it is necessary. The composition of the intermediate and final supervisors defines the result of the compositional synthesis for the considered system.

# D

# Conversion algorithms

In this appendix, the two algorithms defined in [Vie17] for the conversion of plant automata in view of their transformation into Moore machines are given. These algorithms ensure that the resulting Moore machines are deterministic by removing the self-loop transitions (Algorithm 1) from the plant automata and by associating a single event to each state (Algorithm 2) of the plants. The algorithms manipulate the following automata:

$$G_x = (Q^{G_x}, \Sigma^{G_x}, \delta^{G_x}, q_0^{G_x}, Q_m^{G_x})$$
$$\dot{G}_x = (Q^{\dot{G}_x}, \Sigma^{\dot{G}_x}, \delta^{\dot{G}_x}, q_0^{\dot{G}_x}, Q_m^{\dot{G}_x})$$
$$\ddot{G}_x = (Q^{\ddot{G}_x}, \Sigma^{\ddot{G}_x}, \delta^{\ddot{G}_x}, q_0^{\ddot{G}_x}, Q_m^{\ddot{G}_x})$$

***Algorithm 1 (Conversion of automaton $G_x$ into automaton $\dot{G}_x$)***

*Start with* $\dot{G}_x = G_x$;

*For every* $q \in Q^{\dot{G}_x}$ :

*If* $\Sigma_q \subseteq \Sigma^{\dot{G}_x}$ *is the set of all events in self-loop at q;*

*If* $\Sigma_q \neq \emptyset$, *then perform items* $(i)$ *through* $(v)$:

*i) Increase the set of states as:* $Q^{\dot{G}_x} = Q^{\dot{G}_x} \cup \{q_1, q_2\}$ *such that* $q_1, q_2 \notin Q^{\dot{G}_x}$;

*ii)* $q_0^{\dot{G}_x} = q_1$ *if and only if* $q_0^{G_x} = q$;

*iii)* $q_1, q_2 \in Q_m^{\dot{G}_x}$ *if and only if* $q \in Q_m^{G_x}$;

*iv) Considering that* $\Sigma_q = \{\sigma_1, \sigma_2, \ldots, \sigma_n\}$, *modify the state transition function by:*

*- excluding arcs with events in self-loop at state q:*

$\delta^{\dot{G}_x} = \delta^{\dot{G}_x} - \{(q, \sigma_1, q), (q, \sigma_2, q), \ldots, (q, \sigma_n, q)\}$;

*- including an arc from* $q_1$ *to* $q_2$ *labeled with every event in* $\Sigma_q$:

$\delta^{\dot{G}_x} = \delta^{\dot{G}_x} \cup \{(q_1, \sigma_1, q_2), (q_1, \sigma_2, q_2), \ldots, (q_1, \sigma_n, q_2)\}$;

*- including an arc from* $q_2$ *to* $q_1$ *labeled with every event in* $\Sigma_q$:

$\delta^{\dot{G}_x} = \delta^{\dot{G}_x} \cup \{(q_2, \sigma_1, q_1), (q_2, \sigma_2, q_1), \ldots, (q_2, \sigma_n, q_1)\}$;

*- for every* $p \in Q^{\dot{G}_x}$, *with* $p \neq q$:

*- for every* $\sigma \in \Sigma^{\dot{G}_x}$:

*- if* $(p, \sigma, q) \in \delta^{\dot{G}_x}$, *then replace the existing arc from p to* $q_1$ *with a new arc from p to* $q_1$: $\delta^{\dot{G}_x} = (\delta^{\dot{G}_x} - \{(p, \sigma, q)\}) \cup \{(p, \sigma, q_1)\}$;

*- if* $(q, \sigma, p) \in \delta^{\dot{G}_x}$, *then replace the existing arc from* $q_1$ *to* $q_2$ *with new arcs from* $q_1$ *and* $q_2$ *to p:*

$\delta^{\dot{G}_x} = (\delta^{\dot{G}_x} - \{(q, \sigma, p)\}) \cup \{(q_1, \sigma, p), (q_2, \sigma, p)\}$;

*v) Exclude the state that had events in self-loop:* $Q^{\dot{G}_x} = Q^{\dot{G}_x} - \{q\}$. ♦

***Algorithm 2 (Conversion of automaton $\dot{G}_x$ into automaton $\ddot{G}_x$)***

*i)* $\Sigma^{\ddot{G}_x} = \Sigma^{\dot{G}_x}$;

*ii) Start with* $Q^{\ddot{G}_x} = \emptyset$;

*iii)* $q_0^{\ddot{G}_x} = (q_0^{\dot{G}_x}, \sigma_{dummy})$ *such that* $\sigma_{dummy} \notin \Sigma^{\dot{G}_x}$; *include* $q_0^{\ddot{G}_x}$ *in* $Q^{\ddot{G}_x}$;

*iv) Repeat until every state* $(q, \sigma_2) \in Q^{\ddot{G}_x}$ *has been evaluated, where* $q \in Q^{\dot{G}_x}$ *and* $\sigma_2 \in (\Sigma^{\dot{G}_x} \cup \{\sigma_{dummy}\})$:

*- Choose* $(q, \sigma_2) \in Q^{\ddot{G}_x}$, *a state that has not been evaluated;*

*- For every* $\sigma_1 \in \Sigma^{\dot{G}_x}$:

*- If* $\delta^{\dot{G}_x}(q, \sigma_1)$ *is defined, then:* $\delta^{\ddot{G}_x}((q, \sigma_2), \sigma_1) = \delta^{\dot{G}_x}((q, \sigma_1), \sigma_1)$;

*include* $(\delta^{\dot{G}_x}(q, \sigma_1), \sigma_1)$ *into* $Q^{\ddot{G}_x}$;

*v)* $(q, \sigma) \in Q_m^{\ddot{G}_x}$ *if and only if* $q \in Q_m^{\dot{G}_x}$. ♦

# E

# Simulation parameters

The simulation parameters used in Chapter 6 are detailed in Table E.1. These parameters can be found in [Den13; Lou17; Shi17].

| | Parameter | | Value |
|---|---|---|---|
| | Simulation time-step | $\Delta$t | 10 µs |
| Station AC side | Line-to-line AC grid voltage | $V_{ac}$ | 320 kVrms |
| | Nominal AC grid frequency | $f$ | 50 Hz |
| | Transformer rated power | $S_{ac}$ | 500 MVA |
| MMC | Rated MMC voltage | $V_{mmc}$ | 640 kV |
| | Number of SMs per arm | $N$ | 400 |
| | SM capacitance | $C_{sm}$ | 13.02 mF |
| | Total capacitance of an arm | $C_{arm}$ | 32.55 µF |
| | Aggregated capacitance of MMC | $C_{mmc}$ | 195.3 µF |
| | Electrostatic constant | $H_c$ | 40 ms |
| | Arm inductance | $L_{arm}$ | 48.9 mH |
| | Arm resistance | $R_{arm}$ | 0.4 Ω |
| | AC filter inductance | $L_f$ | 58.7 mH |
| | AC filter resistance | $R_f$ | 0.102 Ω |
| DC grid | Rated DC grid voltage | $V_{dc}$ | ±320 kV |
| | Rated power per pole | $S_n$ | 500 MVA |
| | Rated current | $I_{dc}$ | 1560 A |
| | Length of Cable 11/Cable 22 | $l_{12}$ | 120 km |
| | Length of Cable 12/Cable 31 | $l_{13}$ | 150 km |
| | Length of Cable 21/Cable 32 | $l_{23}$ | 100 km |

Table E.1 – Simulation parameters

# Résumé étendu en français

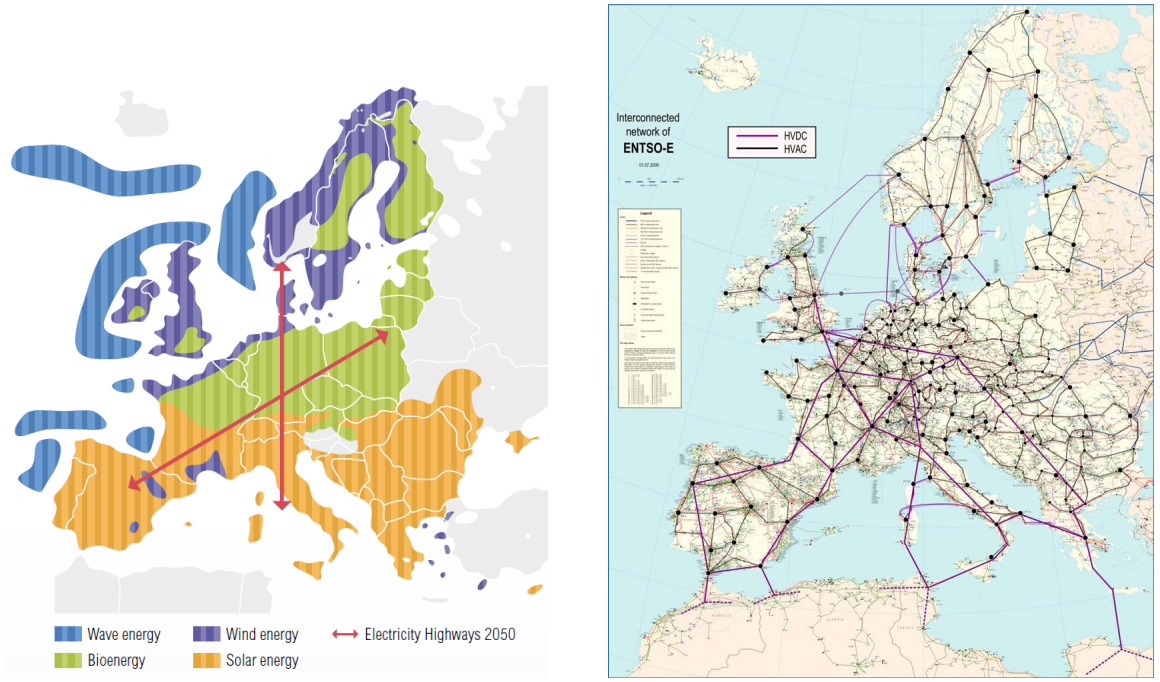
## Contexte de la thèse

La technologie est l'un des principaux piliers sur lesquels repose la société moderne: la communication, la finance, la logistique et les transports font partie d'une variété de secteurs basés sur des solutions technologiques. Malgré l'extraordinaire polyvalence des technologies actuelles, une grande partie d'entre elles sont alimentées par l'électricité: ordinateurs, éclairage, trains ...; l'électricité se révèle essentielle pour notre vie. À l'avenir, cette dépendance devrait augmenter en raison de la consommation croissante d'énergie et de la diminution de l'utilisation des combustibles fossiles comme principales sources d'énergie. En effet, les préoccupations concernant le développement durable et les questions environnementales ont poussé de nombreux pays à revoir leurs politiques nationales afin d'encourager la décarbonisation du secteur énergétique. L'Union européenne et ses États membres, par exemple, ont convenu de produire 20% de leur mix énergétique à partir de sources d'énergie renouvelable d'ici 2020, tandis que la Chine s'attend à une part de 15% d'énergie non fossile pour la même période [Nat16; Eur10]. Sous l'impulsion de ces initiatives, les énergies renouvelables ont connu une croissance fulgurante, puisqu'elles représentaient 77% de la capacité de production nouvellement installée en Europe de 2007 à 2015 [Eur16].

Cependant, l'intégration d'une quantité massive d'électricité produite à partir de sources d'énergie renouvelables dans les systèmes de transport d'énergie existants représente encore un défi à ce jour. La volatilité inhérente aux sources d'énergie renouvelables perturbe le fonctionnement traditionnel des réseaux électriques, qui repose sur une consommation largement prévisible et des générateurs entièrement contrôlables. De plus, étant donné que les centrales renouvelables, telles que les parcs éoliens en mer, sont très probablement situées loin des centres de consommation, elles conduisent les systèmes de transmission électrique existants, basés sur la technologie de courant alternatif (CA), proches de leurs limites physiques. . En conséquence, le Réseau Européen des Gestionnaires de Réseaux de Transport d'Electricité (REGRT-E) a signalé la nécessité de renforcer le réseau actuel si les corridors de transport à haute tension (HT) reliant les régions disposant de ressources renouvelables à fort potentiel dans la mer du Nord, l'océan

Atlantique et la région méditerranéenne aux zones densément peuplées de tout le continent européen doivent être développés. Une augmentation significative de la capacité d'interconnexion avec les pays voisins devrait améliorer la sécurité et la fiabilité de l'alimentation électrique en atténuant l'effet de la volatilité de ce type de sources d'énergie [Shi17].

La transmission à courant continu haute tension (HVDC) est considérée comme la solution la plus prometteuse et la plus réalisable sur le plan technique pour la mise à niveau fondamentale des systèmes de transmission AC existants [Mar09; Ahm12]. Pour les longues distances, la transmission de puissance devient impossible en CA à cause de l'effet capacitif dans la liaison haute tension, que ce soit un câble sous-marin ou souterrain ou une ligne aérienne. A une certaine distance, qui dépend de la puissance échangée, les pertes capacitives deviennent trop importantes: la puissance réactive échangée devient trop importante par rapport à la puissance active utile. Cela rend l'énergie éolienne en mer pratiquement inaccessible grâce à la technologie conventionnelle haute tension CA au-delà de 100 km. Dans la transmission HVDC, toutefois, comme seule la puissance active est échangée, les pertes capacitives sont éliminées et il existe moins de limitations sur la longueur de la liaison. Cela fait de la transmission HVDC la solution la plus économique après une distance de sécurité, qui peut être considérée comme étant d'environ 50 km pour la transmission par câble [Her10]. De plus, l'absence de fréquence dans les systèmes à courant continu permet l'interconnexion entre deux systèmes de transmission à courant alternatif asynchrone. En conséquence, la stabilité et la sécurité de l'ensemble




(a) Développement des sources d'énergie renouvelable d'ici 2050 [ENT13]

(b) Possible réseau HT d'ici 2050 [Ene11]

Figure 1 – Perspectives sur le développement d'un supergrid européen

du réseau sont grandement améliorées, car une perturbation sur un système ne se propage pas de l'autre côté de la liaison HVDC [Dor13].

Bien que la technologie HVDC soit utilisée depuis plus de 60 ans, les applications existantes se limitent pour la plupart à des liaisons point à point ad hoc répondant à un besoin spécifique [Arr07]. Ces dernières années, toutefois, l'idée de travailler à l'interconnexion à grande échelle de liaisons HVDC afin de former un réseau multi-terminal CC (MTDC) [Bui17] a suscité l'intérêt de chercheurs du monde universitaire et de l'industrie. Si le nombre de liaisons interconnectées augmente, les réseaux MTDC peuvent évoluer vers un réseau électrique en masse, nommé "supergrid", superposé au système de transmission CA [Her16]. Le supergrid est considéré comme une solution attrayante pour prendre en compte les sources d'énergie renouvelable dispersées géographiquement (Figure 1a) et pour atténuer la congestion des réseaux CA existants; tout en permettant une meilleure intégration des différents marchés de l'électricité et en augmentant la fiabilité et la flexibilité de l'alimentation [Her10]. La réalisation d'un projet aussi ambitieux devrait toutefois se faire progressivement à partir de l'interconnexion de liaisons HVDC distinctes, étant donné le grand nombre de défis que la construction d'un réseau international de grandes routes électriques implique (Figure 1b).

Les systèmes de puissance électrique sont généralement composés d'un grand nombre de composants disposés dans des zones géographiques distinctes afin d'assurer la production, la transmission et la distribution de l'énergie des centrales aux utilisateurs finaux. Tel que défini par la Commission européenne, l'entité chargée de la transmission de l'énergie des unités de production au réseau électrique aux opérateurs de distribution d'électricité régionaux ou locaux est le gestionnaire du réseau de transport (GRT).

Les systèmes d'alimentation en courant alternatif traditionnels sont principalement dominés par les grandes machines tournantes synchrones qui sont exploitées par un nombre réduit de services publics. Ensuite, des réseaux de transmission ont été développés pour relier les zones de production et de consommation en fonction des besoins et des prévisions à long terme de quelques services publics, ce qui facilite la surveillance et l'exploitation de toute la chaîne d'approvisionnement à partir d'un centre de contrôle centralisé régions ou régions densément peuplées. Cependant, la déréglementation économique du marché de l'énergie et l'émergence de la production d'énergie renouvelable, conjuguée à l'interconnexion croissante des réseaux nationaux, ont accru le nombre d'acteurs impliqués dans la transmission de l'électricité, ce qui réseau et dans une énorme quantité de données brutes à traiter au centre de contrôle. De plus, en raison du nombre élevé d'appareils électroniques de puissance dans un système HVDC, les perturbations se propagent rapidement dans le réseau CC. Si de telles perturbations ne sont pas correctement neutralisées par le centre de contrôle, des défaillances de composants en cascade pourraient éventuellement conduire à une panne du réseau et laisser la perturbation se propager aux réseaux CA adjacents. Tous ces aspects poussent les capacités des centres de contrôle actuels, qui reposent principalement sur l'expertise humaine et le contrôle local des centrales

électriques, à leurs limites.

Pour cette raison, les principales fonctions de surveillance, d'analyse et de contrôle des futurs centres de contrôle devraient être automatisées et les différents composants du système devraient être coordonnés efficacement. Comme indiqué dans [Zha10], pour répondre aux besoins opérationnels, les futurs centres de contrôle fourniront aux opérateurs humains des informations pertinentes plutôt que des données brutes afin que l'opérateur puisse facilement comprendre la situation du système. En outre, les méthodes en ligne et les conditions futures du système pourraient être prises en compte dans l'analyse du système afin d'en améliorer la fiabilité au moyen d'actions proactives compte tenu des incertitudes liées à l'intégration des sources d'énergie renouvelables. Enfin, les réseaux électriques modernes étant de plus en plus interconnectés, une perturbation peut entraîner de multiples systèmes de protection et de contrôle, alors que les systèmes de contrôle traditionnels exécutent des actions isolées pour résoudre un problème particulier. Par conséquent, comme il peut exister des interactions négatives susceptibles d'aggraver les conditions du système, les stratégies de protection et de contrôle des différentes stations doivent être coordonnées et amener rapidement le système à un état de fonctionnement sécurisé avec un minimum d'efforts de contrôle.

En 2010, le gouvernement français a lancé un appel à propositions pour la création d' *Instituts pour la Transition Energétique* (ITEs), qui devraient être des plateformes de recherche collaborative dans le domaine des énergies sobres en carbone, rassemblant l'expertise de la recherche industrielle et publique dans la logique du co-investissement public-privé et la coopération étroite entre toutes les parties prenantes du secteur. Le projet SuperGrid Institute (SGI) a été sélectionné avec succès pour développer les technologies des futurs réseaux de transmission HVDC. L'institut compte plusieurs partenaires industriels et académiques parmi ses actionnaires, résumés dans le Tableau 1.

Les travaux de recherche sont menés en collaboration avec des laboratoires publics et canal-

| SuperGrid Institute | Industrial | Academic and institutions |
|---|---|---|
| **Shareholders** | Alstom<br>EDF<br>GE<br>Nexans<br>Vettiner | CentraleSupélec<br>Ecole Centrale de Lyon<br>Grenoble INP<br>INSA Lyon<br>Université Claude Bernard Lyon 1<br>Université Joseph Fourier<br>Université Paris Sud |
| **Associated laboratories** | | CREMHYG<br>G2Elab<br>Laboratoire Ampère<br>Laboratoire des Signaux et Systèmes (L2S) |

Tableau 1 – Partenaires de SuperGrid Institute

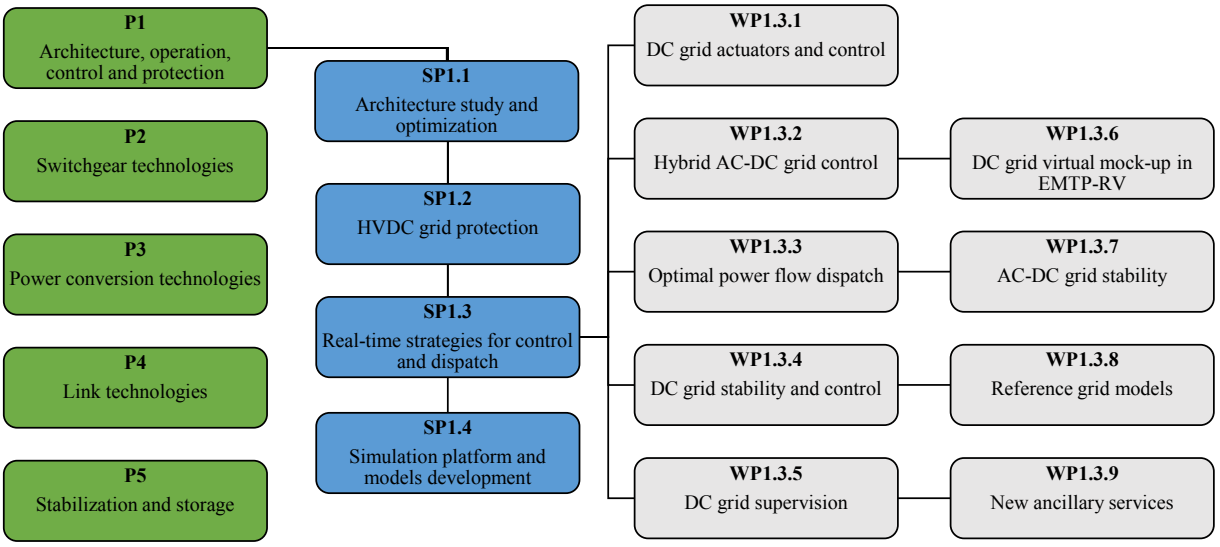



Figure 2 – Programmes de recherche de SuperGrid Institute

isés au travers de cinq programmes, présentés dans la Figure 2. Le programme 1 est consacré à l'étude de l'architecture du réseau ainsi qu'aux stratégies d'exploitation, de contrôle et de protection associées; et au développement d'outils de simulation pour le reste de SGI. L'un des principaux défis liés au développement des réseaux MTDC concerne la gestion opérationnelle de systèmes de plus en plus grands et complexes. Ainsi, le sous-programme 1.3 est chargé de la recherche sur les solutions de commande et d'exploitation pour les réseaux HVDC. En collaboration avec le Laboratoire Ampère, cette thèse a été réalisée dans le cadre du work package (WP) 1.3.5 " Supervision des réseau à courant continu".

Ainsi, la motivation de cette thèse reflète l'intérêt croissant pour les réseaux MTDC et les préoccupations suscitées par les défis opérationnels auxquels les réseaux futurs seront confrontés. Cette thèse a pour objectif de développer, tester et valider une méthode pour la conception et la mise en oeuvre pratique d'un système de contrôle automatisé basé sur la modélisation des systèmes à événements discrets et des méthodes formelles dans le cadre de la théorie du contrôle un fonctionnement coordonné et sécurisé du réseau du niveau composant au niveau du réseau. La principale contribution de cette thèse est donc de proposer une méthode systématique, autonome et incrémentale permettant de concevoir et de mettre en oeuvre un contrôle par supervision pour les systèmes HVDC.

Dans un système de transmission à courant alternatif, l'un des rôles du GRT est de gérer la sécurité du système électrique en temps réel et de maintenir un équilibre continu entre l'électricité fournie par les centrales et la demande des consommateurs, de manière à compenser les fluctuations de fréquence. car et les interruptions d'approvisionnement sont évitées. En outre, le GRT doit prévoir la consommation électrique future pour s'assurer que l'alimentation peut répondre à la demande et que la sécurité du système peut être maintenue. Le travail de planification comprend la coordination de l'entretien périodique des groupes électrogènes et l'achat

| System | AC | DC |
|---|---|---|
| Common variable | Angular frequency | DC voltage |
| | $\omega$ | $v_{dc}$ |
| Stored energy | Kinetic energy in rotating mass | Electrostatic energy in capacitor |
| | $E_r = \frac{1}{2} J\omega^2$ | $E_c = \frac{1}{2} C v_{dc}^2$ |
| Inertia or electrostatic constant | Generator | Converter |
| | $H = 2 \sim 10$ s | $H_c = 40$ ms |
| System inertia or electrostatic constant | Continental Europe | Reasonable value |
| | $H_{sys} = 5 \sim 6$ s | $H_{c\,sys} \approx 40$ ms |
| Swing equation (per unit) | $\frac{d\bar{\omega}}{dt} = \frac{\bar{P}_g - \bar{P}_l}{2H_{sys}}$ | $\frac{d\bar{v}_{dc}}{dt} = \frac{\bar{P}_{rec} - \bar{P}_{inv}}{2H_{c,sys}}$ |

Tableau 2 – Analogie entre les systèmes CA et CC [Shi17]

de services auxiliaires pour soutenir le fonctionnement du système électrique. De manière analogue aux systèmes à courant alternatif, pour un fonctionnement sécurisé d'un système HVDC, le GRT doit maintenir l'équilibre entre la puissance injectée et retirée du réseau; et par conséquent maintenir la tension continue dans une plage de fonctionnement nominale.

Traditionnellement, la répartition en temps réel de la combinaison optimale de production d'électricité et de services auxiliaires est assistée par ordinateur, tandis que la gestion de la sécurité de tout événement occasionnant l'équilibre entre l'offre et la demande peut être assistée par ordinateur ou fondée sur des décisions humaines. . Cependant, l'augmentation des besoins en énergie, des interconnexions de systèmes et des défis opérationnels liés aux différences technologiques entre les systèmes CA et CC nécessite le développement de méthodes de contrôle permettant de gérer les interactions complexes des nombreux événements apparaissant dans un système de transmission de puissance [Ess99].

Néanmoins, les différentes analogies faites entre les systèmes AC et DC sont résumées dans le Tableau 2. Les données données impliquent que, pour la même perturbation de puissance (dans le système par unité), le taux de variation de la tension continue serait environ 100 fois plus important que celui de la fréquence dans un réseau alternatif. entraîner la contrainte sur le temps de réponse du système de contrôle. Lorsque le contrôle de la tension ne parvient pas à rétablir l'équilibre de la puissance et que la tension continue s'effondre, des contrôles de sécurité doivent

être initiés afin que les convertisseurs soient toujours en mesure de rétablir le flux d'énergie dans le réseau. Alors que dans les systèmes à courant alternatif, ces actions de contrôle peuvent durer plusieurs heures ou jours, dans des systèmes HVDC, une condition d'équilibre de puissance doit être rétablie dans des délais très brefs (centaines de millisecondes). systèmes, ce qui est inacceptable [Gon18]. En particulier, compte tenu de la faible inertie du réseau, un effondrement de la tension continue se propage rapidement dans le réseau, provoquant une réponse presque simultanée de plusieurs systèmes de contrôle et de protection sur tout le système, interactions Par conséquent, en raison de la rapidité de réponse requise, un opérateur humain ne peut pas intervenir, comme c'est le cas pour les systèmes de transmission à courant alternatif. De plus, le contrôle local seul ne suffit pas pour ramener le système à une opération sécurisée et les actions de contrôle de sécurité appliquées par les différentes stations de conversion doivent être coordonnées.

Ainsi, alors que la coordination des actions de contrôle de sécurité repose sur la prise de décision des opérateurs humains dans les systèmes à courant alternatif, cela n'est plus possible dans les systèmes HVDC, qui doivent être restaurés après une panne dans une plage de temps courte. Ainsi, il est nécessaire de développer un contrôle par supervision automatisé qui remplace l'opérateur humain pour la surveillance en temps réel et la coordination des composants. Les exigences opérationnelles d'un système de transmission HVDC, comme pour les systèmes à courant alternatif traditionnels, sont satisfaites par la gestion des interactions entre la dynamique à événements discrets et à temps continu.

Des approches basées sur la modélisation DES et la synthèse des superviseurs ont été appliquées dans la littérature au fonctionnement des systèmes d'alimentation en courant alternatif. Les méthodes proposées ne conviennent toutefois pas à leur utilisation générale dans des systèmes réels. L'absence de méthode générique et formelle rend difficile l'application des solutions proposées en dehors des études de cas considérées, d'autant plus que le comportement physique des composants constituant le réseau n'est souvent pas parfaitement modélisé. De plus, seuls les contrôleurs centralisés étaient présentés dans la littérature, alors qu'un contrôle par supervision distribué définissant l'enveloppe d'exploitation à différents niveaux de détail serait plus approprié pour les systèmes de transmission de puissance géographiquement dispersés et exploités de manière hiérarchique.

Par conséquent, l'objectif de cette thèse est de proposer une méthode systématique de synthèse et de mise en oeuvre d'un contrôle par supervision pour les systèmes HVDC, basée sur la terminologie du DES et l'utilisation des méthodes formelles dans le cadre du SCT.

L'utilisation du SCT permet d'adapter l'architecture du contrôle par supervision à l'application correspondante. Par exemple, compte tenu de la nature distribuée des réseaux HVDC (stations de conversion réparties sur tout le réseau), leur structure hiérarchique (schémas de protection et de contrôle locaux interagissant avec un centre de contrôle) et les diverses situations rencontrées par le système restauration de l'équilibre de puissance, etc.), un contrôle par supervision

dédié doit être conçu par des approches de décomposition horizontale, verticale et modale. Concernant la décomposition verticale, l'approche compositionnelle [Moh14] est préférée car elle permet de définir plusieurs niveaux de contrôle lors de la phase de conception et n'est pas limitée par une structure prédéfinie. Cet aspect est particulièrement intéressant pour les systèmes HVDC. En effet, l'absence de réseaux MTDC existantes exige une méthode de conception flexible qui intègre de nouveaux aspects avec un impact minimal sur la structure existante, étant donné que des niveaux de contrôle supplémentaires pourraient être nécessaires à l'avenir à mesure que la compréhension du système augmente. En ce qui concerne la décomposition horizontale, l'approche décentralisée est privilégiée car elle permet d'appliquer les spécifications du système (impliquant des stations de conversion interconnectées et des phénomènes communs) au moyen de la communication. Cependant, une analyse approfondie des exigences de communication des systèmes HVDC doit être effectuée avant que la décomposition verticale du contrôle par supervision ne soit traitée, ce qui est hors de la portée de cette thèse. Ainsi, seules la décomposition verticale et modale [Far10] sont traitées dans notre travail.

Aussi, plusieurs méthodes sont proposées dans la littérature pour la mise en oeuvre pratique de l'architecture de contrôle par supervision synthétisée au moyen du SCT. Cependant, la quasi-totalité des travaux connexes sont appliqués au contrôle des systèmes de fabrication, ce qui peut être considéré comme du pur DES. Ainsi, les structures de contrôle proposées ne sont pas toujours adaptées au contrôle des systèmes hybrides, tels que les réseaux HVDC. De plus, les méthodes utilisées dans la littérature tentent d'adapter les solutions proposées aux langages de mise en oeuvre associés aux automates. Cependant, en raison de la coordination rapide requise par le contrôle par supervision d'un système HVDC, il est peu probable que les automates soient utilisés. En conséquence, une méthode d'implémentation qui adapte l'architecture de contrôle obtenue avec le SCT à la spécificité des systèmes HVDC est proposée dans cette thèse.

## Conception et mise en oeuvre du contrôle par supervision

En raison de la nature hybride des systèmes HVDC, le contrôleur discret doit interagir avec le système physique via une interface, comme illustré à la Figure 3. Cette interface traduit les mesures en temps continu reçues des capteurs en événements discrets. Inversement, il associe un ensemble de signaux envoyés aux boucles de contrôle à temps continu et aux relais de protection du système (qui imposent en fin de compte les actions de contrôle et de protection) aux commandes à événements discrets données par le contrôle par supervision. Ainsi, le contrôleur discret et l'interface dédiée constituent le contrôle par supervision, tandis que le système physique et les dispositifs de contrôle, de protection et de comptage associés constituent le procédé à superviser.

En conséquence, une approche systématique, illustrée dans la Figure 4 sous la forme d'un diagramme d'activité UML, pour la conception et la mise en oeuvre d'un contrôle de changement

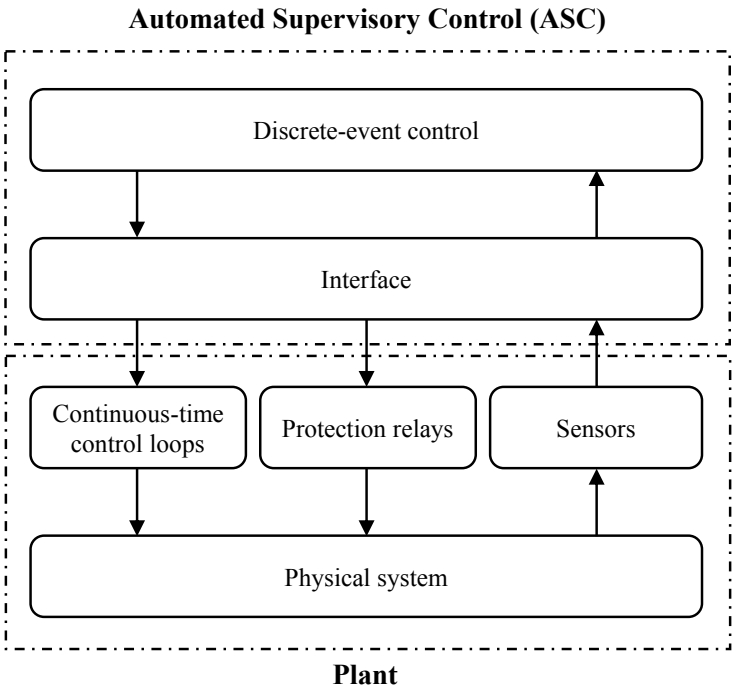



Figure 3 – Structure du contrôle par supervision dans un système hybride

de mode basé sur l'utilisation de la décomposition modale et compositionelle est utilisée dans notre travail.

Avant la conception du contrôle par supervision, il est nécessaire d'identifier le comportement des composants de station, ainsi que le comportement commutatif du système HVDC, au moyen d'une analyse fonctionnelle et de surveillance spécifique, puis de modéliser ces comportements sous la forme d'automates (étape 1 de la Figure 4). L'étape 2 concerne l'étude du comportement du système dans chaque mode par la modélisation de spécifications intramodales et la synthèse de superviseurs dédiés. A l'étape 2, le contrôle par supervision est décomposé horizontalement ou verticalement, comme indiqué dans [Rom17] pour l'étude de cas présentée. Puis, à l'étape 3, les automates obtenus à l'étape précédente sont étendus au moyen de spécifications intermodales afin de prendre en compte les commutations modélisées par l'automate de modes. Dans la dernière étape de la conception, les modèles obtenus sont réduits et les états non significatifs sont fusionnés pour montrer l'inactivité du mode. Enfin, ces modèles sont manipulés en vue de leur implémentation dans un logiciel dédié.

## Construction des modèles

### Modèles génériques de composant

Bien que le nombre de composants dans chaque station puisse varier en fonction de la configuration du réseau, le type de composants utilisé reste presque invariable pour tout système

HVDC à base de convertisseurs MMC. Ainsi, en termes de supervision, il est avantageux de considérer une approche orientée objet pour la modélisation du comportement du composant: le comportement inhérent au composant peut être considéré dans un modèle générique du composant, alors que le comportement lié à la configuration de le système ou une situation opérationnelle spécifique est traité dans une instance du modèle générique.

Donc, une analyse fonctionnelle et de surveillance est d'abord effectuée sur les composants de station afin d'identifier leur comportement générique (étape 1 de la Figure 4) . L'analyse fonctionnelle et de surveillance proposée vise à établir une méthode standardisée pour l'étude

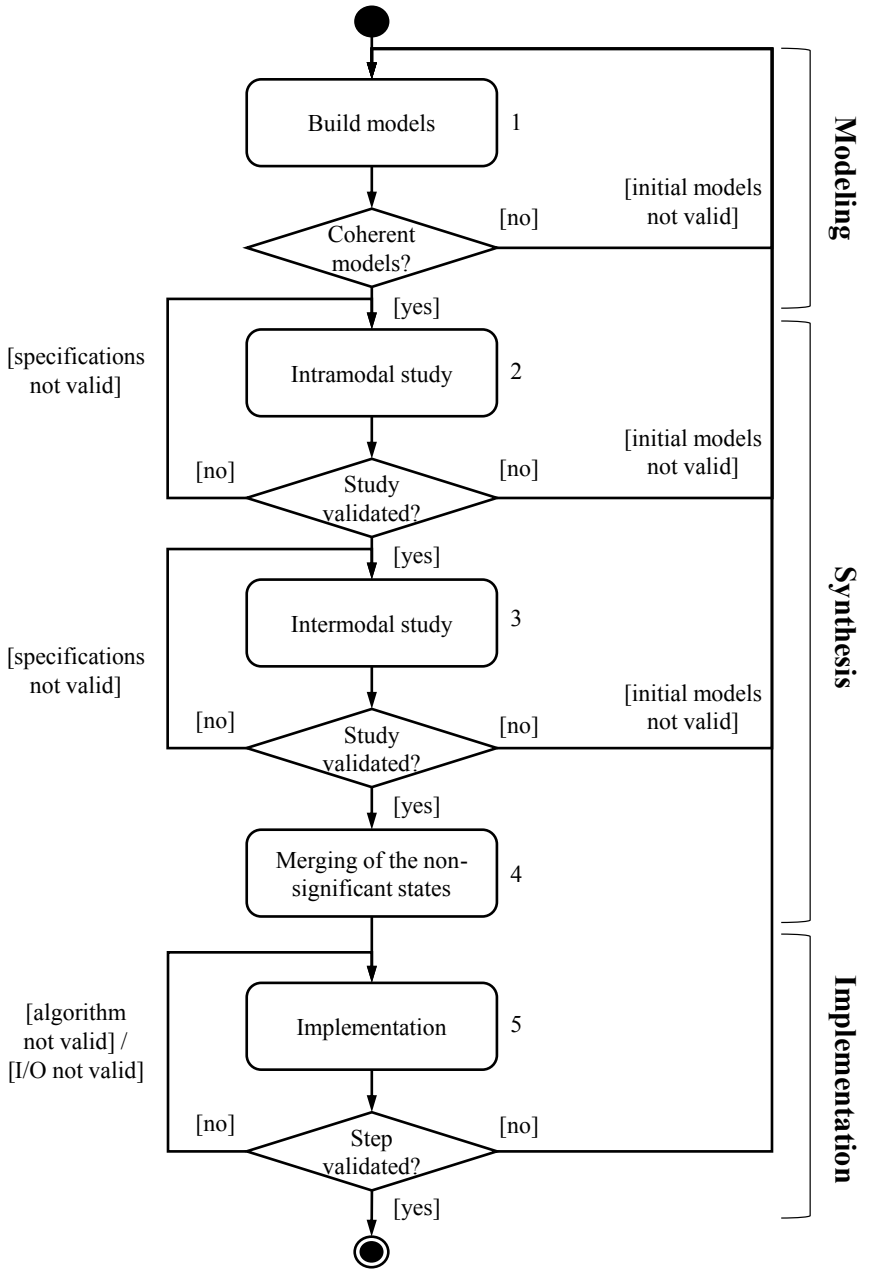



Figure 4 – Démarche proposée pour la conception et mise en oeuvre du contrôle par supervision

des composants futurs, permettant de définir un cadre de collaboration entre les ingénieurs système et le concepteur du contrôle par supervision. Les résultats de l'analyse sont ensuite utilisés pour la construction des modèles de composants génériques sous la forme d'automates.

Les modèles génériques des composants doivent être obtenus par la modélisation du comportement lié au contrôle et/ou le comportement physique de chaque composant $C_i$ dans une station, avec $i \in \mathbb{N}$. Le comportement lié à la commande doit être modélisé si nécessaire en associant un événement discret à chaque valeur qu'un signal numérique interagissant avec le contrôle par supervision peut prendre, comme décrit dans l'analyse fonctionnelle. En revanche, le comportement physique correspond à une évolution en temps continu des variables du système (tension, courant, etc.) identifiées dans l'analyse de surveillance. Ces dynamiques en temps continu peuvent être abstraites afin de modéliser les états qualitatifs sous-jacents et les événements correspondants en étiquetant les transitions entre eux [Vás16].

Le comportement générique de chaque composant $C_i$ est alors modélisé sous la forme d'un automate générique $G^{C_i}$ qui génère un langage $L(G^{C_i})$ couvrant les différents comportements du composant indépendamment du état de fonctionnement du système. Par définition, aucune condition de fonctionnement particulière n'est prise en compte lors de la modélisation du comportement générique d'un composant. En conséquence, tous les états de $G^{C_i}$ sont marqués. De plus, pour que l'automate génère un langage, le concepteur doit choisir un état initial lors de la modélisation du comportement générique de $C_i$. Cependant, l'ensemble $Q_m^{C_i}$ des états marqués, ainsi que l'état initial $q_0^{C_i}$, peuvent être modifiés ultérieurement par le concepteur lors de l'instantiation du modèle générique du composant dans une situation spécifique du système.

Ainsi, l'automate générique $G^{mmc}$ donné dans la Figure 5 représente le comportement générique du convertisseur sans tenir compte des conditions de fonctionnement du système, tel que $Q_m^{mmc} = Q^{mmc}$ et $\Sigma^{mmc} = \Sigma_{ctrl}^{mmc} \cup \Sigma_{ph}^{mmc}$. L'état initial $q_0^{mmc}$ est choisi par défaut pour être l'état où le convertisseur est bloqué et non chargé, mais cela peut être modifié en fonction de la situation du système.

Le comportement générique du câble, quant à lui, est modélisé par l'automate $G^{cable}$, montré dans la Figure 6. Le comportement physique du câble étant modélisé à des fins de surveillance,

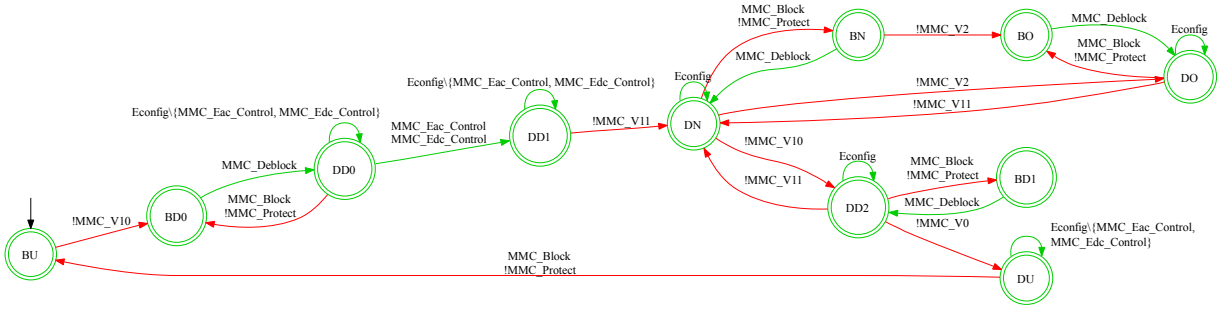



Figure 5 – Automate générique $G^{mmc}$ d'un MMC

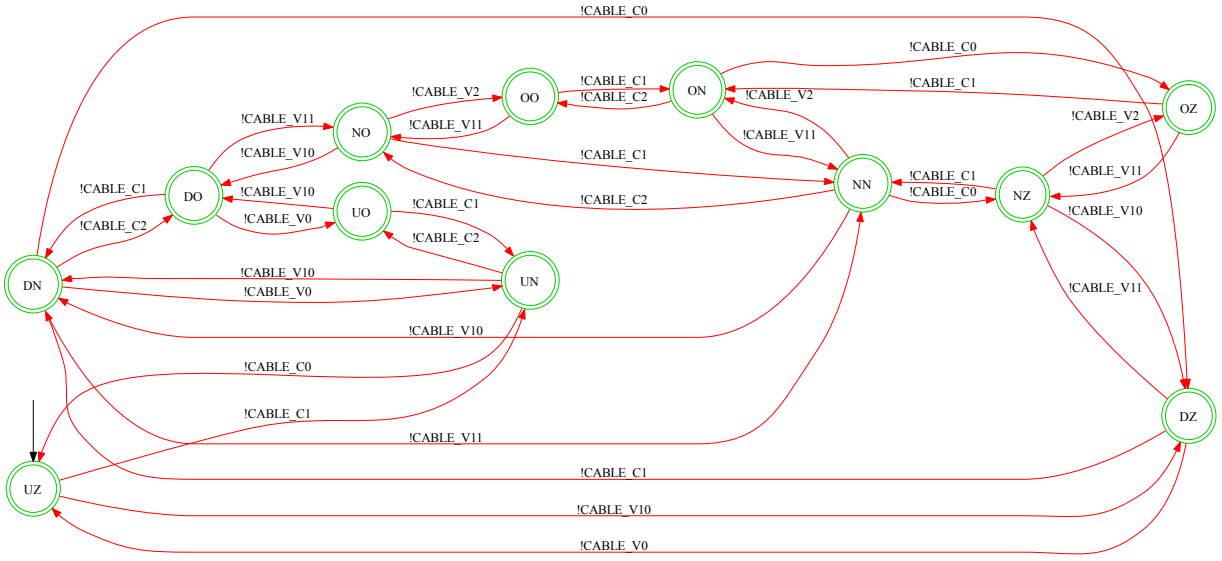



Figure 6 – Automate générique $G^{cable}$ d'un câble

tous les événements de $\Sigma^{cable}$ sont incontrôlables, car ils sont générés à partir des mesures du système, de sorte que $\Sigma^{cable} = \Sigma^{cble}_{v_{cble}} \cup \Sigma^{cble}_{i_{cble}}$, avec $\Sigma^{cble}_{c} = \emptyset$.

Finalement, il n'est pas dans la portée de cette thèse de modéliser en profondeur le comportement des procédés inclus dans le sous-système de protection, et nous considérons dans notre analyse fonctionnelle un comportement simplifié du disjoncteur et du relais de protection dédié.

Par conséquent, nous modélisons simplement dans l'automate générique $G^{cb}$ de la Figure 7 l'évolution de l'état d'un disjoncteur, qui peut être ouvert ou fermé, suite à un Commande d'ouverture ou de fermeture du relais de protection. Cependant, il est possible que le contrôle par supervision ou tout agent externe tel qu'un opérateur humain interagisse avec le relais et demande la fermeture ou l'ouverture du disjoncteur. Le relais analyse ensuite si les conditions de courant et de tension autour du disjoncteur endommageraient le dispositif physique ou non, et accepterait ou rejetterait la demande (c.-à-d. qu'il émettrait une commande d'ouverture ou de fermeture) en conséquence. Par conséquent, une demande n'implique pas nécessairement un changement de l'état du disjoncteur, tel que représenté par la transition de boucle automatique associée dans $G^{cb}$. Les événements Open et Close sont incontrôlables car ils sont supposés être générés de manière autonome par le relais de protection et ne peuvent donc pas être désactivés par un contrôleur externe; alors que les événements de requête RequestOpen et RequestClose

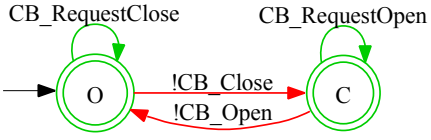



Figure 7 – Automate générique $G^{cb}$ d'un disjoncteur

sont contrôlables et générés par un agent de contrôle externe au sous-système de protection (tel que le contrôle par supervision).

Alors, $G^{cb}$ est défini tel que $\Sigma^{cb} = \{\text{Close}, \text{Open}, \text{RequestClose}, \text{RequestOpen}\}$, $Q^{cb} = \{\text{O}, \text{C}\}$, $Q_m^{cb} = Q^{cb}$ et $q_0^{cb} = \text{O}$, avec $\Sigma^{cb} = \Sigma_{uc}^{cb} \cup \Sigma_c^{cb}$ tel que $\Sigma_c^{cb} = \{\text{RequestClose}, \text{RequestOpen}\}$ et $\Sigma_{uc}^{cb} = \{\text{Close}, \text{Open}\}$.

Une fois les modèles génériques de composants validés, il est nécessaire de déterminer l'ordre séquentiel dans lequel un ensemble de blocs fonctionnels doit être agencé pour qu'une séquence complète d'actions de contrôle/protection puisse être réalisée au niveau du réseau.

**Automate de modes**

Dans cette thèse, une approche qui classe les états opérationnels du système en fonction de la capacité des composants du réseau à transférer la puissance et à maintenir l'équilibre de puissance est proposée. Dans le cas des réseaux HVDC basés sur MMC, ces composants seraient les MMC eux-mêmes et les câbles DC. La capacité de transférer la puissance de ces composants peut être identifiée à travers les régions d'opération définies pour la tension du MMC et la tension et le courant du câble. En effet, un MMC dont la tension est inférieure au seuil de contrôlabilité est incapable de transférer la puissance requise par le GRT. D'un autre côté, si un MMC subit une surtension, la probabilité de dommages augmente et sa capacité à transférer de l'énergie est donc compromise. On peut en dire autant de la tension du câble. Dans le cas du courant, si sa valeur est zéro, aucune puissance n'est transmise par le câble. De même, une surintensité implique généralement l'apparition d'un défaut de court-circuit et, par conséquent, aucune puissance n'est transférée non plus (à mesure que la tension dans les câbles s'effondre).

Ainsi, comme on peut le constater, les régions de fonctionnement des différentes variables de composant déterminent les conditions de fonctionnement et de sécurité du réseau. Ensuite, les états opérationnels du système sont définis comme l'ensemble des combinaisons obtenues via le produit cartésien entre les trois régions d'opération des variables identifiées (tension MMC, tension du câble et courant du câble). Étant donné que le nombre de combinaisons augmente de façon exponentielle avec le nombre de stations dans le réseau, on fait l'hypothèse que les mêmes types de composants dans toutes les stations partagent la même région de fonctionnement.

Cependant, certaines des combinaisons obtenues peuvent ne jamais être physiquement accessibles. Par conséquent, une analyse approfondie devrait être entreprise afin de les supprimer de la conception du contrôle par supervision. Une fois que les états opérationnels réalisables sont identifiés, une étude exhaustive doit être réalisée afin de définir comment les états opérationnels sont atteints les uns des autres. L'ensemble des événements qui provoquent un changement d'état peut soit correspondre à des actions de contrôle et de protection, soit à des perturbations externes au système de contrôle. A titre d'illustration, certaines des actions et perturbations qui modifient quelques états opérationnels d'un système HVDC sont présentées dans la Figure 8.

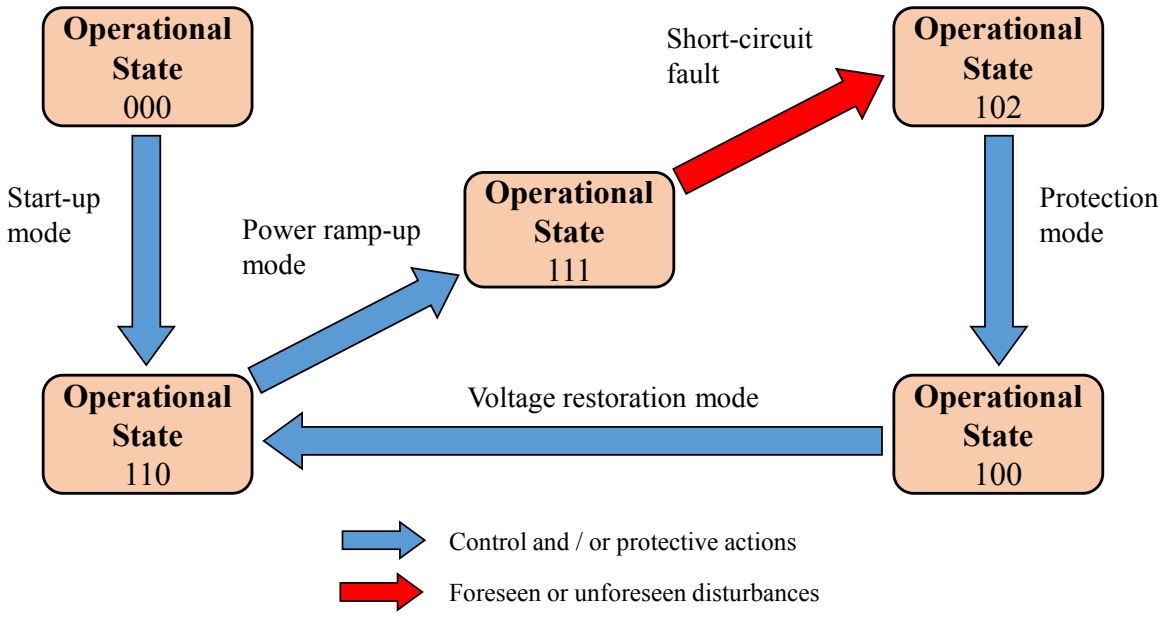



Figure 8 – Quelques états opérationnels d'un système HVDC

Chaque flèche bleue de la Figure 8 est une représentation abstraite des séquences diverses et détaillées d'actions de contrôle et de protection qui commencent à un état opérationnel donné par les experts du système. Ainsi, cette représentation abstraite, appelée mode opérationnel, regroupe sous un même label toutes les différentes procédures opérationnelles qui commencent et se terminent au même état opérationnel, c'est-à-dire qu'elles correspondent au même objectif opérationnel.

Tous les modes du système doivent être définis dans un ensemble $\mathcal{M} = \{M_0, M_1, \ldots, M_j\}$ ($j \in \mathbb{N}$), où $M_j$ est le nom du mode considéré donné par le concepteur afin d'améliorer l'interprétation des modèles. Les modes d'un système HVDC peuvent être naturellement définis comme les modes opérationnels identifiés dans la Figure 8. De plus, les états opérationnels de la Figure 8 peuvent être considérés comme *modes statiques* du point de vue de la modélisation, étant donné qu'ils manquent fréquemment d'un comportement interne; en opposition aux modes opérationnels, qui sont considérés comme des *modes dynamiques*. Ainsi, les modes considérés sont listés dans le tableau suivant:

| **Name** | **Mode** |
|---|---|
| $M_1$ | OS000 |
| $M_2$ | OM1: Mise sous tension |
| $M_3$ | OS110 |
| $M_4$ | OM2: Mise sous puissance |
| $M_5$ | OS111 |
| $M_6$ | OS102 |
| $M_7$ | OM3: Protection |
| $M_8$ | OS100 |
| $M_9$ | OM4: Restoration |

Tableau 3 – Modes du système considérés

Le système HVDC évolue alors par l'alternance des modes dans le Tableau 3, de sorte qu'un seul $M_j(j = \{1,\ldots,9\})$ est actif à la fois. Le comportement commutatif entre les modes du système est alors modélisé par le concepteur sous la forme d'un automate de modes donné dans la Figure 9. L'état initial correspond à $M_1$, où tous les MMC et les câbles sont déchargés et où aucun courant ne circule à travers eux. Au début, il est nécessaire d'activer le mode de démarrage ($M_2$) pour alimenter les MMC et les câbles ($M_3$). Deuxièmement, le mode de rampe d'alimentation ($M_4$) doit être démarré pour réaliser le flux de puissance souhaité. Ensuite, le mode nominal $M_5$ est atteint, ce qui correspond à l'état marqué de l'automate. Si un défaut de court-circuit incontrôlable se produit, une surintensité apparaît et la tension dans les câbles s'effondre ($M_6$). Il est donc nécessaire d'activer d'abord les modes de protection ($M_7$), puis de restauration de la tension ($M_9$) une fois la panne supprimée ($M_8$), afin de revenir à un état où les câbles sont sous tension et le défaut est supprimé du réseau ($M_3$) Enfin, $M_4$ est redémarré pour rétablir un flux d'énergie dans le système et atteindre le mode marqué $M_5$.

L'ensemble des événements de commutation $\Sigma^{\mathcal{M}}$ est formé soit par des événements contrôlables correspondant au début et à la fin d'un mode donné (démarrage, protection ...), soit par des événements incontrôlables indiquant l'apparition d'un défaut (par exemple un défaut de court-circuit). Dans tous les cas, les événements de commutation étant associés à l'atteinte d'une configuration particulière par l'ensemble des composants du réseau, ils ne sont inclus dans aucun alphabet des composants, contrairement à [Far10]. En conséquence, $\Sigma^{\mathcal{M}} \bigcap (\bigcup_{i=1}^{n} \Sigma^{C_i}) = \emptyset$, étant donné qu'aucun composant ne génère un événement de commutation, c.-à-d. $\bigcup_{i=1}^{n} \Sigma_{\rightleftarrows}^{C_i} = \emptyset$. Par conséquent, l'ensemble $\mathcal{C}^{M_j}$ des composants utilisés dans un mode $M_j$ est entièrement inclus dans l'ensemble $\mathcal{C}_{\circlearrowleft}^{M_j}$ des composants représentant le comportement intramodal du système dans le mode $M_j$, avec l'ensemble des composants qui génèrent un

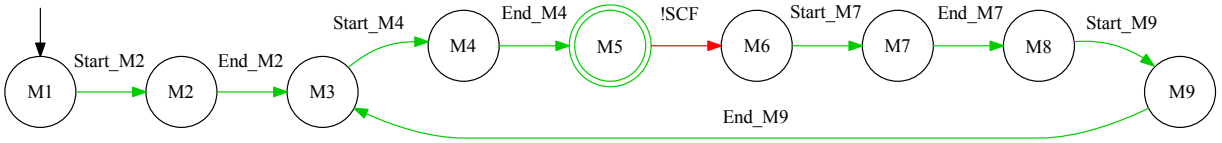



Figure 9 – Automate de modes $G^{\mathcal{M}}$

événement de commutation $\mathcal{C}^{M_j}_{\rightleftarrows} = \mathcal{C}^{M_j}_{\leftarrow} \cup \mathcal{C}^{M_j}_{\rightarrow} = \emptyset$.

### Vérification de la cohérence entre modèles

Comme l'ensemble $\mathcal{M}$ est intégré dans la définition de $G^{\mathcal{M}}$, ces deux modèles sont nécessairement cohérents l'un par rapport à l'autre. Par contre, les relations entre les composants $G^{C_i,M_j}$, les ensembles $\mathcal{C}^{M_j}_{\circlearrowleft}$ des composants dans un mode et $G^{\mathcal{M}}$ ne sont pas formellement définis. Il est donc nécessaire de vérifier que chaque modèle n'est pas en contradiction avec les autres. Les conditions suivantes répertorient les propriétés à respecter pour valider cette cohérence:

- *Condition 1*: Pour tout mode $M_j$ de $\mathcal{M}$ sans comportement interne, l'ensemble des composants est vide.
- *Condition 2*: Pour tout modèle $G^{C_i,M_j}$ d'une composante $C_i$ dans un mode $M_j$, tel que $C_i \in \mathcal{C}^{M_j}_{\circlearrowleft}$, l'état initial $q_0^{C_i,M_j}$ de $G^{C_i,M_j}$ est cohérent avec le mode à partir duquel $M_j$ est atteint, et les états marqués dans $Q_m^{C_i,M_j}$ sont cohérents avec le mode atteint à partir de $M_j$.

La cohérence entre les états initial et marqué de $G^{C_i,M_j}$ et les modes dans $G^{\,mathcalM}$ signifie que le statut du composant $C_i$ en mode $M_j$ est en conformément au mode à partir duquel $q_0^{C_i,M_j}$ est atteint ou ceux obtenus à partir des états dans $Q_m^{C_i,M_j}$. Cette cohérence est vérifiée au moyen du suivi et de l'analyse fonctionnelle du concepteur.

Dans le cas où toutes ces conditions sont respectées, la construction des modèles a été bien menée et la propriété de cohérence est validée. Cela signifie que chaque mode avec un comportement interne contient un ensemble de composants (Condition 1) et que les états initial et marqué des modèles de composants dans un mode sont cohérents avec les modes avant et après le mode étudié (Condition 2).

Dans le cas contraire, le concepteur peut identifier le problème en fonction des conditions non respectées et modifier tout ou partie des modèles précédemment construits. Ainsi, si la Condition 1 n'est pas respectée, les états de $G^{\mathcal{M}}$ sont mal définis dans l'analyse de surveillance et fonctionnelle. Enfin, si la Condition 2 est violée, alors les automates $G^{C_i,M_j}$ ont été mal construits.

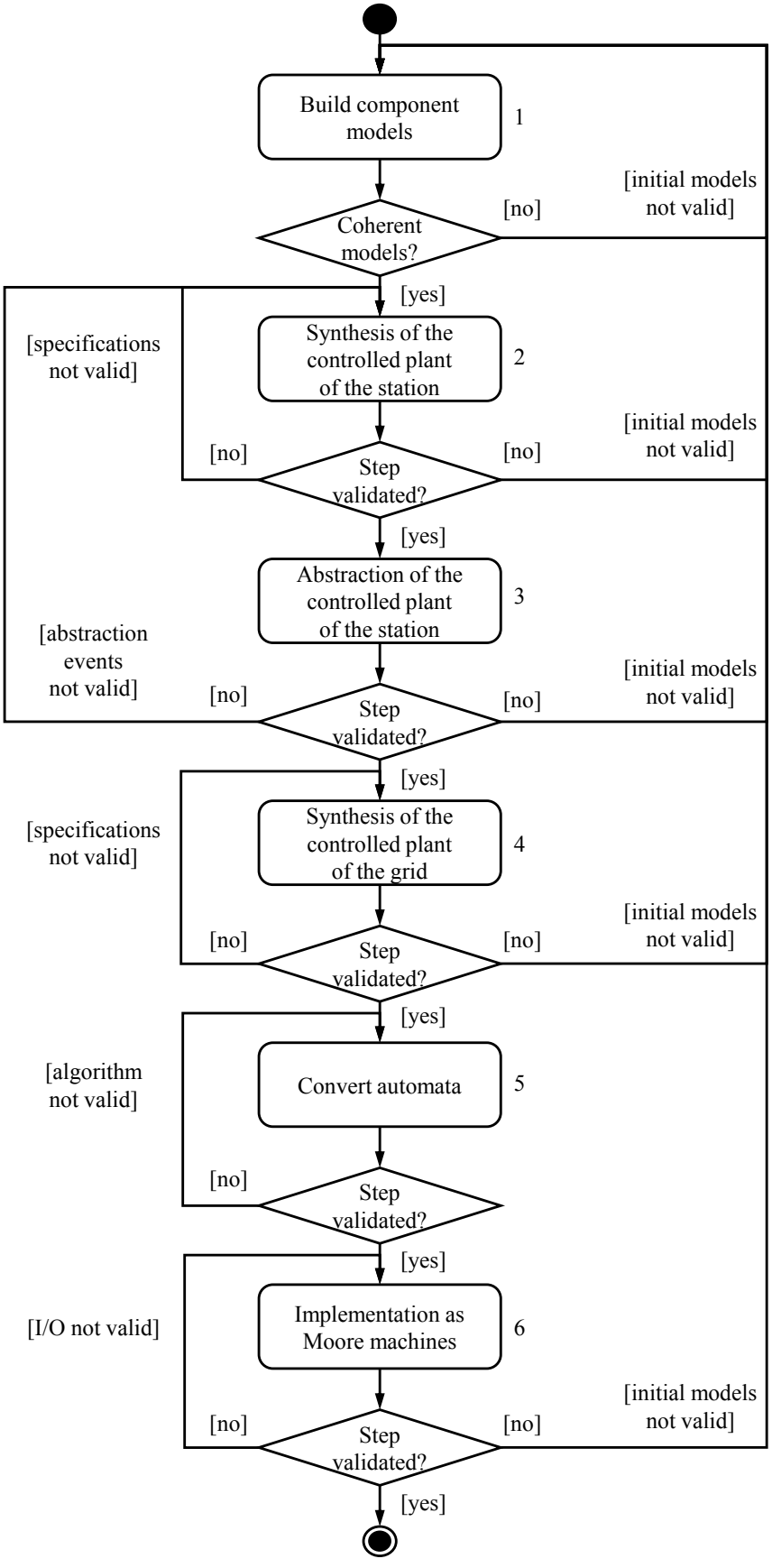



Figure 10 – Démarche proposée pour l'étude intramodale

## Étude intramodale

La démarche proposée pour l'étude intramodale, illustrée dans la Figure 10 sous la forme d'un diagramme d'activité UML, est basée sur l'utilisation d'une approche compositionnelle qui construit les modèles nécessaires à différents niveaux d'abstraction. Étant donné qu'un réseau HVDC comprend un ensemble de stations réparties géographiquement, chacune comprenant à son tour un ensemble de composants différents, le procédé globale du réseau entière est représentée par un ensemble d'automates $\mathcal{G}$, qui inclut chaque composant. A ce stade, le triple de synthèse original est défini comme étant $(\mathcal{G};\mathcal{S};\mathrm{id})$. Puisque l'alphabet des composants d'un réseau est exclusif à chaque composant, il n'existe aucun événement partagé entre eux, c'est-à-dire que $\Omega = \bigcap_{i=1}^{n} \Sigma^{C_i} = \emptyset$. Cependant, du fait de la proximité géographique, des phénomènes électromagnétiques communs apparaissent entre les composants, notamment entre ceux-ci dans une même station.

Par conséquent, en ce qui concerne la conception du contrôle par supervision, car il n'existe pas d'événements communs entre les composants du système, il est avantageux de composer d'abord les installations de chaque station pour que de nouveaux événements locaux puissent être abstraits de l'automate résultant. Les modèles abstraits peuvent alors être utilisés pour obtenir le comportement contrôlé du réseau. Cette approche compositionnelle rend les modèles obtenus tels que seuls les comportements pertinents sont considérés à chaque niveau de détail. Dans cette thèse, toutes les stations du système sont considérées comme identiques dans leur configuration (nombre et type de composants), de sorte que seul le comportement contrôlé d'une station est considéré sans perte de généralité.

La première étape concerne la restriction du comportement générique des composants dans une station. L'étape 2 concerne la construction du procédé contrôlée d'une station par synthèse, telle que définie dans le SCT. Au cours de l'étape 3, les modèles obtenus sont abstraits selon l'approche compositionnelle, de sorte que les modèles abstraits ont une taille réduite tout en étant équivalents d'un point de vue de la supervision. Ces modèles réduits sont ensuite combinés à l'étape 4 afin de modéliser le procédé du réseau et de synthétiser son homologue contrôlé.

### Procédé sous contrôle de la station

Premièrement, il est nécessaire de restreindre le langage $L(G^{C_i})$ généré par le modèle de type de composant générique $G^{C_i}$ au langage $L(G^{C_i,M_j})$ généré par le modèle particulier $G^{C_i,M_j}$ d'un composant dans une procédure opérationnelle $M_j$ $(j = 1,\ldots,n)$. Ainsi, l'automate générique d'un type de composant est d'abord instancié autant de fois qu'il y a de composants de ce type dans la station.

Ensuite, les comportements du composant n'apparaissant pas dans la procédure étudiée ne sont pas pris en compte, c'est-à-dire que les événements non nécessaires à la conception sont supprimés via l'opération de projection l'automate résultant est coupé afin de supprimer les

états de blocage. Comme l'alphabet de chaque composant est spécifique à chaque composant, les événements de l'automate générique sont renommés de manière à rendre visible le nom du composant particulier. Un index est également ajouté aux événements si plusieurs composants du même type sont situés du même côté de la station. De plus, le concepteur doit sélectionner les états initial et marqué du composant $C_i$ en fonction des états initial et final de $M_j$.

Le fonctionnement des composants de la station pendant une procédure doit cependant respecter un ensemble de contraintes physiques élémentaires qui doivent être explicitement modélisées. Ces contraintes physiques peuvent être liées soit à la réduction du comportement d'un composant dans $M_j$, soit aux contraintes imposées par le fonctionnement simultané de plusieurs composants au sein de la station. Différents types de contraintes sont distingués au moyen de la décomposition du flux de contrôle présentée dans [Fen06]. Par conséquent, si un automate de contrainte élémentaire $G_{pc_k}^{C_i}$ $(k = 1,\ldots,n)$ partage des événements avec un seul modèle de composant $G^{C_i,M_j}$, il représente un contrainte sur le comportement dudit composant pendant $M_j$. Sinon, il représente une contrainte sur le comportement simultané de plusieurs composants et est noté $G_{pc_k}^{st}$. Le procédé non contrôlé de la station en $M_j$ est alors obtenu via la composition parallèle des différents automates, tel que $G^{st,M_j} = (\parallel G^{C_i,M_j}) \parallel (\parallel G_{pc_k}^{C_i}) \parallel (\parallel G_{pc_k}^{st})$, avec $\Sigma^{st,M_j} = \bigcup_{i=1}^{n} \Sigma^{C_i,M_j}$ et $Q^{st,M_j} = (\bigtimes_{i=1}^{n} Q^{C_i,M_j}) \times (\bigtimes_{k=1}^{n} Q_{pc_k}^{C_i}) \times (\bigtimes_{k=1}^{n} Q_{pc_k}^{st})$.

A ce stade, le triple de synthèse est toujours défini comme $(\mathcal{G};\mathcal{S};\mathrm{id})$. Cependant, le procédé globale $\mathcal{G}$ a été réorganisée comme l'ensemble comprenant les automates $G^{st,M_j}$ de chaque station.

Le modèle $G^{st,M_j}$ obtenu à l'étape précédente préserve l'équivalence de synthèse mais doit être restreint pour respecter les spécifications imposées par le mode aux composants du système (et par conséquent à la station). Il est donc avantageux de construire le modèle de la station contrôlée par synthèse monolithique à ce stade, car un modèle réduit est plus facile à manipuler et que l'équivalence de synthèse est préservée [Moh14].

Ainsi, en plus du modèle d'procédé, il est nécessaire de disposer d'un modèle de cahier des charges pour effectuer une synthèse et obtenir le procédé contrôlée de la station. Le SCT permet de construire ce modèle de spécifications par la composition parallèle des modèles représentant chaque spécification individuelle. Donc, si nous appelons $E^{st,M_j}$ le modèle des spécifications à respecter dans $M_j$, les modèles $E_l^{st,M_j}$ (avec $l \in \mathbb{N}$) représente les spécifications élémentaires.

Ensuite, une fois que le modèle de la station non contrôlée $G^{st,M_j}$ et le modèle de spécification $E^{st,M_j}$ sont obtenus, nous pouvons construire le procédé contrôlée $H^{st,M_j}$ de la station pendant le mode $M_j$: $H^{st,M_j} = (Y^{st,M_j}, \Sigma^{st,M_j}, \tau^{st,M_j}, y_0^{st,M_j}, Y_m^{st,M_j})$ tel que $L_m(H^{st,M_j}) = [L_m(G^{st,M_j} \parallel E^{st,M_j})]^{\uparrow c}$.

Pour conclure, le triple de synthèse est maintenant défini comme $(\mathcal{G};\mathcal{S};\mathrm{id})$. A ce stade, qui correspond à la fin de l'étape 2 de la Figure 10, l'ensemble $\mathcal{S}$ est formé par les procédés contrôlées $H_i^{st,M_j}$ de chaque station ($i \in \{1,\ldots,n\}$, $n$ étant le nombre de stations composant le réseau). Concernant $\mathcal{G}$, l'ensemble n'est pas vide et est composé de la même manière par les au-

tomates $H_i^{st,M_j}$ de chaque station, car il est toujours nécessaire de restreindre le fonctionnement simultané des différentes stations, cette fois au niveau du réseau.

Les automates résultant de la synthèse à chaque station peuvent cependant être de grande taille, ce qui les rend difficiles à manipuler et à combiner au niveau du réseau. Malgré le fait que $\bigcap_{i=1}^{n} \Sigma^{C_i,M_j} = \emptyset$, il est possible pour le concepteur d'abstraire de $G^{st,M_j}$ *a posteriori* les séquences d'événements qui représentent des phénomènes physiques, communs à plusieurs composants. À cette fin, il est nécessaire d'identifier les concaténations d'événements qui partent d'un état unique $q_0^{abs}$ et atteignent un état unique $q_n^{abs}$ à la fin de la chaîne, avec un unique occurrence de tous les événements de la chaîne sans ordre particulier, comme défini ci-dessous:

L'importance de telles chaînes repose sur l'occurrence de tous les événements qui y sont compris, et pas tant sur leur ordre d'apparition. En conséquence, chacune de ces séquences peut être abstraite en introduisant une fin de chaîne" événement $\sigma^{abs}$ indiquant la fin de $s_{abs}$, telle que $\delta(q_n^{abs}, \sigma^{abs}) \in Q^{st,M_j}$. Ces événements sont inclus dans l'ensemble $\Sigma^{abs}$ et ne servent qu'à des fins de conception, car ils n'ont aucune relation physique avec aucun des composants de la station.

La contrôlabilité de $\sigma^{abs}$ doit être définie par le concepteur en fonction de la contrôlabilité des événements dans la chaîne: si tous les événements de $s^{abs}$ sont incontrôlables, alors $\sigma^{abs}$ est également incontrôlable; d'autre part, si tous les événements de $s^{abs}$ sont contrôlables, $\sigma^{abs}$ est défini pour être contrôlable. Enfin, si certains des événements contenus dans une chaîne pouvant être analysée sont contrôlables alors que les autres sont incontrôlables, la possibilité de contrôler $\sigma^{abs}$ doit être définie par le concepteur en fonction de la signification physique de la chaîne. Par exemple, un événement incontrôlable pourrait dépendre de l'occurrence d'un événement contrôlable précédent. Par conséquent, la chaîne complète peut être considérée comme contrôlable et l'événement d'abstraction défini comme tel. Les relations entre les événements de $\Sigma^{abs}$ et les événements spécifiques au composant sont ensuite modélisées sous la forme d'automates élémentaires, notés $G_k^{abs}$. le procédé contrôlée de la station $H^{st,M_j}$ est alors à nouveau synthétisée, telle que $L_m(H^{st,M_j}) = [L_m(G^{st,M_j} \parallel E^{st,M_j})]^{\uparrow c}$, avec $G^{st,M_j} = (\parallel G^{C_i,M_j}) \parallel (\parallel G_{pc_k}^{C_i}) \parallel (\parallel G_{pc_k}^{st}) \parallel (\parallel G_k^{abs})$.

Ensuite, parce que les événements d'abstraction introduits lors de la synthèse de le procédé contrôlée de la station $H^{st,M_j}$ indiquent la fin d'une séquence d'événements, indépendamment de l'ordre d'apparition, tous les événements inclus dans la chaîne attachée peut être projeté à partir de $H^{st,M_j}$ tout en préservant l'équivalence de synthèse. En effet, ces événements peuvent maintenant être considérés comme locaux à chaque station et donc seuls les événements d'abstraction sont conservés au niveau du réseau (étape 3 de la Figure 10). Bien qu'il n'existe pas d'événements partagés entre les modèles de la station ($\Omega = \emptyset$), les autres événements de la station sont nécessaires pour la supervision et la coordination des stations, contrairement aux événements projetés sur $H^{st,M_j}$. Par conséquent, l'alphabet du modèle abstrait $\tilde{H}^{st,M_j}$ est obtenu au moyen de la projection $P_{\Sigma\tilde{\Sigma}}\ddot{\Sigma}^{st,M_j} \rightarrow \tilde{\Sigma}^{st,M_j}$. Une fois l'abstraction des

modèles de station réalisée, le triple de synthèse est défini comme étant égal à $(\mathcal{G};\mathcal{S};\mathrm{id})$, avec $\mathcal{G} = \{\tilde{H}_1^{st,M_j}, \tilde{H}_2^{st,M_j}, \ldots, \tilde{H}_n^{st,M_j}\}$ et $\mathcal{S} = \{H_1^{st,M_j}, H_2^{st,M_j}, \ldots, H_n^{st,M_j}\}$.

**Procédé sous contrôle du réseau**

À ce stade, comme nous considérons que les stations partagent le même type et le même nombre de composants, le modèle abstrait du procédé contrôlée des stations est instancié autant de fois qu'il y a de stations dans le réseau et ses événements sont renommés pour refléter au moyen d'un index à quelle station ils correspondent (cela pourrait ne pas être nécessaire dans le cas où les stations ne sont pas identiques). Par conséquent, il n'existe pas d'événements partagés entre les modèles réduits des stations, et la seule méthode permettant de simplifier davantage les automates considérés est la synthèse monolithique. Avant cela, il est nécessaire de modéliser les contraintes physiques imposées par l'interconnexion des différentes stations [Rom17]. Chaque contrainte physique individuelle est modélisée sous la forme d'un automate, noté $G_{pc_k}^{gr}$. Le procédé incontrôlée $G^{gr,M_j}$ du système au niveau du réseau lors de la procédure $M_j$ est alors défini tel que $G^{gr,M_j} = (\parallel G_{pc_k}^{gr}) \parallel (\parallel \tilde{H}_{st_i}^{M_j})$.

Outre le modèle de procédé, il est nécessaire de disposer d'un modèle de cahier des charges pour effectuer une synthèse et obtenir le procédé contrôlée du réseau. Le SCT permet de construire ce modèle de spécifications par la composition parallèle des modèles représentant chaque spécification individuelle. Donc, si nous appelons $E^{gr,M_j}$ le modèle des spécifications à respecter dans $M_j$, les modèles $E_l^{gr,M_j}$ ($l \in \mathbb{N}$) représente les spécifications individuelles.

Ensuite, une fois que les modèles et les spécifications du procédé au cours du mode considérée sont obtenus, nous pouvons construire le procédé contrôlée $H^{gr,M_j}$ du réseau dans $M_j$: $H^{gr,M_j} = (Y^{gr,M_j}, \Sigma^{gr,M_j}, \tau^{gr,M_j}, y_0^{gr,M_j}, Y_m^{gr,M_j})$ tel que $L_m(H^{gr,M_j}) = [L_m(G^{gr,M_j} \parallel E^{gr,M_j})]^{\uparrow c}$.

A la fin de cette étape, qui correspond à l'étape 4 de la Figure 10, le résultat de synthèse du triple de synthèse considéré est obtenu en composant le superviseur monolithique représentant le suprème sous-langage contrôlable pour les autres procédés, $\sup\mathcal{C}(\mathcal{G})$, avec les superviseurs collectés dans $\mathcal{S}$. Ceci peut être défini par $(\emptyset;\mathcal{S} \cup \{\sup\mathcal{C}(\mathcal{G})\};\mathrm{id})$.

**Validation de l'étude intramodale**

Lors de la conception du contrôle par supervision, le travail principal du concepteur consiste à modéliser les spécifications et les contraintes physiques du système aux différents niveaux de synthèse, ainsi que l'identification des chaînes pouvant être abstraites afin de simplifier le plus possible les modèles.

Dans le cas où le procédé contrôlée (au niveau de la station ou du réseau) dans une procédure $M_j$ n'existe pas, cela peut être dû au fait que les spécifications $E_l^{M_j}$ sont trop restrictives, à un mauvaise modélisation de la dynamique du composant particulier dans la procédure via les automates $G^{C_i,M_j}$, ou impossibilité de contrôler le procédé. Généralement, si le concepteur

a correctement formalisé le cahier des charges et la dynamique des composants, le manque d'installation contrôlée entraîne une modification du cahier des charges de $M_j$, ce qui conduira à une nouvelle étude de synthèse pour la procédure opérationnelle concernée. Il est également possible que les modifications souhaitées soient liées à la dynamique d'un composant $C_i$, auquel cas il s'agit de l'étape initiale de la modélisation à revoir, ainsi que des étapes de synthèse qui en découlent. Enfin, si les installations contrôlées ont le comportement attendu et sont validées par tous les collaborateurs, nous sommes certains que les spécifications sont respectées par définition.

Cependant, indépendamment des modifications souhaitées, la synthèse d'un contrôle par supervision pour une procédure nécessite soit un concepteur ayant une connaissance approfondie du système, soit une coopération étroite entre le concepteur et les ingénieurs experts. D'où l'importance du suivi et de l'analyse fonctionnelle antérieurs qui aident à structurer le travail de chaque collaborateur.

## Etude intermodale

L'objectif de l'étude intermodale, la troisième étape de la Figure 4, est d'étendre les modèles internes à un mode pour prendre en compte toutes les commutations possibles et leur impact sur la dynamique des procédés, tout en assurant que les spécifications intermodales sont respectées. Pour cela, le concepteur doit modéliser les spécifications de changement de mode avec les modes automates, afin que le comportement commutatif soit pris en compte.

Les modes d'un système HVDC sont modélisés par l'automate $G^{\mathcal{M}}$ dans la Figure 9. Étant donné que les spécifications intramodales sont construites exclusivement sur les ensembles d'événements générés par les composants utilisés dans les modes, il est nécessaire de définir des spécifications intermodales pour coupler les alphabets internes $\Sigma_{in,i}^{st,M_j}$ et $\Sigma_{in}^{gr,M_j}$ des procédés contrôlées dans un mode et l'ensemble $\Sigma^{\mathcal{M}}$ des événements de commutation de $G^{\mathcal{M}}$, comme $\Sigma^{\mathcal{M}} \cap (\cup_{i=1}^{n} \Sigma_{in,}^{st,M_j}) \cup \Sigma_{in}^{gr,M_j}) = \emptyset$.

En conséquence, un nouveau modèle $E_{\rightleftarrows}^{M_j}$ représentant les spécifications de changement de mode est ajouté à cette étape. Ce modèle permet de placer le langage des procédés contrôlées en interne dans le mode correspondant des modes automates. Formellement, le modèle de spécification $E_{\rightleftarrows}^{M_j}$ est construit à partir de la composition parallèle des automates de spécification élémentaires $E_{\rightleftarrows}^{l,M_j}$, tels que $E_{\rightleftarrows}^{M_j}$ est défini par $E_{\rightleftarrows}^{M_j} = (X_{\rightleftarrows}^{M_j}, \Sigma_{\rightleftarrows}^{M_j}, x_{\rightleftarrows,i}^{M_j}, x_{0,\rightleftarrows}^{M_j}, X_{m,\rightleftarrows}^{M_j})$ et $E_{\rightleftarrows}^{M_j} = \|_l E_{\rightleftarrows}^{l,M_j}$. Dans le cas où la commutation de mode implique un comportement interne incontrôlable, la dynamique de commutation est naturellement modélisée par un automate végétal, noté $G_{\rightleftarrows}^{M_j} = \|_{l\in\mathbb{N}} G_{\rightleftarrows}^{l,M_j}$.

Le concepteur dispose alors de le procédé abstraite de la station et de le procédé contrôlée du réseau, désignées respectivement par $\tilde{H}_{in,i}^{st,M_j}$ et $H_{in}^{gr,M_j}$, les modes automates $G^{\mathcal{M}}$ et les modèles des spécifications intermodales $E_{\rightleftarrows}^{M_j}$ et $G_{\rightleftarrows}^{M_j}$ . Cependant, au niveau de la station, les modèles qui

étendent les transitions de commutation de mode au comportement interne ne sont basés que sur l'alphabet local de la station. En conséquence, ils sont respectivement notés $E^{M_j}_{\rightleftarrows,i}$ et $G^{M_j}_{\rightleftarrows,i}$.

$\tilde{H}^{st,M_j}_{in,i}$ et $H^{gr,M_j}_{in}$ peuvent alors être respectivement étendus par synthèse aux modèles $H^{M_j}_{st,i}$ et $H^{M_j}_{gr}$, qui considèrent le comportement commutatif du système comme suit:
$H^{M_j}_{st,i} = (Y^{M_j}_{st,i}, \Sigma^{M_j}_{st,i}, \tau^{M_j}_{st,i}, y^{M_j}_{0,st,i}, Y^{M_j}_{m,st,i})$ tel que $L_m(H^{M_j}_{st,i}) = [L_m(H^{M_j}_{in,st,i} \parallel G^{\mathcal{M}} \parallel E^{M_j}_{\rightleftarrows,i}) \parallel G^{M_j}_{\rightleftarrows,i})]^{\uparrow c}$;
$H^{M_j}_{gr} = (Y^{M_j}_{gr}, \Sigma^{M_j}_{gr}, \tau^{M_j}_{gr}, y^{M_j}_{0,gr}, Y^{M_j}_{m,gr})$ tel que $L_m(H^{M_j}_{gr}) = [L_m(H^{M_j}_{in,gr} \parallel G^{\mathcal{M}} \parallel E^{M_j}_{\rightleftarrows} \parallel G^{M_j}_{\rightleftarrows})]^{\uparrow c}$.

Le concepteur a encore une fois l'occasion de valider les modèles construits. Comme lors de la phase de validation de l'étude intramodale, les deux questions qui se posent concernent l'existence de modèles $H^{M_j}$ et le comportement qu'ils sont censés modéliser. Si les modèles $H^{M_j}$ n'existent pas, il faudra revoir les spécifications de commutation car ce sont surtout ces spécifications qui peuvent poser problème, comme les modèles $G^{C_i,M_j}$ et $G^{\mathcal{M}}$ ont déjà été validés. En pratique, ces spécifications sont très simples à écrire et il est donc peu probable que, si ces contraintes empêchent effectivement les commutations, le modèle de gestion de l'ensemble du mode doit être revu. Dans le cas où tous les modèles $H^{M_j}$ existent, il est possible encore une fois que le comportement de le procédé contrôlé soit trop restrictif. Cette restriction ne peut être causée que par les spécifications de commutation. Il est donc important que le concepteur identifie les comportements problématiques et fournisse des spécifications pour éviter les comportements indésirables afin d'obtenir un comportement de procédé contrôlé satisfaisant.

Bien que la preuve de l'équivalence entre l'approche de conception proposée et la méthode centralisée soit hors de la portée de cette thèse, certaines lignes directrices sont données ci-après.

Parce que l'alphabet interne à un mode et l'ensemble des événements de commutation sont disjoints, les automates de spécification intramodaux, ainsi que les installations de contraintes physiques, doivent être étendus dans l'approche centralisée afin de prendre en compte le comportement commutatif. Ces modèles étendus, obtenus par la composition parallèle des automates étendus élémentaires, peuvent être respectivement désignés par $E_{\circlearrowleft}$ et $G_{\circlearrowleft,pc}$. L'approche compositionnelle n'étant plus utilisée, les automates impliquant les événements d'abstraction ne seraient pas nécessaires.

En résumé, le procédé globale contrôlé $H = (Q, \Sigma, \delta, q_0, Q_m)$ obtenue par synthèse monolithique serait calculée de telle manière que:

$$L_m(H) = [L_m(G \parallel E \parallel G^{\mathcal{M}})]^{\uparrow c},$$

avec $\Sigma = (\cup_{i=1}^{n} \Sigma^{C_i}) \bigcup \Sigma^{\mathcal{M}}$, $G = (\|_i G^{C_i}) \parallel (\|_k G_{\circlearrowleft,pc_k})$ et $E = (\|_l E_{\circlearrowleft}) \parallel (\|_l E_{\rightleftarrows})$. Par conséquent, la prise en compte à la fois de l'intervalle intramodal étendu $E_{\circlearrowleft}$ et des spécifications intermodales $E_{\rightleftarrows}$ garantit le respect des contraintes internes et commutatives dans l'approche centralisée. Cependant, cette extension ne peut pas être effectuée automatiquement, car l'ajout des événements de commutation dépend de l'interaction des événements déjà présents avec ceux qui seront ajoutés. Cela ne peut être effectué que manuellement par un expert et les modèles

peuvent être sensiblement différents de ceux de l'approche présentée, bien que le résultat final soit équivalent en termes de supervision.

## Fusion des états non-significatifs

La dernière étape de la conception du contrôle (étape 4 de la Figure 4) concerne la réduction de la complexité des modèles obtenus afin de ne garder que le comportement interne et le comportement commutatif de chaque mode [Far10]. Pour obtenir cette dynamique réduite, l'automate des modes $G^{\mathcal{M}}$ a permis d'inclure dans chaque état des modèles les informations concernant le mode actif. Il est ainsi possible d'isoler les états correspondant au comportement interne du système dans chaque mode et de fusionner les états non inclus dans le comportement interne du mode. Ainsi, l'ensemble des états qui ne correspondent pas au mode étudié, appelés *états non significatifs*, forment un nouvel état appelé *état inactif*, noté $y_{id}^{M_j}$, qui représente l'inactivité du mode. Cet état est utilisé pour mettre en évidence l'unicité du mode actif, de sorte que pour les modes $N$, les modes $N-1$ seront dans cet état inactif et qu'un seul mode sera actif. Les nouveaux modèles de procédés contrôlées, où les états non significatifs ont été fusionnés dans l'état inactif, sont respectivement notés $H_{fusion,st,i}^{M_j}$ et $H_{merge,gr}^{M_j}$ pour les niveaux de la station et du réseau.

Étant donné que la composition parallèle synchronise les événements courants, lorsque les modes $N-1$ sont dans leur état inactif, il convient de s'assurer qu'ils ne restreignent pas le comportement du mode qui représente les points communs du comportement actuel du système. Pour cette raison, il faut s'assurer que l'état inactif peut générer tous les événements possibles, que ce soit dans une boucle ou en dehors de cet état. Formellement, si $Y_{mer}^{M_j}$ est l'ensemble des états non significatifs, l'automate $H_{merge,gr}^{M_j}$ est construit comme proposé dans [Far10].

Grâce à cette fonction de fusion, le concepteur dispose d'un modèle de mode qui représente le comportement interne du système dans le mode dans lequel les événements de commutation activent ou désactivent le mode en entrant ou en quittant l'état inactif. Ces comportements distincts respectent les contraintes liées à chaque mode et le concepteur peut extraire la loi de contrôle à mettre en oeuvre.

Ensuite, pour que le comportement équivalent global obtenu au moyen de la composition parallèle des automates $H_{merge}^{M_j}$ pour chaque mode $M_j$ de $G^{\mathcal{M}}$ soit non bloquant, les conditions suivantes doivent être remplies:

*Condition 3* L'automate de modes $G^{\mathcal{M}}$ est non bloquant, c'est-à-dire que l'ensemble des états marqués peut toujours être atteint. Formellement, $L_m(G^{\mathcal{M}}) = L(G^{\mathcal{M}})$.

*Condition 4* Le comportement interne, s'il existe, de chaque mode $M_j$ est non bloquant. Formellement, $\forall M_j \in \mathcal{M} | \mathcal{C}_{\circlearrowleft}^{M_j} \neq \emptyset, \{L_m(H_{in}^{M_j}) = L(H_{in}^{M_j})\}$.

*Condition 5* Il existe un comportement de commutation dans chaque mode permettant d'activer et de désactiver le mode, garantissant ainsi que l'ensemble des états marqués dans $G^{\mathcal{M}}$ est at-

teint. Officiellement:

$$\forall M_j \in G^{\mathcal{M}}, \forall H^{M_j}_{merge}, \{\exists y^{M_j}_{id} | \exists ((\tau^{M_j}_{merge}(y^{M_j}_{id}, \sigma^{\mathcal{M}}) = q^{M_j}_{in}), (\tau^{M_j}_{merge}(q^{M_j}_{in}, \sigma^{\mathcal{M}}) = y^{M_j}_{id}))\},$$

avec $q^{M_j}_{in} \in Q^{M_j}_{in}$ et $\sigma^{\mathcal{M}} \in \Sigma^{\mathcal{M}}$.

## Implémentation

Il existe un écart entre la structure asynchrone pilotée par les événements proposée dans SCT et les périphériques à base de signaux synchrones (généralement des périphériques informatiques) dans lesquels le contrôle par supervision est implémenté. Les méthodes de mise en oeuvre proposées dans la littérature ne sont cependant pas souvent adaptées au contrôle des systèmes hybrides, tels que les réseaux HVDC. En conséquence, une méthode d'implémentation capable d'intégrer le contrôle par supervision conçu en interaction avec le système physique à temps continu est proposée dans cette section. Étant donné que le contrôle par supervision

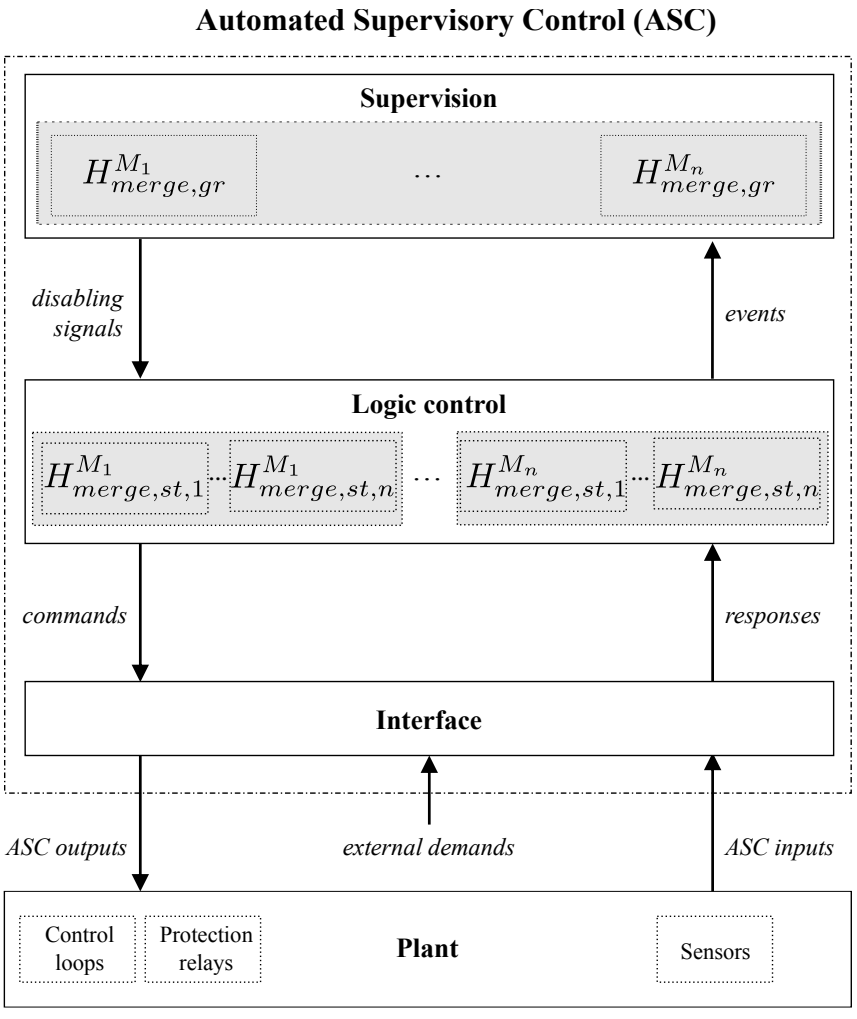



Figure 11 – Structure du contrôle implémente

dédié doit être distribué (décomposition horizontale, verticale et modale), l'approche *contrôle supervisé* présentée dans [Cha95] est considérée. L'architecture de contrôle proposée, donnée dans la Figure 11, repose sur une implémentation à plusieurs niveaux, inspirée de [Vie17].

En conséquence, le contrôle des événements discrets présenté dans la Figure 3 est divisé en deux niveaux distincts, de sorte que le procédé contrôlée obtenue du réseau est implémentée dans un niveau *Supervision* réalisant des fonctions de supervision. , tandis que les installations contrôlées des différentes stations sont implémentées dans un niveau *Contrôle logique* qui réalise des fonctions de contrôle classiques. Enfin, le niveau *Interface* relie les signaux à temps continu des capteurs et des signaux numériques aux actionneurs du système physique à leur contrepartie à événements discrets utilisée dans le contrôle par supervision [Rom19]. De plus, en raison de la nécessité d'un contrôle par supervision hautement réactif et personnalisé dans les systèmes HVDC, les automates programmables industriels ne sont probablement pas adaptés aux applications HVDC. En conséquence, la méthode de mise en oeuvre proposée dans cette thèse est basée sur des langages communs, tels que le code C.

De plus, en raison de la structure multi-niveaux présentée dans la Figure 11, plusieurs symboles d'entrée et de sortie seront attachés aux différents événements des modèles à implémenter. Par conséquent, contrairement aux machines Moore traditionnelles, une transition peut être déclenchée par plusieurs symboles d'entrée provenant de différents niveaux de contrôle. De même, il peut être nécessaire de générer plusieurs symboles de sortie dans chaque état. Par conséquent, nous proposons dans cette thèse l'utilisation d'un cas particulier de la machine de Moore, que nous appelons *machine de Moore étendue*, de sorte que chaque transition des automates implémentés soit associée à un ensemble de conditions booléennes et à chaque état est associé à un ensemble de symboles de sortie.

Aussi, comme les événements de commutation, s'ils sont contrôlables, ne sont associés à aucun signal numérique de le procédé réelle, ils ne génèrent aucune commande associée et seul le signal d'événement est modifié lorsque la transition de commutation de mode associée se produit. De plus, certains événements pouvant être partagés entre les modèles de différents modes, il est nécessaire de distinguer au niveau de la supervision le mode qui a généré un événement, de sorte qu'une commande associée à un événement contrôlable autorisé par un mode inactif mais désactivée par le le mode actif n'est pas généré. Étant donné que le problème mentionné concerne la synchronisation des signaux de désactivation des différents modèles de mode, les signaux de commande et de réponse ne sont pas affectés, car le niveau de supervision génère les signaux de désactivation uniquement sur les signaux d'événement.

Les expressions d'entrée et de sortie définies pour les différents niveaux de contrôle sont conçues comme des expressions génériques pouvant être interprétées et implémentées dans le langage choisi par le concepteur, qui doit être compatible avec les exigences d'implémentation des systèmes HVDC. Par conséquent, le langage choisi pour l'implémentation dans cette thèse est le code C. La structure de transition d'état des automates est ensuite implémentée à l'aide

d'instructions switch-case. Cependant, si l'utilisation des automates programmables était envisagée pour l'implémentation du contrôle par supervision, la méthode proposée pourrait être implémentée sous la forme d'un langage structuré (ST) [Int03], étant donné la similarité de sa syntaxe avec le code C [Bas07].

Si le contrôle par supervision est implémenté en tant que code C, chaque niveau de contrôle peut être codé en tant que fonction individuelle. La fonction correspondant au niveau de contrôle logique comprend un ensemble de fonctions correspondant chacune à une installation contrôlée de la station. Par conséquent, les symboles d'entrée et de sortie de chaque niveau, à savoir $\sigma, rsp\sigma, cmd\sigma, \sigma d, asc_i, asc_o$, sont déclarés comme des variables entières qui seront manipulées par les fonctions correspondantes. Il convient de noter qu'il suffit de classer d'abord les conditions d'entrée d'un événement incontrôlable pour indiquer sa priorité sur un événement contrôlable, car le code est exécuté ligne par ligne pendant un cycle d'analyse.

Les différentes fonctions de code C écrites pour chaque niveau du contrôle par supervision sont alors coordonnées via une fonction *ASC* appelée par un périphérique ou un logiciel externe et fonctionne comme principal du contrôle par supervision. Par conséquent, les différentes variables sont déclarées et initialisées dans cette fonction, à l'exception des entrées et des sorties du contrôle par supervision, qui sont les paramètres de la fonction ASC car ils sont associés aux signaux d'E/S du système de contrôle échangés avec le réseau HVDC. Les différentes fonctions correspondant aux niveaux de contrôle sont alors appelées chaque fois que la fonction ASC est appelée par le périphérique externe ou le logiciel et que les variables associées sont données en paramètres. Pour permettre aux fonctions de modifier la valeur de leurs variables de sortie, ces dernières doivent cependant être déclarées en tant que pointeurs.

Par conséquent, toutes les variables internes à le contrôle par supervision sont déclarées en tant que variables *static* afin qu'elles ne soient pas initialisées à chaque cycle d'exécution, bien qu'il soit important de garder à l'esprit qu'elles doivent être réinitialisées à chaque cycle d'exécution. L'effet décrit dans [Fab98] n'apparaît pas.

Par conséquent, le contrôle par supervision est exécuté dans l'ordre suivant:

- Remise à zéro des variables de désactivation $\sigma d$
- Appel de la fonction *supervision*
- Appel de la fonction *merge_ds*
- Remise à zéro des variables d'événement $\sigma$
- Appel de la fonction *logic control*
- Remise à zéro des variables de réponse $rsp\sigma$
- Appel de la fonction *interface*
- Remise à zéro des variables de commande $cmd\sigma$

Pour conclure, la génération du code associé aux différents niveaux de contrôle peut être largement automatisée étant donné que les informations nécessaires à la construction de l'expression générique et leur traduction dans le code sont fournies par les modèles d'automates. La fonction

d'interface nécessite toutefois l'intervention manuelle du concepteur, car elle est spécifique au système.

# Validation de la méthode par simulation

L'étude de cas correspond à un réseau HVDC multi-terminaux composé de trois stations de conversion par pôle, comme l'illustre la Figure 12, lors d'un cycle d'opération comportant le comportement de commutation de mode présenté à la Figure 8.

## Description du système

Le système considéré comprend deux pôles avec une polarité de tension opposée, de sorte que chaque pôle est contrôlé indépendamment. Chaque pôle est connecté à son réseau alternatif adjacent via un ensemble de stations de conversion. De plus, toutes les stations d'un même pôle sont interconnectées au moyen de câbles HVDC qui transfèrent la puissance dans tout le réseau. De cette manière, si un câble est hors service, le courant peut toujours être transféré car il existe un chemin électrique alternatif.

En raison de la topologie bipolaire, le point médian des deux stations de conversion en cascade est connecté à une référence de terre. En conséquence, la sortie inférieure des MMC au pôle positif est connectée à la terre et inversement pour les MMC dans le pôle négatif. La référence de terre est située à la station 1, tandis que le reste est relié à la référence commune au moyen d'un retour métallique dédié à basse tension afin d'empêcher le courant de circuler librement dans la terre.

Les différents câbles connectés à chaque station sont raccordés du côté DC de la station au moyen d'une tension de bus (notée N1, N2 et N3 dans la Figure 12), de sorte que l'ajout de la puissance transférée chaque câble connecté au bus est égal à la puissance totale transmise par le convertisseur. De plus, les modules de coupure côté DC se distinguent selon leur emplacement. Si le module est situé avant la tension du bus, il est appelé module de coupure du convertisseur et est noté $C_{BM}$. Lorsque le module est associé à une ligne ou à un câble et situé à son extrémité dans la station correspondante, il est appelé un module de rupture de ligne et désigné par $L_{BM}$. Alors que les algorithmes implémentés dans un relais $C_{BM}$ reposent uniquement sur des données locales, ceux implémentés dans un relais $L_{relay}$ peuvent être basés sur des informations communiquées par le relais situé à l'extrémité opposée du câble ou des autres relais de la station, en plus des mesures locales. Les données communiquées sont des signaux binaires ou des mesures de courant et de tension.

Enfin, dans la mesure où un sous-système de protection contre les courts-circuits est considéré comme correctement isolé par les sous-systèmes de protection lors de leur première tentative, les modules DCCB sont uniquement composés dans notre étude de cas, les commutateurs rapi-

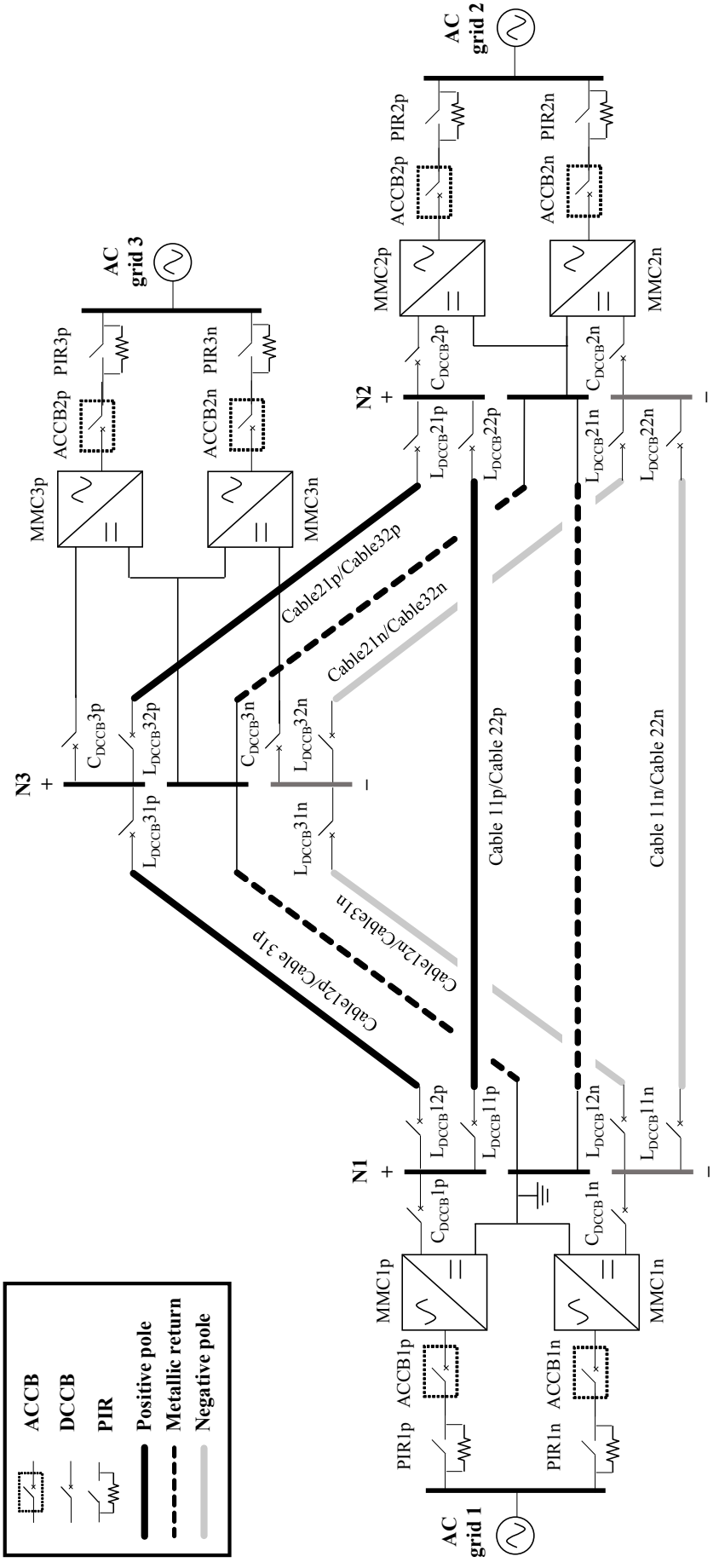



Figure 12 – Réseau MTDC bipolaire à trois terminaux

des n'étant pas nécessaires. De même, les modules de résistance de pré-insertion ne sont pas nécessaires du côté DC de la station, par opposition au côté AC. contrairement au côté AC. Pour conclure, le module ACCB et le module PIR sont commandés par le même relais de protection, tandis que chaque DCCB est commandé par un relais dédié.

## Logiciels utilisés

Pour traiter cet exemple, nous utiliserons plusieurs logiciels. Tout d'abord, le logiciel Supremica [Mal17] est utilisé pour les tâches liées à la conception du contrôle par supervision, telles que la construction de modèles, l'abstraction de modèles et la synthèse contrôlée de procédés; en raison de sa grande bibliothèque d'opérateurs sur les automates. Deuxièmement, afin de générer le code C résultant de la méthode d'implémentation, un programme informatique basé sur Python [Oli07] a été développé dans le cadre de cette thèse. Le logiciel construit prend en entrée un fichier XML créé par Supremica qui contient les automates à implémenter et permet de quasi-automatiquement générer le code C correspondant au contrôle par supervision associé, selon la méthode d'implémentation proposée dans cette thèse de doctorat. Enfin, en raison de la grande taille des systèmes de transmission de puissance, de leur complexité et du coût élevé associé à leur construction et à leur fonctionnement, tout dispositif ou contrôle à utiliser dans le système doit d'abord être testé et validé dans un environnement de simulation. Plus précisément, les programmes EMTP (ElectroMagnetic-Transient Programs) sont des outils logiciels permettant d'analyser assez précisément la dynamique du système d'alimentation. Le logiciel EMTP utilisé dans cette thèse est EMTP-RV [Mah07]. Ce programme permet de construire un modèle du système à simuler au moyen d'une bibliothèque dédiée de composants, y compris des composants HVDC, tels que le MMC. Le code C contenant le programme de contrôle par supervision a été implémenté dans EMTP-RV sous la forme d'une bibliothèque de liens dynamiques (DLL) [Har05] afin d'effectuer une simulation hors ligne de l'étude de cas considérée.

## Résultats de simulation

Le réseau MTDC à trois terminaux donnée à la Figure 12 a été modélisée dans le logiciel de simulation des transitoires électromagnétiques EMTP-RV grâce à sa bibliothèque de composants dédiée. Les contrôleurs MMC [Shi17] et les algorithmes de protection associés aux relais de protection [Lou17] ont été également intégrés. Ensuite, en suivant la méthode d'implémentation proposée, les automates de réseau fusionnés de chaque mode ont été implémentés dans la fonction *Supervision* du C proposé, tandis que les automates de station fusionnés de chaque mode ont été implémentés dans la fonction *Contrôle logique*. La valeur des différents seuils qui relient les signaux numériques correspondant aux mesures de tension et de courant du composant dans EMTP-RV avec les variables de réponse dans la fonction *Interface* a été déterminée à travers les analyses fonctionnelles et de surveillance des différents composants.

À ce stade, le fichier de code source contenant le programme de contrôle par supervision est intégré dans le logiciel de simulation EMTP-RV au moyen d'une DLL. Afin de créer la DLL, un ensemble de fichiers spécifique au logiciel est fourni par la société chargée du développement d'EMTP-RV. Le code C personnalisé contenant le programme de contrôle par supervision peut alors être appelé à partir des fichiers mentionnés via un fichier d'en-tête que nous avons précédemment écrit. Ensuite, les signaux numériques associés aux composants du logiciel sont liés manuellement à l'entrée ou à la sortie du contrôle par supervision correspondant et la DLL est générée. La DLL obtenue peut finalement être implémentée dans EMTP-RV à des fins de simulation. De plus, étant donné que les deux pôles du réseau MTDC sont identiques (sauf pour la polarité de la tension continue) et fonctionnent indépendamment, la commande de supervision doit être exécutée deux fois (une commande de surveillance par pôle). À cette fin, deux copies de la même DLL peuvent être utilisées. En effet, les variables internes de chaque copie restent locales à la DLL car les fonctions internes ont été codées avec des pointeurs dans cette thèse. Sinon, l'utilisation de variables globales obligerait à nommer différemment les variables de chaque copie, ce qui prendrait beaucoup de temps.

Une simulation hors ligne est ensuite effectuée afin de valider le comportement intramodal et intermodal du contrôle par supervision mis en oeuvre, qui est exécuté à chaque pas de temps (fixé à 10 micro-seconde). Étant donné le degré de détail du modèle de simulation EMTP-RV et l'impossibilité de tester les contrôleurs obtenus dans un système réel, la simulation effectuée est considérée comme une méthode valide pour la validation du contrôle par supervision proposé. Aussi, afin de montrer l'indépendance de fonctionnement des deux pôles du réseau MTDC, un défaut de court-circuit écran-sol n'est provoqué que dans le pôle positif, tandis que le pôle négatif reste sans défaut pendant toute la simulation. En conséquence, le contrôle par supervision lié au pôle positif suivra la dynamique de changement de mode décrite par les modes automates (cf. Figure 9), alors que celle liée au pôle négatif ne sera concernée que par $M_1$, $M_2$, $M_3$, $M_4$ et $M_5$, étant donné que l'événement SCF ne se produit jamais. Les résultats de simulation obtenus sont présentés à la Figure 13, Figure 14, Figure 15 et Figure 16 pour les deux pôles du réseau. De plus, le mode du système à chaque instant est indiqué dans les Figures, sauf pour $M_1$, $M_3$, $M_6$ et $M_8$ étant donné qu'ils n'ont pas de comportement interne et donc l'intervalle de temps pendant lequel le système est dans ces modes est très court pour être apprécié.

Dans la Figure 13, l'évolution de la tension du bus DC pendant le fonctionnement du réseau MTDC est affichée. Au début de la simulation, les deux pôles sont dans $M_1$, bien que le mode de démarrage $M_2$ soit immédiatement activé par le contrôle par supervision. La raison pour laquelle la charge de démarrage ne commence qu'à 0,05 est due au retard associé à la fermeture des disjoncteurs, qui est de 10 ms pour les DCCB et de 40 ms pour l'ACCB. Lors du démarrage, le contrôle par supervision impose dans le réseau les commandes nécessaires selon la stratégie considérée et l'évolution des composants du réseau jusqu'à ce que tous les MMC et les câbles soient chargés à la tension nominale, qui se produit autour de 1,25s. Par conséquent, le mode

$M_3$ est atteint à ce stade. Immédiatement après, le contrôle par supervision active le mode $M_4$ pour injecter la puissance désirée

le réseau, bien que cela puisse être difficilement apprécié dans la Figure 13. Une fois que les courants CC atteignent la valeur désirée, le mode marqué $M_5$ correspondant au fonctionnement nominal du réseau est activé et le système reste actif tant qu'il n'y a pas de défaut de court-circuit. À ce stade, les résultats de la simulation diffèrent pour les deux pôles. Alors que le pôle négatif reste à $M_5$ pour le reste de la simulation, un défaut de court-circuit est provoqué dans le pôle positif à 2s. Comme le réseau est obligée d'entrer $M_6$, les algorithmes de protection au sein du modèle de simulation répondent de manière autonome en exécutant la stratégie de protection décrite par les modèles intégrés dans le mode $M_7$ du contrôle par supervision. Comme le contrôle est capable de suivre l'exécution de la stratégie de protection, il est capable d'activer rapidement le mode de restauration de tension $M_9$ à la fin de celui-ci (environ 2,1 s), une fois que le système a entré $M_8$. Finalement, le système est ramené au mode marqué $M_5$ par le contrôle par supervision une fois que le flux d'énergie est rétabli à la valeur souhaitée dans $M_4$ .In Figure 14, le courant continu traversant chaque câble des deux pôles est montré. Pour chaque câble, les mesures des deux stations à la fin sont données. Ces mesures sont identiques, sauf pour la direction du courant mesuré (une station voit le courant comme sortant de la station tandis que l'autre voit le courant comme entrant dans la station). le réseau pendant le démarrage (de 0 à 1,25 s) jusqu'à ce que le réseau atteigne un équilibre de tension et que tous les MMC et les câbles soient chargés. En outre, la charge incontrôlée (de 0 à 1,1 s) peut être clairement distinguée de l'étape de charge contrôlée (de 1,1 à 1,25 s) par l'allumage des courants continus. Dans la Figure 14b, l'effet de l'activation de $M_4$ par le contrôle par supervision peut être clairement vu lorsque les courants CC sont modifiés jusqu'à ce que la valeur souhaitée soit atteinte. Ensuite, le défaut de court-circuit dans le pôle positif se produit à 2s. Bien que l'impact sur le pôle négatif soit négligeable, le courant continu dans les câbles des pôles positifs augmente brusquement jusqu'à des valeurs de l'ordre de 5 à 10 p.u. Les algorithmes de protection dans chaque station détectent localement le défaut dès que le courant atteint 2 p.u. et ordonner aux CDCCB de s'ouvrir pour éteindre le courant continu. Ensuite, le câble défectueux est correctement identifié et les LDCCB associés sont ouverts. Dans la Figure 14a, le câble défectueux (câble 13) peut être clairement reconnu étant donné que les mesures aux deux extrémités du câble ont la même direction (en sortant de la station), que les courants continus se dirigent vers le défaut . Ce n'est pas le cas pour les câbles sains. Puis, le contrôle par supervision étant capable de suivre le comportement interne de $M_7$, il configure automatiquement les contrôleurs MMC pour restaurer la tension et les courants CC afin de ramener le système à $M_5$ immédiatement après que le câble défectueux a été isolé à environ 2,1 s, à l'exception du câble 13 qui est hors service.

Dans la Figure 15, le courant continu transmis par chaque station est indiqué pour les deux pôles. Le courant continu étant le produit de la tension continue et du courant continu, l'évolution

de la puissance suit la même évolution que le courant continu. Il convient de noter cependant que si les courants continus dans le pôle négatif sont légèrement affectés par la restauration de la tension du pôle positif autour de 2,2s dans la Figure 14b, le courant continu transmis par le pôle négatif reste constant tout au long du défaut de court-circuit. Cela est dû au fait que le contrôleur MMC régule l'alimentation CC entrant et sortant de la station et non la répartition de cette alimentation dans chaque câble. En conséquence, une perturbation du réseau alternatif commune aux deux pôles due à la restauration d'un pôle n'a pas d'impact sur la puissance transmise par l'autre, mais peut avoir une incidence sur la répartition de cette puissance à travers les câbles. Cependant, le courant continu dans les câbles reste dans la zone de fonctionnement non nulle et sécurisée et donc cette légère perturbation n'a pas un impact majeur sur le fonctionnement du pôle négatif, car le ASC ne désactive jamais $M_5$.

Enfin, la tension dans les MMC des deux pôles est indiquée à la Figure 16. Tout d'abord, le contrôle par supervision applique les commandes correspondantes au démarrage afin de ramener la tension dans les MMC à sa valeur nominale. Pendant la phase de charge contrôlée, les MMC sont chargées les unes après les autres afin de minimiser les fluctuations de la tension continue (voir la Figure 13 de 1.1s à 1.25s). Ensuite, comme on peut l'observer, le niveau de tension dans les MMC reste constant pendant toute la simulation pour les deux pôles, grâce à la protection autonome des MMC et à l'action du contrôle de surveillance. En effet, celle-ci modifie effectivement la configuration des contrôleurs MMC afin de maintenir la MMC dans une zone de fonctionnement contrôlable afin de rétablir l'équilibre énergétique dans chaque pôle lorsque cela est nécessaire.

En conséquence, les performances du contrôle par supervision automatisé développé pour l'étude de cas considérée se sont révélées satisfaisantes, car elles ont efficacement géré les transitions entre les modes du système lors de la simulation du fonctionnement du réseau et coordonné les actions des composants qui déterminent le comportement interne de chaque mode.

# Conclusion générale

Face à l'augmentation continue de la consommation mondiale d'énergie et aux préoccupations croissantes en matière de développement durable, les sources d'énergie renouvelables telles que le vent, le soleil, la biomasse, la géothermie, etc. sont considérées comme des alternatives prometteuses aux sources d'énergie traditionnelles. devrait remplacer les centrales à combustibles fossiles classiques dans les décennies à venir. Néanmoins, l'intégration de ces sources dans les réseaux électriques a déjà exercé une pression sans précédent sur les systèmes de transmission de puissance existants, basés sur la technologie traditionnelle à courant alternatif, en raison de leur nature intermittente et de leur emplacement probable. des centres de consommation. Par conséquent, une mise à niveau fondamentale du système de transmission à courant alternatif existant utilisant la technologie plus flexible du courant continu à haute ten-

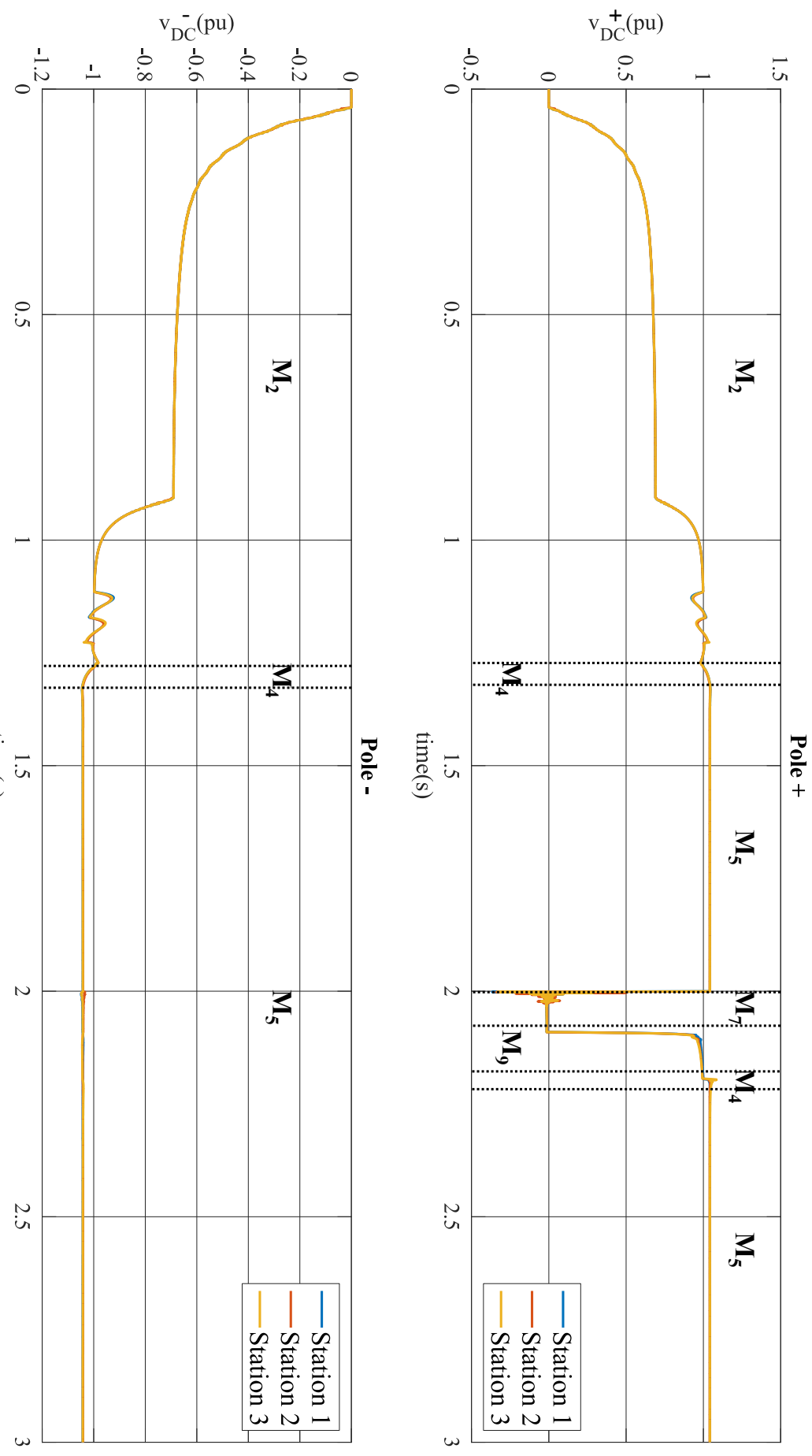
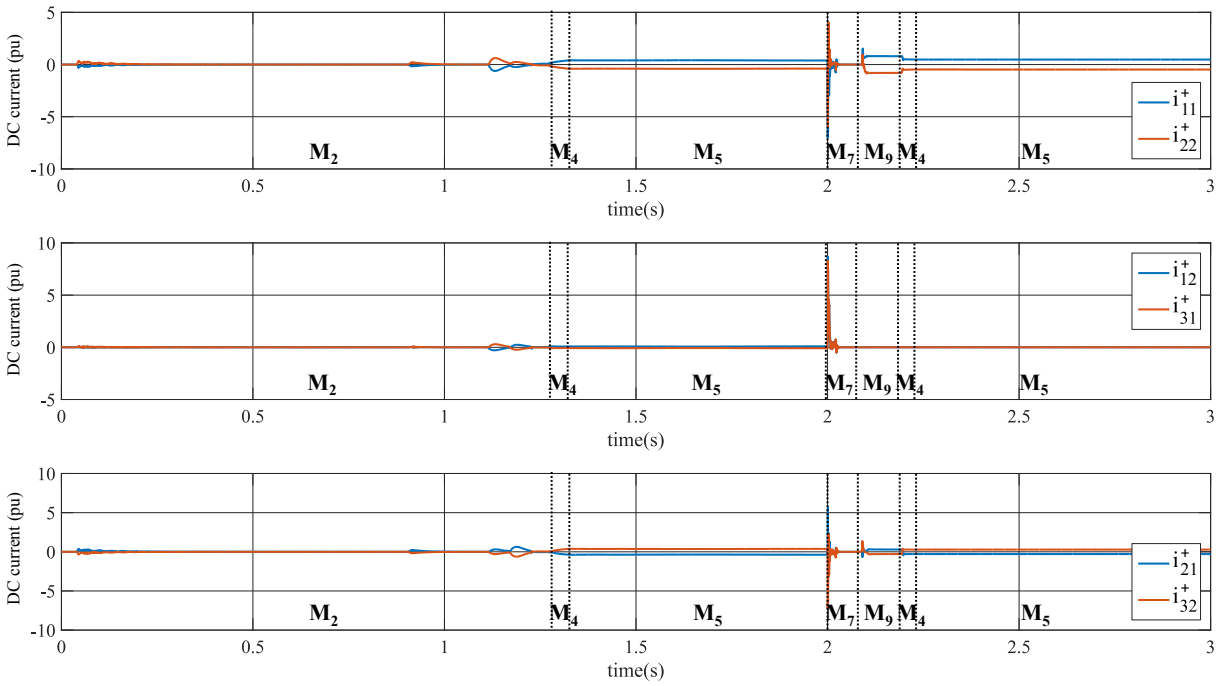
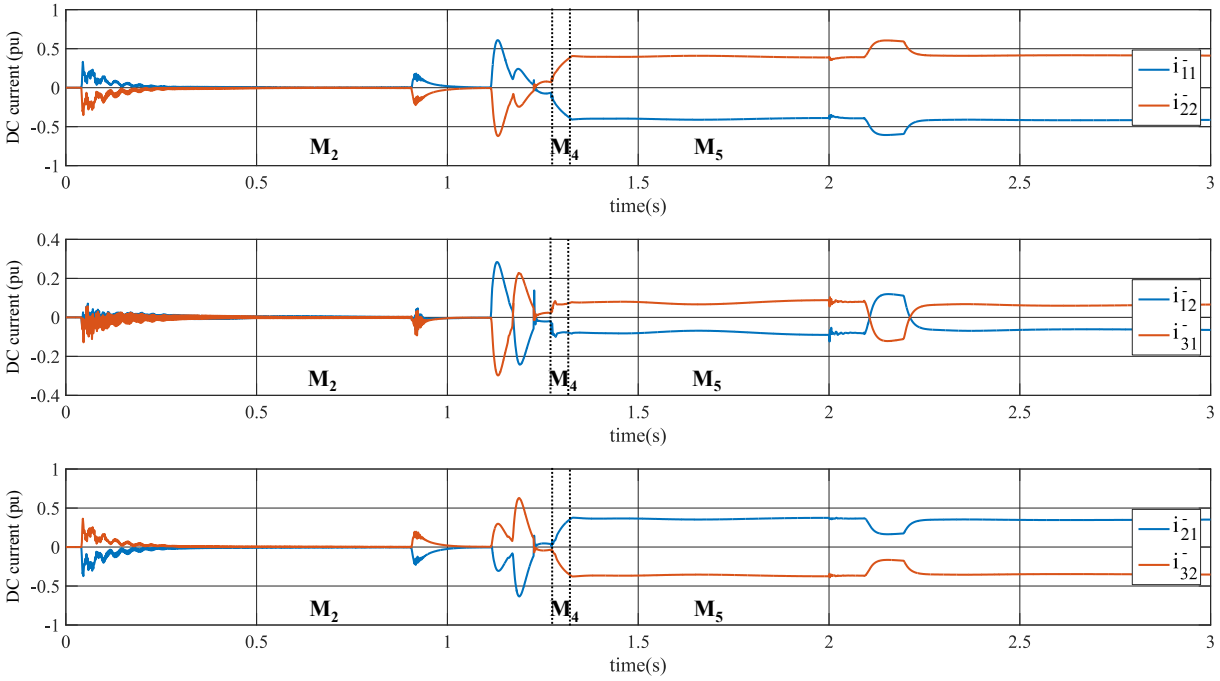



Figure 13 – Tension DC




(a) Pôle positif




(b) Pôle négatif

Figure 14 – Courant DC dans les câbles

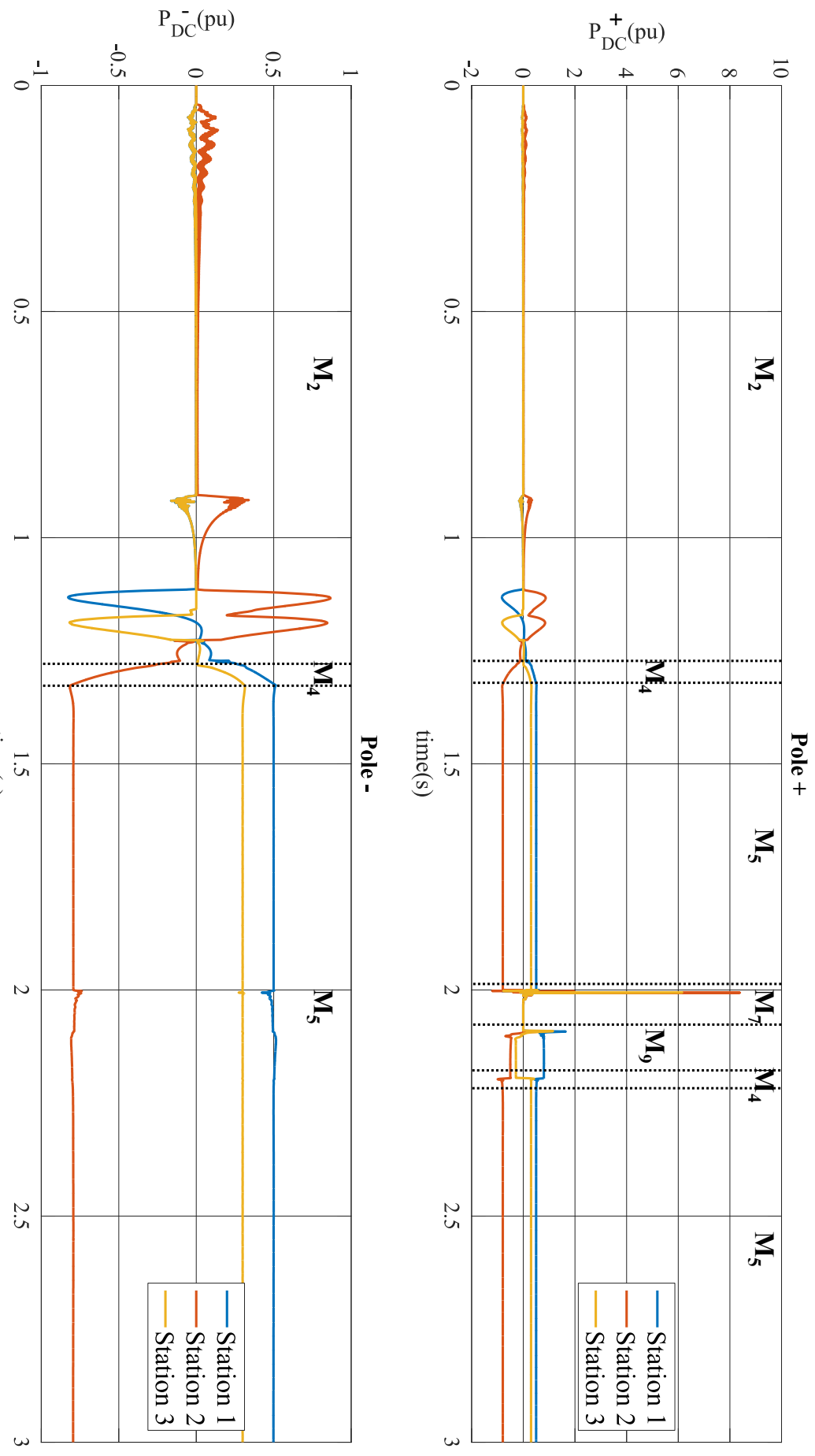



Figure 15 – Puissance DC

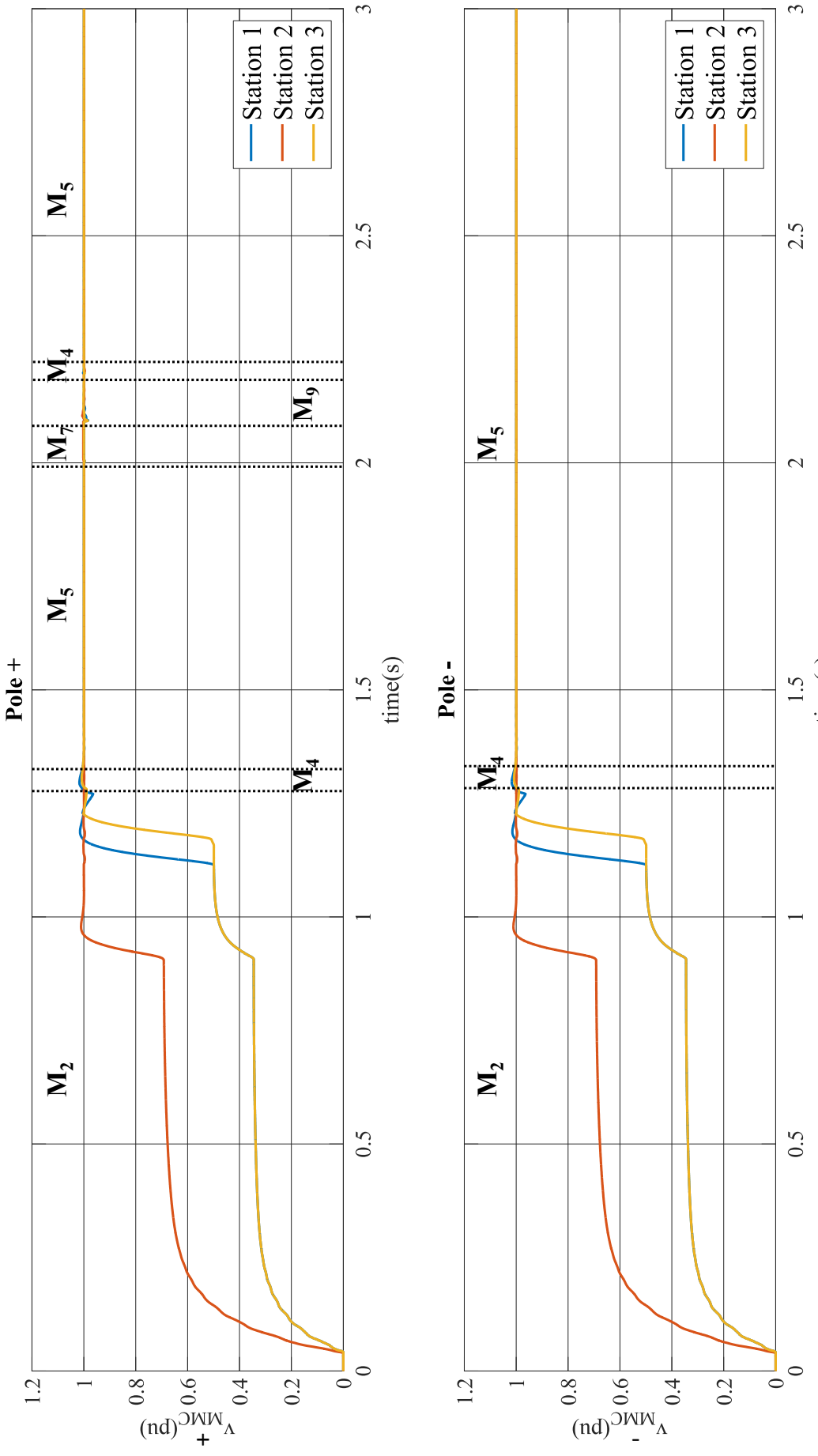



Figure 16 – Tension dans les MMC

sion (HVDC) est considérée comme la solution la plus prometteuse et techniquement réalisable pour l'intégration des sources d'énergie renouvelable dans le réseau. Compte tenu du nombre croissant d'applications, le concept de connexion de plusieurs stations HVDC pour former un réseau multi-terminal DC (MTDC) est apparu. Il est prévu de fournir plus de flexibilité dans la transmission de puissance et la fiabilité de l'alimentation électrique en servant de réseau électrique en vrac supplémentaire superposé au système de transmission CA.

Cependant, en raison du nombre élevé de dispositifs électroniques de puissance dans un système HVDC et de leur dynamique rapide, les perturbations se propagent rapidement dans un réseau MTDC, ce qui pourrait éventuellement entraîner une panne de courant et propager la perturbation dans tout le système de transmission. . En outre, une certaine coordination entre les composants du système HVDC est nécessaire car des interactions négatives peuvent survenir entre les multiples systèmes de protection et de contrôle déclenchés par une perturbation. Dans les systèmes HVDC, cependant, l'intervention humaine est fortement limitée pendant le fonctionnement en temps réel du réseau, par opposition aux systèmes à courant alternatif, en raison de la courte période pendant laquelle l'opération doit être restaurée après une perturbation. En conséquence, le développement d'un contrôle automatisé de supervision qui substitue l'opérateur humain aux tâches de surveillance en temps réel et de coordination des composants est essentiel si l'on veut s'attendre à un fonctionnement acceptable du réseau.

Ainsi, la principale contribution de cette thèse est de proposer une méthodologie systématique, autosuffisante et incrémentale pour la conception et la mise en oeuvre d'un contrôle par supervision capable de gérer les composants dans un système HVDC à différents niveaux de contrôle modes de fonctionnement de la réseau. Considérant que les exigences opérationnelles d'un système de transmission de puissance, qu'il soit CA ou HVDC, sont satisfaites par la gestion des interactions entre sa dynamique à événements discrets et continus, la méthode proposée est basée sur la modélisation des systèmes à événements discrets (DES). De plus, étant donné qu'un mode est une configuration particulière du système où un ensemble de composants doit répondre à un ensemble de spécifications, la méthode de conception proposée dans cette thèse repose sur des approches formelles dans le cadre de la théorie du contrôle de supervision (SCT). outil puissant pour la construction de modèles sûrs et éviter les étapes de vérification des spécifications par la suite sur les modèles construits.

Bien que l'utilisation de techniques basées sur SED ait déjà été proposée dans la littérature pour le fonctionnement des systèmes d'alimentation en courant alternatif, l'absence d'une architecture de contrôle cohérente et non ambiguë prenant en compte un comportement réaliste du système a rendu difficile son application. les solutions proposées en dehors des études de cas considérées. Par conséquent, les approches formelles utilisées dans cette thèse ont été choisies de manière à adapter l'architecture du contrôle par supervision à la nature des systèmes de transmission de puissance géographiquement dispersés (stations de conversion réparties sur l'ensemble du réseau). un centre de contrôle) et faire l'expérience de diverses situations pendant

leur fonctionnement (traitement des pannes, restauration de l'équilibre de puissance, etc.).

En conséquence, la méthode de conception proposée fusionne les approches de décomposition verticale et modale. Concernant la décomposition verticale, l'approche compositionnelle [Moh14] est retenue car elle permet de définir plusieurs niveaux de contrôle lors de la phase de conception et n'est pas limitée par une structure prédéfinie. Cela permet une conception flexible qui facilite l'intégration de nouveaux aspects avec un impact minimal sur la structure existante, ce qui est particulièrement intéressant pour le système MTDC, étant donné que ces systèmes ne sont pas encore totalement compris. En ce qui concerne la décomposition modale, cette thèse suit les travaux antérieurs concernant la gestion des modes [Far10], qui ont abouti à une architecture de contrôle de modèles concurrents, avec un modèle dans chaque mode représentant à la fois le comportement interne et commutatif de le mode actif est respecté et aucune contradiction n'existe dans la loi de contrôle produite.

En ce qui concerne les aspects pratiques de la thèse, la conception proposée et la méthode de mise en oeuvre d a d'abord été illustré par une étude de cas portant sur le démarrage d'un lien point à point [Rom17]. De plus, un outil logiciel qui automatise largement la méthode d'implémentation proposée a été développé. Cet outil manipule les modèles construits dans le logiciel Supremica et génère en sortie un fichier de code source C pouvant être implémenté dans le logiciel EMTP-RV. L'applicabilité de la méthodologie de conception et de mise en oeuvre proposée a ensuite été validée pour un cycle de fonctionnement réaliste. une réseau MTDC à 3 terminaux et la simulation qui en résulte s'est révélée satisfaisante.

Pour conclure, il faut invariablement trois choses pour développer un contrôle par supervision automatisé: il est nécessaire de définir les états de départ et d'arrivée de le contrôle par supervision pour tout scénario de fonctionnement du système; il est nécessaire d'identifier les fonctions de chaque composant à appliquer par le contrôle par supervision et de définir leur relation avec le reste du système pour tout scénario d'exploitation donné; enfin, le contrôle par supervision doit être conçu et mis en oeuvre au moyen d'une méthode choisie. Ainsi, une méthodologie approfondie et approfondie pour l'analyse des exigences fonctionnelles et de surveillance du système est fournie afin de garantir la cohérence de le contrôle par supervision avec le système physique. Nous avons également fourni une approche claire et précise de la représentation formelle, de la conception méthodique et de la vérification mathématique des comportements internes et commutatifs des modes à plusieurs niveaux de contrôle. La décomposition modale et verticale du système, ainsi que l'approche multi-modèle, permettent de limiter l'explosion combinatoire tout en permettant un comportement global non bloquant et contrôlable, de sorte que tous les modèles sont sûrs par construction et cohérents les uns avec les autres . Enfin, une méthode d'implémentation facilitant le test et la validation du contrôle par supervision conçu lors d'une simulation réaliste d'une opération de réseau MTDC dans un logiciel dédié a été proposée.

En outre, le contexte dans lequel la thèse a été menée (programme de recherche P1 au sein de

l'institut SuperGrid) a permis de travailler en étroite collaboration avec des experts en systèmes électriques, qui ont suscité de nombreux échanges. l'utilité de formaliser leur travail à des fins de supervision par des approches basées sur les SED et la TCS.

Une perspective de notre travail serait d'étendre les modes et les commutations considérés lors de l'étude intermodale afin de couvrir l'ensemble des scénarios d'exploitation résultant de l'analyse de suivi. Une autre perspective intéressante serait d'établir une partition hiérarchique des modes, de sorte que les modes des composants soient différenciés des modes de station et des modes de réseau, à un degré tel que les différents niveaux de fonctionnement auraient des modes de fonctionnement différents. De cette manière, le mode d'un niveau d'opération classé supérieur serait le résultat de la composition des modes de niveau inférieur, soulignant ainsi l'influence qu'un composant unique peut avoir sur le comportement global et augmentant la connaissance du système.

De plus, bien que nous ayons indiqué dans cette thèse qu'un contrôle par supervision pour les applications HVDC devrait idéalement être conçu au moyen d'approches de décomposition horizontale, verticale et modale, la décomposition horizontale était hors de la portée de nos travaux. Ainsi, les travaux futurs envisageront certainement la possibilité de décentraliser le contrôle par supervision et de localiser le programme de prise de décision aussi près que possible des capteurs et des actionneurs de chaque station. Pour cela, une analyse approfondie des exigences de communication dans un système HVDC doit être effectuée avant que toute approche formelle soit intégrée dans la méthode de conception du contrôle par supervision.

De plus, le niveau d'interface du contrôle par supervision mis en oeuvre pourrait être largement amélioré. Par exemple, les signaux à temps continu provenant des capteurs de réseau sont actuellement liés à leur événement discret correspondant via la détection d'un franchissement de seuil prédéfini. Une interface plus sophistiquée pourrait interagir avec des observateurs complexes afin de faciliter l'intégration de techniques de contrôle avancées, telles que le contrôle prédictif des modèles [Mor11], au système de supervision.

D'un point de vue pratique, les travaux futurs pourraient envisager l'extension de la méthode d'implémentation proposée à d'autres langages d'implémentation autres que le code C. Par exemple, la méthode étant basée sur des expressions textuelles génériques, la première pourrait être étendue au langage Structured Text (ST) [Int03], étant donné la similarité de sa syntaxe avec le code C [Bas07]. Enfin, bien que la simulation hors ligne soit considérée comme valide pour le test du contrôle par supervision (qui est exécuté toutes les 10 µs), les performances du contrôle par supervision développé pendant le fonctionnement d'un réseau MTDC pourraient être évaluées dans un des logiciels de simulation en temps réel, tels que HYPERSIM [Do99]. Ce programme est très similaire à EMTP-RV et le code C obtenu dans cette thèse pourrait donc être implémenté dans HYPERSIM. La simulation en temps réel pourrait être particulièrement intéressante pour l'analyse des besoins de communication dans un système HVDC en vue de la décentralisation du contrôle par supervision.

# FOLIO ADMINISTRATIF

## THESE DE L'UNIVERSITE DE LYON OPEREE AU SEIN DE L'INSA LYON

NOM : ROMERO RODRÍGUEZ
(avec précision du nom de jeune fille, le cas échéant)

DATE de SOUTENANCE : 09/11/2018

Prénoms : Miguel

TITRE : «Synthèse de contrôle par supervision pour des systèmes HVDC à base de convertisseurs modulaires multiniveaux / Supervisory control synthesis for MMC-based HVDC systems »

NATURE : Doctorat

Numéro d'ordre : 2018LYSEI081

Ecole doctorale : EDA160 E.E.A

Spécialité : Automatique

RESUME :

Ces dernières années, les technologies à courant continu haute tension (en anglais, HVDC) basées sur les convertisseurs modulaires multiniveaux (MMC) sont adoptées comme solution pour l'intégration efficace des énergies renouvelables dans les réseaux électriques. Cependant, ces technologies présentent de nouveaux défis dans la façon dont les systèmes de transmission de puissance sont contrôlés et exploités, car des stratégies de contrôle plus rapides et plus complexes seront nécessaires dans un domaine qui repose aujourd'hui fortement sur la décision humaine. Dans ce contexte, la modélisation des systèmes à événements discrets (SED) et la théorie du contrôle par supervision (TCS) sont des outils puissants pour la synthèse de superviseurs qui assurent que le système à contrôler respecte un ensemble de spécifications comportementales, imposées par le concepteur, dans ses limites physiques.
Ce travail propose une méthode pour le développement complet, de la conception à la mise en œuvre, du contrôle par supervision d'un système Multi-Terminal DC (MTDC). Une analyse du système considéré a été effectuée afin d'identifier les principaux composants et modes de fonctionnement du réseau. La solution proposée repose sur la modélisation par événements discrets du comportement en temps continu des composants du système. A partir de là, les concepts de la TCS sont appliqués de manière à obtenir une architecture de contrôle hiérarchique prenant en compte la priorité de certaines actions de contrôle à traiter au niveau local. De plus, les contrôleurs discrets obtenus présentent une structure de commutation de mode afin de réaliser une gestion de mode pendant le fonctionnement du réseau MTDC. Enfin, une méthode pour la mise en œuvre des contrôleurs obtenus dans un logiciel de simulation de système électrique répandu est proposée. L'ensemble du travail a été validé par la simulation d'une étude de cas impliquant la gestion des modes d'un système MTDC bipolaire à trois terminaux.

MOTS-CLÉS : Systèmes à évènements discrets, Systèmes HVDC de transmission de puissance, Théorie de contrôle par supervision, Gestion de modes, Implémentation de contrôleur automatique, Analyse de systèmes HVDC.

Laboratoire (s) de recherche : Laboratoire Ampère

Directeur de thèse: Éric NIEL

Président de jury :

Composition du jury :

| | | |
|---|---|---|
| FABIAN, Martin | Professeur des Universités, Univ. de Technologie de Chalmers | Rapporteur |
| ZAMAÏ, Éric | Maître de conférences HDR, Grenoble INP | Rapporteur |
| SECHILARIU, Manuela | Professeur des Universités, Univ. de Technologie de Compiègne | Examinatrice |
| MOREL, Hervé | Directeur de Recherche CNRS, INSA Lyon | Examinateur |
| CARDOZO, Carmen | Docteur, Ingénieur de recherche, Réseau de Transport d'Electricité | Examinatrice |
| DELPOUX, Romain | Maître de conférences, INSA Lyon | Co-encadrant de thèse |
| PIÉTRAC, Laurent | Maître de conférences HDR, INSA Lyon | Co-directeur de thèse |
| NIEL, Éric | Professeur des Universités, INSA Lyon | Directeur de thèse |
| BENCHAÏB, Abdelkrim | Docteur HDR, Manager R&D, SuperGrid Institute | Invité, Co-encadrant de thèse |
| DAI, Jing | Professeur assistant, CentraleSupélec | Invité, Co-encadrant de thèse |