HVDC-WISE

# Deliverable 4.2

## Technology modelling

## Document Properties

| | | |
|---|---|---|
| Funding Program | Horizon Europe and UK Horizon Europe funding guarantee | |
| Grant Agreement Number | 101075424 (UK 10041877 and 10051113) | |
| Project | **HVDC-**WISE | |
| Deliverable Id | D4.2 | |
| Title | Deliverable 4.2: Technology modelling | |
| Distribution Level | Public | |
| Due Date | 31/03/2024 | |
| Date Submitted | | |
| Status | Draft | |
| Version | 1 | |
| Work package / Task | WP4, Task 4.2 | |
| Authors | Yang, Ning | University of Strathclyde |
| | Morel, Florent | SuperGrid Institute |
| | Ouoba, Sidlawendé | SuperGrid Institute |
| | Lazarotto, Damiano | SuperGrid Institute |
| | Gonzales, Juan-Carlos | SuperGrid Institute |
| | Tishenin, Georgii | RWTH Aachen |
| | Ivanov, Chavdar | TenneT TSO, GmbH |
| | Cosic, Said | TenneT TSO, GmbH |

## Version History

| **Version** | **Date** | Comment |
|---|---|---|
| 1.0 | 18/02/2025 | |

## Approval Flow

| **Title** | **Person** | **Date** | Comment |
|---|---|---|---|
| Task Lead | | | |
| WP Lead | Florent Morel | 11/02/2025 | |
| Coordinator | | | |



HVDC-WISE is supported by the European Union's Horizon Europe program under agreement 101075424.

UK Research and Innovation (UKRI) funding for HVDC-WISE is provided under the UK government's Horizon Europe funding guarantee [grant numbers 10041877 and 10051113].

Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Climate, Infrastructure and Environment Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.

# Table of Contents

# List of Figures

# List of Tables

# Executive Summary

The operation and planning of power systems require extensive use of simulations and corresponding models. The emergence of new equipment and system configurations raises the need for new models and simulation approaches. This report is part of the HVDC-WISE project and deals with modelling approaches with three different perspectives: providing models for the next steps of the project; prepare the exchange of standardized models of HVDC equipment; prepare models of power electronic converters for a promising simulation method (dynamic phasors).

Firstly, based on the outcomes of the previous steps of the project (deliverables D2.2 and D4.1) and considering the software tool mainly used in WP6 (DIgSILENT PowerFactory), this report describes custom models and model for new devices not included in the standard library of this software platform. Two modelling frameworks are investigated: one is based on the development of models on the native language of this platform (DIgSILENT Simulation Language (DSL)); the other is based on compiled black-box models and allows reuse of existing models from other software platforms (functional mock-up units (FMUs)). The developed models are as follows: modular multilevel converter (MMC) in grid-following mode with DSL modelling, in grid-forming mode with DSL and FMU modelling, energy dissipating system, DC connected energy storage system, wind farm with wind turbines in grid-forming mode. They are tested individually and shared with the project partners. These models will be used in a common simulation to demonstrate their usefulness in deliverable D4.4 and can be used in next steps of the project (WP6).

Secondly, the project investigated the development of standardized models of HVDC equipment. The IEC Common Information Model (CIM) standards, including the IEC Common Grid Model Exchange Standard (CGMES) are considered as they are widely employed by Transmission System Operators (TSOs) to facilitate the exchange of AC power system models in both long-term and operational planning processes. A critical aspect of this exchange is the incorporation of HVDC models, which are essential for accurately representing power system behaviour. While the IEC standards provide a foundational framework, they require extensions to meet the additional demands associated with HVDC systems and their controls. This deliverable describes research activities held in the HVDC-WISE project towards creation of HVDC-Wise Lib, a public library of models of HVDC equipment, based on IEC CIM standards (deliverable D4.3). The report highlights the goals and demonstrates the proposed approach for representing the static and dynamic parts of the models. It highlights the identified gaps and lessons learnt. It also paves the way for further recommendations for extensions of standards (deliverable D8.3).

The third modelling approach investigated in this document is related to dynamic phasors (DP), a novel modelling domain, that can potentially bridge the gap between electromagnetic transient (EMT) and root mean square (RMS) simulations. Models of AC/DC converters in grid-forming mode in EMT, DP and RMS domains have been developed and tested for DPsim, an open-source software tool. This tool supports modelling and simulation in all three domains and then allows to compare the results of different simulation approaches. The report presents the current status of power electronics modelling in DP domain, results of development in HVDC-WISE, existing limitations and further research questions. The models of inverters developed here will be used in D7.1 to assess converter interactions.

# 1. Introduction

There is a trend for more complex HVDC systems (with more than two terminals, extending beyond traditional AC/DC converters and cables). Simulations are mandatory to assess their benefits, validate control schemes and guarantee their integration in AC systems. Previous outcomes of the HVDC-WISE project have highlighted the need for enhanced modelling approaches both for the success of the project and for further exploitation of the results as summarised hereunder.

Firstly, as the goal of the project is to identify reliable and resilient AC/DC hybrid architectures, the needed models should be available to support the simulations required for the project. This is the case for equipment used in existing installations but not for new technologies highlighted in the deliverable D4.1. As some models have been developed before the project by different partners, there is a need to define modelling approaches to build models for the software platform used for use cases (WP6) from models initially developed for various software platforms. Also, even if models are available for converters like modular multilevel converters, the development of control schemes in WP3 necessitates the creation of customized models of these devices.

Secondly, for the future exploitation of the results, transmission system operators and vendors need to exchange models based on common formats, highlighting the need of standardised approaches. It is then needed to develop a library of standardized models of HVDC equipment (initiated in deliverable D4.3). However, prior to this, a well-defined modelling approach must be established to facilitate the development of these models. The development of the first models is also the opportunity to assess the capabilities and limitations of the existing standards highlighting the need for further extensions. These insights will contribute to the final recommendations of the project (further deliverable D8.3).

Thirdly, dynamic phasors (DP) offer a promising approach to study power systems: more accurate than root mean square (RMS) simulations and more computationally efficient than electromagnetic transient (EMT) simulations. There is a need to assess the benefits and limitations of dynamic phasors to study phenomena like converter interactions but the current experience in converter modelling in DP domain is poor, especially considering grid-forming converters and open-source implementations.

Regarding these challenges and needs, this deliverable has the following scope and objectives. As DIgSILENT PowerFactory will be used in use cases, the first goal is to define modelling approaches for this software platform, including considering the import of models initially built for other software packages. The second goal is to share with partners some models built upon these approaches. This allows to validate the approaches and gain valuable experience regarding the advantages and limitations of the modelling frameworks. The second goal is to define a modelling approach to develop models of HVDC equipment according to IEC Common Information Model (CIM) standards such as IEC Common Grid Model Exchange Standard (CGMES). This modelling approach must include static and dynamic parts, and the first examples of developed models should highlight the need of extensions of standards. The last objective is to improve modelling capabilities of DPsim (an open-source simulator for the RMS, DP and EMT domains) for power electronic devices.

This report is organised as follows. Section 2 is dedicated to DIgSILENT PowerFactory. Two modelling approaches for this software platform are presented in subsection 2.1: with the DIgSILENT Simulation Language (DSL) and with Functional Mock-up Units (FMUs). This second approach allows to exchange compiled models generated from other software platforms. Then, subsection 2.2 describes the models developed in the project and shared with partners. Section 3 is dedicated to CIM/CGMES modelling of HVDC devices. A workflow highlighting the need to distinguish static and dynamic parts of the models is presented. The application of variants of the workflow is presented as well as the identified gaps

and lessons learned from the first developments. This section also paves the way for further recommendations regarding the needed extensions of the standards. Section 4 is dedicated to DP and DPsim and the development of models for grid-forming inverters tested in RMS, DP and EMT domains. Finally general conclusions are drawn in section 5.

# 2. Modelling approaches in PowerFactory and description of released models

In previous steps of the project, tools and methodology to plan resilient hybrid AC/DC architectures have been defined in WP5 and T6.1. Coordination with partners involved in WP6 highlighted that most of them will use DIgSILENT PowerFactory to perform studies for use cases. Also, WP2 highlighted demonstration needs for the project and T4.1 dealt with potential technologies for the use cases. Many but not all the needed models were already available in the DIgSILENT PowerFactory library. It has been then decided to provide the project with models of building blocks for this software platform. Based on D2.2, on D4.1 and on an internal survey, taking into consideration both technological maturity and developments in WP3, several critical components have been identified for model development either due to lack of standard model in DIgSILENT PowerFactory (e.g., energy storage system connected to HVDC terminals or wind farm with wind turbines in grid-forming mode) either because some building blocks are of special interest for the project (and then a custom model is appreciated) e.g., modular multilevel converter (MMC) in grid-forming mode.

This section presents then two modelling frameworks in DIgSILENT PowerFactory and provides key building blocks essential for use case studies. The modelling approaches comprise the DIgSILENT Simulation Language (DSL), which facilitates power system controller development through both graphical block diagrams and direct coding, and the Modelica-based methodology, which enables integration of external models via the Functional Mock-up Interface (FMI) Standard. Building upon these frameworks, the released models include MMC with both grid-following and grid-forming capabilities, energy dissipating systems, HVDC-connected energy storage solutions, and wind farms with wind turbines in grid-forming mode. Models have been developed thanks to one of the presented approaches or with both approaches in the case of MMC in grid-forming mode. These models will serve as fundamental elements for subsequent use case analyses and studies within the project scope.

## 2.1 Modelling approaches in PowerFactory

Power system simulation models can be broadly divided into static and dynamic models. Root mean square (RMS) dynamic models have been the most widely used type of dynamic models for assessing most power system technical performance issues of classical power systems, from a planning and operations perspective. DIgSILENT PowerFactory is among the most widely adopted power system analysis tools in research and industry. It provides a comprehensive library of electrical power system device models that users can customize and expand with new models for RMS simulation. Building dynamic models in DIgSILENT PowerFactory is realised using one of the following two options:

- using DSL specific objects: DSL Model and DSL Model Type, or
- using Modelica specific objects: Modelica Model and Modelica Model Type.

### 2.1.1 DIgSILENT Simulation Language

The DIgSILENT Simulation Language (DSL) is used to design and simulate dynamic models for representing various dynamic systems, controllers of electrical equipment as well as other components used in electric power systems.

The DSL is employed to define custom dynamic controllers. These controllers receive input signals from the simulated power system and respond by generating various output signals. During the simulation, the model equations of DSL models are integrated with those describing the dynamic behaviour of power system components. These combined equations are then evaluated together, resulting in a cohesive dynamic simulation.

Block diagrams are a popular and intuitive way to create Domain Specific Language (DSL) Model Types. These diagrams visually represent complex systems that have multiple inputs and outputs, and can include time-based differential equations and non-linear elements.

At its core, a block diagram shows how output signals are derived from input signals through various operations and transformations. The basic building blocks of these diagrams include Block References, Summation Points, Multipliers, Divisors, Switches, Input/Output nodes, Routed signal lines and Signal labels.

While block diagrams represent the predominant methodology for constructing DSL Model Types, an alternative approach exists: the utilization of pure DSL code within the Equations page of the Edit Dialog. This method, although less frequently employed, possesses the capability to fully encapsulate a dynamic system. The DSL code implementation can incorporate the following elements:

- Parameter descriptions: name and unit
- Allowed parameter ranges
- Initial conditions and functions which are used to calculate initial values
- Algebraic and differential equations defining the dynamic model during simulation.

However, the efficacy of this code-based approach is constrained by model complexity. Empirical evidence suggests that its application is optimal for models not exceeding several dozen lines of code. This limitation stems from the inverse relationship between code volume and model comprehensibility.

Overall, DSL offers a comprehensive framework for the mathematical representation of both linear and non-linear systems in continuous time domains. Its architecture, encompassing built-in macros and an array of intrinsic functions, facilitates expeditious development of controller and dynamic models with considerable flexibility. The language incorporates several formal verification procedures, enhancing model integrity. These include:

- Algebraic loop detection
- Identification of unused and undefined variables
- Flagging of missing initial conditions.

Such features enable thorough model validation prior to simulation execution, thereby streamlining the testing process.

However, the utilization of DSL for dynamic model construction requires a high level of proficiency. Users must possess:

- A thorough understanding of dynamic model structuring within the DSL framework
- Familiarity with the specialized terminology and abbreviations intrinsic to DSL syntax.

DSL also faces certain limitations. The inherent complexity of its user interface, coupled with insufficient support for advanced control algorithms, presents challenges in the development of sophisticated controllers and intricate dynamic models. Furthermore, the DSL approach may prove

suboptimal in scenarios requiring frequent code modifications or prioritizing rapid implementation with minimal effort—conditions often encountered during model development and rapid prototyping phases.

### 2.1.2 User-defined dynamic models via Modelica

The implementation of user-defined dynamic models via Modelica objects adheres to the modelling paradigms delineated by the Modelica language. A key feature of this approach is the capability to link Modelica models in DIgSILENT PowerFactory to externally interfaced Functional Mock-up Units (FMUs) based on the Functional Mock-up Interface (FMI) Standard. The standard allows users to make their models and simulation tools accessible in the form of FMU that contains an FMI-based formalisation as well as some forms of representation of the tool in equations. FMI functions are used by a simulation environment to create one or more instances of the FMU and to simulate them, usually with other models. These FMUs can be imported into DIgSILENT PowerFactory and used within a large-scale RMS simulation or a detailed EMT simulation, which is illustrated in Figure 2.1.

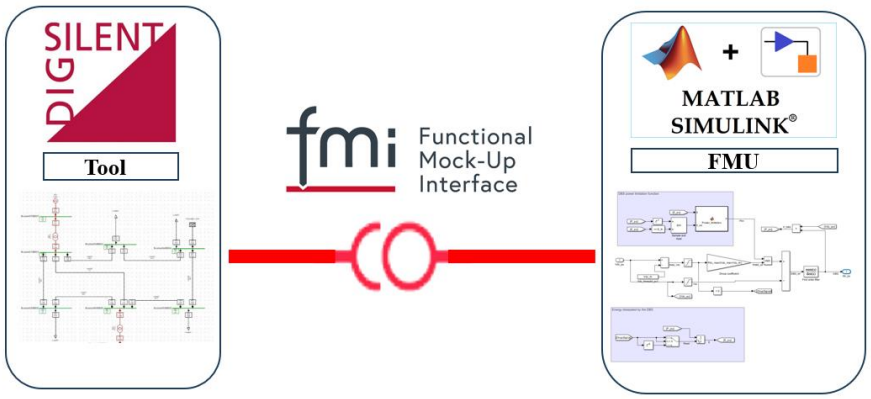


FIGURE 2.1 INTERFACE BETWEEN POWERFACTORY AND FMU MODELS EXPORTED BY MATLAB

To demonstrate the Modelica-FMU approach in interfacing between Matlab Simulink and DIgSILENT PowerFactory, a detailed procedural overview is presented hereunder. In order to export the control system from Matlab Simulink to an FMU the following steps need to be taken.

1. Prepare a Matlab Simulink file only containing the block that needs to be exported and arranged as shown in Figure 2.2.
2. Copy paste in MATLAB command window the following lines:

```
% download and extract the distribution archive to the current folder
unzip(['https://github.com/CATIA-Systems/FMIKit-Simulink/releases/' ...
 'download/v3.1/FMIKit-Simulink-3.1.zip'], 'FMIKit-Simulink-3.1')
% add the folder to the MATLAB path
addpath(fullfile(pwd, 'FMIKit-Simulink-3.1'))
% initialize FMI Kit
FMIKit.initialize()
```

3. In Matlab Simulink:
   - Open the model in Matlab Simulink
   - Load the model parameters to Matlab Workspace
   - Open “Configuration parameters”. Depending on whether you want a Model Exchange or Co-Simulation FMU you need to change two parameters.
   - For Co-Simulation FMU:
     - Under “Solver” set Type to Fixed-step and select a suitable step size.
     - Under “Code generation” set System target file to grtfmi.tlc

- For Model Exchange FMU:
  - Under “Solver” set Type to Variable-step.
  - Under “Code generation” set System target file to rtwsfcnfmi.tlc
- Press CTRL+B to compile the model. It creates an fmu[1].

4. In DIgSILENT PowerFactory:
   - Open data manager
   - Create *Modelica Model Type* in *Dynamic Models* of *Library*
   - Check *Compiled model* and select the exported FMU

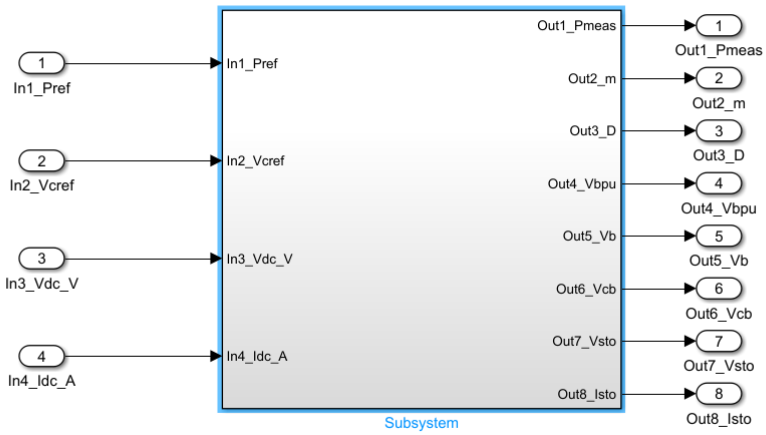


FIGURE 2.2 A SUBSYSTEM GATHERED IN A SIMULINK BLOCK

According to the above steps, the model built in Matlab Simulink can be exported as FMU and imported into DIgSILENT PowerFactory as a part of Modelica model. This approach offers significant advantages in enhancing simulation tool capabilities and model reusability while ensuring the protection of intellectual property (IP). It facilitates seamless integration across diverse simulation languages and platforms, enabling interoperability and extending the functionality of simulation tools. Using Modlica-FMU, models can be effortlessly exported as FMUs from other platforms and imported into different software packages (DIgSILENT PowerFactory) in this case, supporting model reuse. Additionally, FMI/FMU protects IP through black-box modelling, which emulates model functionality without exposing implementation details. The dual advantages of interoperability and IP protection are particularly relevant to this project, as they enable HVDC-WISE partners to securely share models in FMU form, thereby streamlining the model development process and enhancing collaborative efforts.

While offering significant advantages, the Modelica-FMU approach may introduce additional complexities at runtime, such as dependencies on platform-specific compilers or license server connections, which may either be intentional or arise from poor modelling practices. Furthermore, certain modelling limitations exist, such as MATLAB's inability to export acausal Simscape connectors, which restricts the export of electrical component models (e.g., synchronous generators) as FMUs. Additionally, while FMUs function as black-box models within DIgSILENT PowerFactory, their integration presents challenges in debugging compared to models developed using DSL.

[1] Here are the full instructions for export: https://github.com/CATIA-Systems/FMIKit-Simulink/blob/main/docs/fmu_export.md

## 2.2 Released models

This section presents model descriptions of several power electronics components in multi-terminal HVDC systems developed and implemented in DIgSILENT PowerFactory. The models encompass a range of critical components in HVDC systems, including Modular Multilevel Converters (MMC), grid-following and grid-forming converters, energy dissipation systems, energy storage systems, and wind farm models. These models have been constructed using either DSL native to DIgSILENT PowerFactory or through the integration of Modelica-based FMUs.

### 2.2.1 Modular Multilevel Converter

The Modular Multilevel Converter (MMC) model provided by DIgSILENT PowerFactory is based on a specific topology, as illustrated in Figure 2.3. This model demonstrates versatility by supporting both half-bridge and full-bridge submodule configurations, thus accommodating various converter designs and operational requirements. For load flow calculations and RMS simulations, the converter models are based on a fundamental frequency approach.

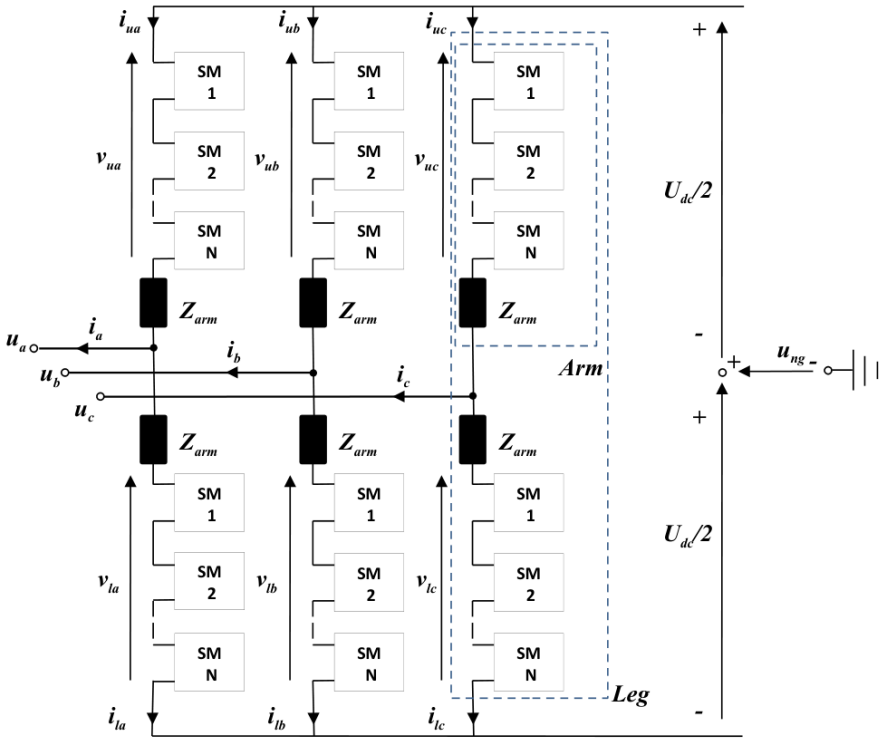


FIGURE 2.3 MODULAR MULTILEVEL CONVERTER TOPOLOGY

The model incorporates various loss mechanisms, including no-load losses, switching losses, and resistive losses, to enhance simulation accuracy. It is important to note that the DC representation of the MMC in fundamental frequency models is simplified. As a consequence, this model does not accurately reproduce phenomena related to energy balancing and circulating currents. This limitation should be considered when interpreting simulation results. For full-bridge MMC configurations, the DC terminal voltage is coupled to the equivalent arm capacitor voltage through the DC bias of the insertion index. This index can vary from -1 to 1, offering greater operational flexibility. In contrast, half-bridge MMC models assume the insertion index DC bias is always 1. It is worth highlighting that the fundamental frequency MMC model exclusively supports sinusoidal modulation.

### MMC in grid-following mode with DSL modelling

Grid-following (GFL) inverters have already been widely used for integrating wind and solar energy into power grids due to their simple control structure, their mature phase-locked loop (PLL) technology, and their feature of operating at a determined current (matching the maximum power point or dispatch point of the resource). A grid-following inverter controls the AC-side current and follows the phase angle of the existing grid voltage through a PLL. This section considers an MMC converter topology under a standard GFL mode, regulating the active power and voltage at the PCC. Its model in DIgSILENT PowerFactory is presented and exemplary results are shown.

The synchronisation of this three-phase MMC converter under GFL is achieved via PLL (shown in the light-blue box in Figure 2.4). The PLL acts on either the q or d-axis voltage measurement at the point of common coupling (PCC), outputting the converter angle for transformation and aims to reduce one qd voltage component to zero. In this case, an a-phase to d-axis alignment is used and the q-axis voltage component is set to zero via a proportional-integral (PI) controller:

$$\theta_{pll} = \frac{-V_{d,PCC} K_{pll}(s)}{s}$$

$$K_{pll}(s) = K_{p,pll} + \frac{K_{i,pll}}{s}$$

where $\theta_{pll}$ is the PLL angle output, $V_{d,PCC}$ is the PCC d-axis voltage and $K_{p,pll}$ and $K_{i,pll}$ are the PLL proportional and integral gains, respectively.

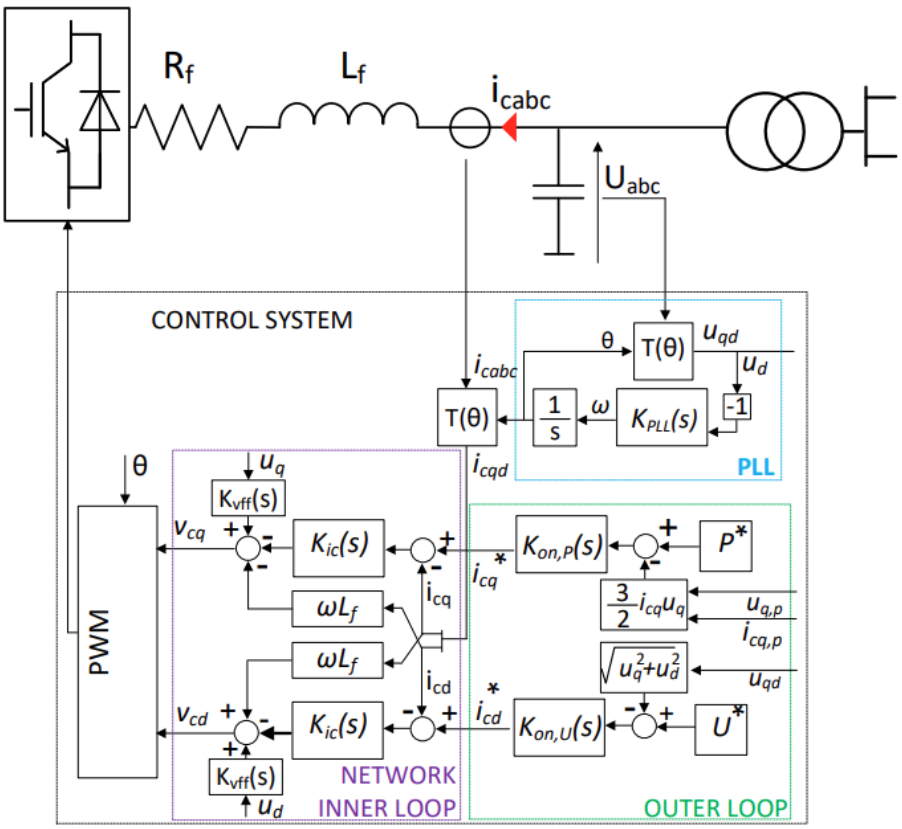


FIGURE 2.4 GFL CONTROL DIAGRAM

Besides, the synchronisation achieved by PLL, the inner loop current control regulates the synchronous reference frame converter currents by altering the terminal voltage commands of the

converter. This subsystem is shown in purple in Figure 2.4. Current is controlled via a PI controller on each axis with proportional and integral gains tuned via the modulus optimum criterion with tuning constant α used to produce a desired settling time.

$$K_{ic}(s) = K_{p,cc} + \frac{K_{i,cc}}{s}$$
$$K_{p,cc} = \frac{L_f}{\alpha}$$
$$K_{i,cc} = \frac{R_f}{\alpha}$$

Two feedforward terms are included in the control. The first deals with the cross-coupling between the q and d axes

$$i_{q,cc} = i_d L_f \omega_g$$
$$i_{d,cc} = -i_q L_f \omega_g$$

where $\mathrm{i_q}$ and $\mathrm{i_d}$ are the q and d-axis current, respectively and $\mathrm{i_{q,cc}}$ and $\mathrm{i_{d,cc}}$ are the q and d-axis converter current cross-coupling terms. The second feedforward provides the PCC voltage to reduce the strain on the current controller.

The outer loop controller allows for various operating modes of the GFL converter. Standard power voltage control (PVCC) is considered here in this work, which can be seen in the green box. Two PI controllers regulate the active power flowing from the converter and the voltage magnitude at the PCC by provided the reference signals for the internal current loops.

$$K_{on,P}(s) = K_{p,P} + \frac{K_{i,P}}{s}$$
$$K_{on,U}(s) = K_{p,U} + \frac{K_{i,U}}{s}$$

Where $\mathrm{K_{p,P}}$ and $\mathrm{K_{i,P}}$ are the power controller proportional and integral gains, respectively and $\mathrm{K_{p,U}}$ and $K_{i,U}$ are the voltage controller proportional and integral gains, respectively.

The above control scheme is built in DIgSILENT PowerFactory through DSL. Meanwhile, by using the built-in full-bridge type MMC PWM converter, then import the built GFL control into. The built GFL MMC converter is connected to a strong grid and the performance has been tested via simulation in DIgSILENT PowerFactory. The main parameters of the tested system are listed in Table 2.1.

TABLE 2.1 PVCC GFL PARAMETERS

| SYMBOL | DESCRIPTION | VALUE |
|---|---|---|
| $\mathrm{V_n}$ | Nominal line-line voltage | 360 kV |
| SCR | Short circuit ratio | 5 |
| $\mathrm{Z_R}$ | Transmission line resistance | 25.92 Ohm |
| $\mathrm{Z_L}$ | Transmission line inductance | 0.8251 H |
| $K_{p,pll}$ | PLL proportional gain | 1.2339 |
| $K_{i,pll}$ | PLL integral gain | 274.1557 |
| $K_{p,cc}$ | Current controller proportional gain | 500 |
| $K_{i,cc}$ | Current controller integral gain | 50000 |
| $K_{p,P}$ | Power controller proportional gain | 0.00000278 |
| $K_{i,P}$ | Power controller integral gain | 0.00002778 |
| $K_{p,U}$ | Voltage controller proportional gain | 0.0007716 |
| $K_{i,U}$ | Voltage controller integral gain | 0.0772 |

The active power reference decreases from 100 MW to 90 MW at 12 s and the voltage reference decreases from 360 kV to 359 kV, respectively. The simulation results are shown in Figure 2.5. It can be observed that the power and voltage can track the power and voltage reference after the initial transients and reference signal step changes. This validates the effectiveness of the built GFL MMC.

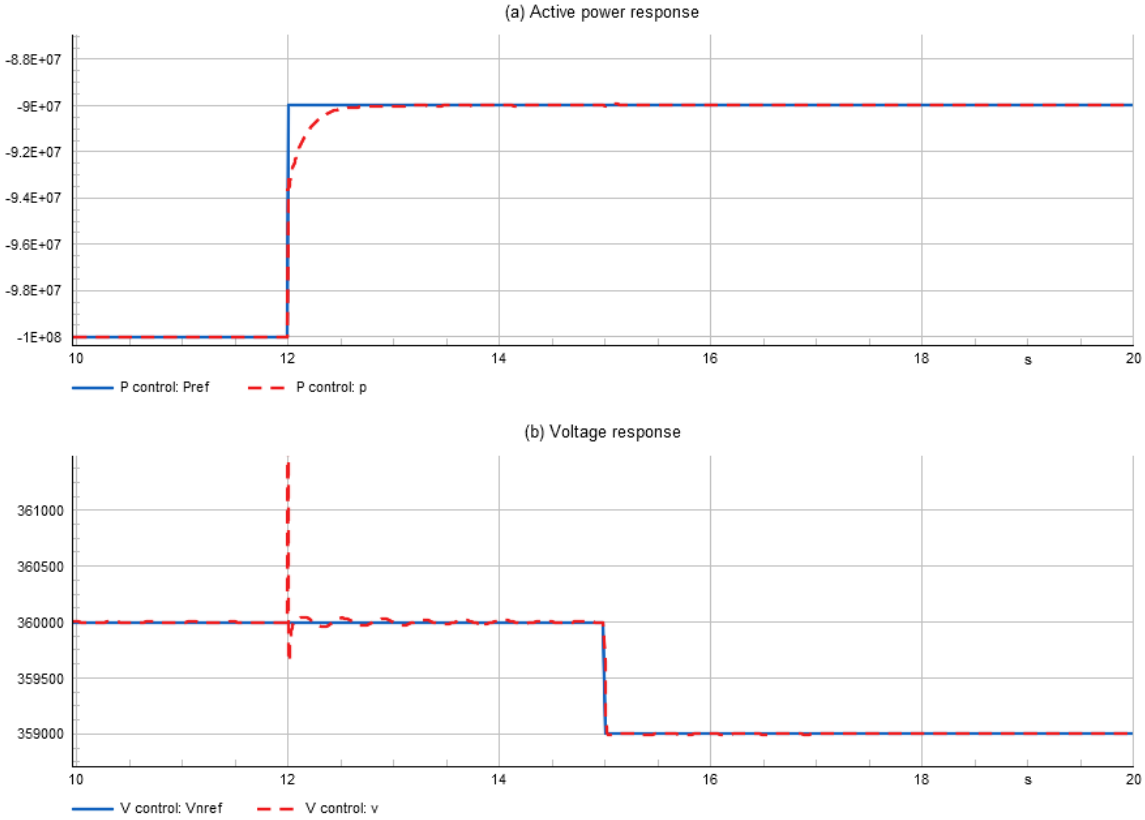


FIGURE 2.5 ACTIVE POWER AND VOLTAGE STEP CHANGE RESPONSE OF PVCC GFL

### MMC in grid-forming mode with FMU modelling

A grid-forming (GFM) inverter controls the AC-side voltage and contributes to the forming of a voltage source grid. It synchronizes with the rest of the grid through frequency droop control (normally P − ω droop), which is similar to the control of a synchronous generator. This section considers an MMC converter topology under a standard GFM mode, regulating the active power and voltage at the PCC. Its model in DIgSILENT PowerFactory is presented and exemplary results are shown.

Droop control (Figure 2.6 left) resembles the speed droop property of the SM governor and trades off deviations of the power injection (from its nominal value) and frequency deviations

$$\dot{\theta} = \omega$$
$$\omega = \omega^{\star} + d_{\omega}(p^{\star} - p)$$

where $d_{\omega}$ denotes the droop gain.

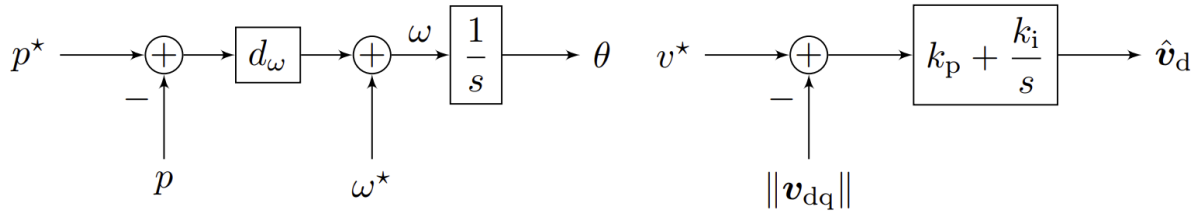


FIGURE 2.6 DROOP CONTROL FREQUENCY AND AC VOLTAGE OUTERLOOP CONTROL BLOCK DIAGRAMS

To replicate the service provided by the automatic voltage regulator (AVR) of synchronous machines, a PI controller acting on the output voltage error (Figure 2.6 right) is used to obtain the direct axis reference $\hat{v}_\mathrm{d}$ for the underlying voltage loop ($v^\star$ and $\|\boldsymbol{v}_\mathrm{dq}\|$ are the reference and measured voltage magnitude).

$$\hat{\boldsymbol{v}}_\mathrm{d} = k_\mathrm{p}\left(v^\star - \|\boldsymbol{v}_\mathrm{dq}\|\right) + k_\mathrm{i}\int_0^t \left(v^\star - \|\boldsymbol{v}_\mathrm{dq}(\tau)\|\right)\mathrm{d}\tau$$

One can note that $\hat{v}_\mathrm{q} = 0$ and the reactive power injection varies such that exact voltage regulation is achieved.

A standard converter control architecture that consists of a reference model providing a reference voltage $\hat{\boldsymbol{v}}_\mathrm{dq}$ with angle $\angle\hat{\boldsymbol{v}}_\mathrm{dq} = \theta$ and magnitude $\|\hat{\boldsymbol{v}}_\mathrm{dq}\|$ is used. The modulation signal $\boldsymbol{m}_{\alpha\beta}$ is determined by cascaded proportional-integral (PI) controllers that are implemented in dq-coordinates (rotating with the reference angle $\theta$ ) and track the voltage reference $\hat{\boldsymbol{v}}_\mathrm{dq}$ (Figure 2.7). The voltage tracking error $\hat{\boldsymbol{v}}_\mathrm{dq} - \boldsymbol{v}_\mathrm{dq}$ is used to compute the reference $\boldsymbol{i}^\star_\mathrm{s,dq} = \left[i^\star_\mathrm{sd} i^\star_\mathrm{sq}\right]^\mathsf{T}$ for the switching node current $\boldsymbol{i}_\mathrm{s,dq}$

$$\begin{aligned}\dot{\boldsymbol{x}}_{v,\mathrm{dq}} &= \hat{\boldsymbol{v}}_\mathrm{dq} - \boldsymbol{v}_\mathrm{dq},\\ \boldsymbol{i}^\star_\mathrm{s,dq} &:= \underbrace{\boldsymbol{i}_\mathrm{dq} + C\omega\mathcal{J}_2\boldsymbol{v}_\mathrm{dq}}_{\text{feed-forward terms}} + \underbrace{\mathcal{K}_{v,\mathrm{p}}\left(\hat{\boldsymbol{v}}_\mathrm{dq} - \boldsymbol{v}_\mathrm{dq}\right) + \mathcal{K}_{v,\mathrm{i}}\boldsymbol{x}_{v,\mathrm{dq}}}_{\text{PI control}}\end{aligned}$$

where $\boldsymbol{x}_{v,\mathrm{dq}} = \left[x_{v,\mathrm{d}} x_{v,\mathrm{q}}\right]^\mathsf{T}$ denotes the integrator state, $\boldsymbol{v}_\mathrm{dq} = \left[v_\mathrm{d} v_\mathrm{q}\right]^\mathsf{T}$ denotes the output voltage measurement, $\hat{\boldsymbol{v}}_\mathrm{dq} = [\hat{v}_\mathrm{d} 0]^\mathsf{T}$ denotes the reference voltage, $\boldsymbol{i}_\mathrm{dq} = \left[i_\mathrm{d} i_\mathrm{q}\right]^\mathsf{T}$ denotes the output current, $\mathcal{J}_2$ is the 2-D identity matrix, $\mathcal{K}_{v,\mathrm{p}} = k_{v,\mathrm{p}}\mathcal{J}_2$ and $\mathcal{K}_{v,\mathrm{i}} = k_{v,\mathrm{i}}\mathcal{J}_2$ denote diagonal matrices of proportional and integral gains, respectively.

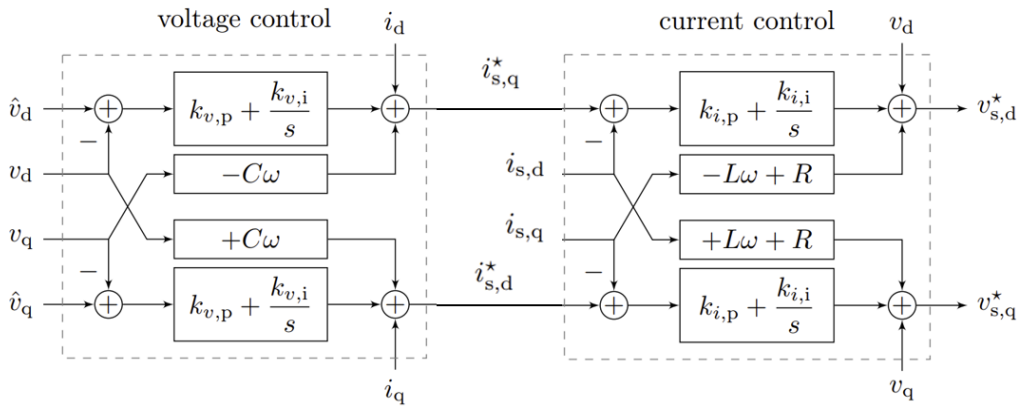


FIGURE 2.7 BLOCK DIAGRAM OF CASCADED INNER VOLTAGE AND CURRENT CONTROL LOOP

The mentioned GFM control is built in Matlab Simulink and exported as an FMU and imported into DIgSILENT PowerFactory through Modelica model. The controlled MMC converter is connected to an infinite bus and the performance has been tested via simulation in DIgSILENT PowerFactory. The main parameters of the tested system are listed in Table 2.2.

TABLE 2.2 DROOP GFM PARAMETERS

| SYMBOL | DESCRIPTION | VALUE |
|---|---|---|
| $V_n$ | Nominal line-line voltage | 360 kV |
| SCR | Short circuit ratio | 5 |
| $Z_R$ | Transmission line resistance | 25.92 Ohm |
| $Z_L$ | Transmission line inductance | 0.8251 H |
| $R$ | Filter resistance | 5e-06 Ohm |
| $L$ | Filter inductance | 1e-06 H |
| $d_\omega$ | Active power droop gain | 3.1416e-08 |
| $K_{p,P}$ | Power controller proportional gain | 0.00000278 |
| $K_{i,P}$ | Power controller integral gain | 0.00002778 |
| $K_{p,U}$ | Voltage controller proportional gain | 0.52 |
| $K_{i,U}$ | Voltage controller integral gain | 232.2 |
| $K_{p,I}$ | Current controller proportional gain | 0.73889 |
| $K_{i,I}$ | Current controller integral gain | 0.00595 |

The terminal voltage reference decreases from 1 p.u. to 0.95 p.u. at 1 s and a three-phase short circuit fault happened at 3 s and cleared after 0.1 s, respectively. The simulation results are shown in Figure 2.8. It can be observed that the power and voltage can track active power and voltage reference after voltage reference signal step changes and three-phase short circuit faults. This validates the effectiveness of the built GFL MMC using DSL.

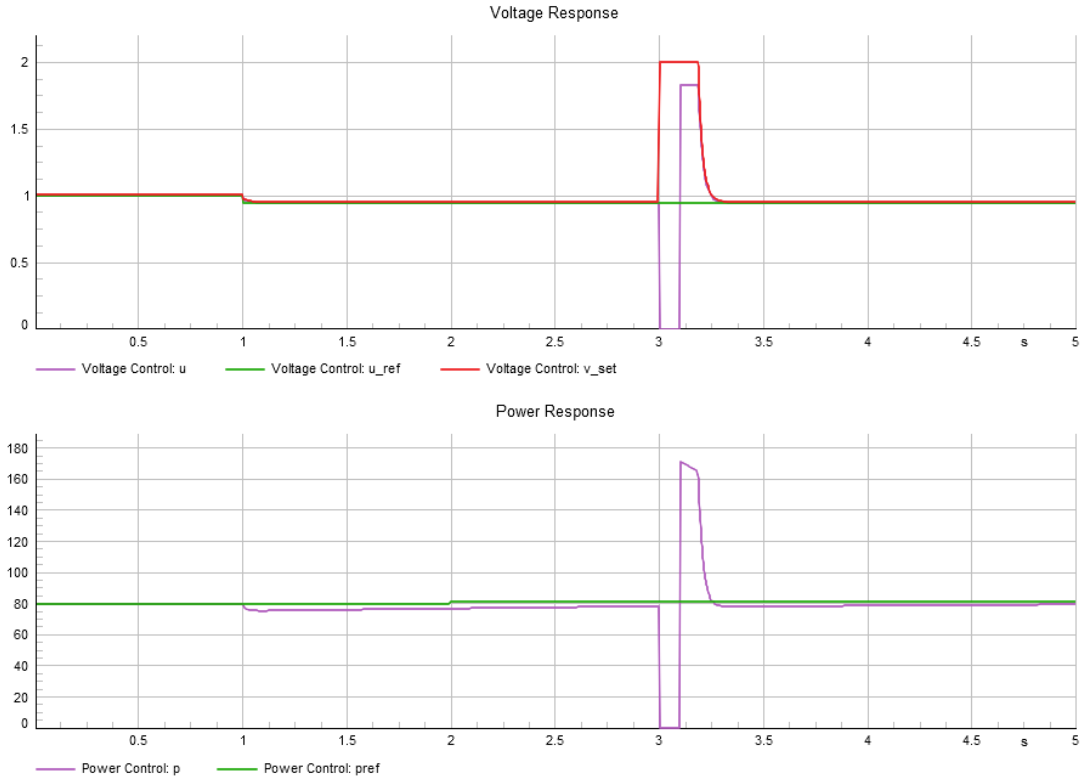


FIGURE 2.8 VOLTAGE AND POWER RESPONSE UNDER GRID-FORMING CONTROL USING DSL-BASED MODEL

The control performance of droop GFM via the FMU-Modelica model is shown in Figure 2.9. The terminal voltage reference decreases from 1 p.u. to 0.95 p.u. at 21 s and a three-phase short circuit fault happened at 23 s and cleared after 0.1 s, respectively. It can be observed that the droop GFM via FMU-Modelica model presents the same control performance as the droop GFM via DSL. This validates the effectiveness of the built GFL MMC using the FMU-Modelica model.

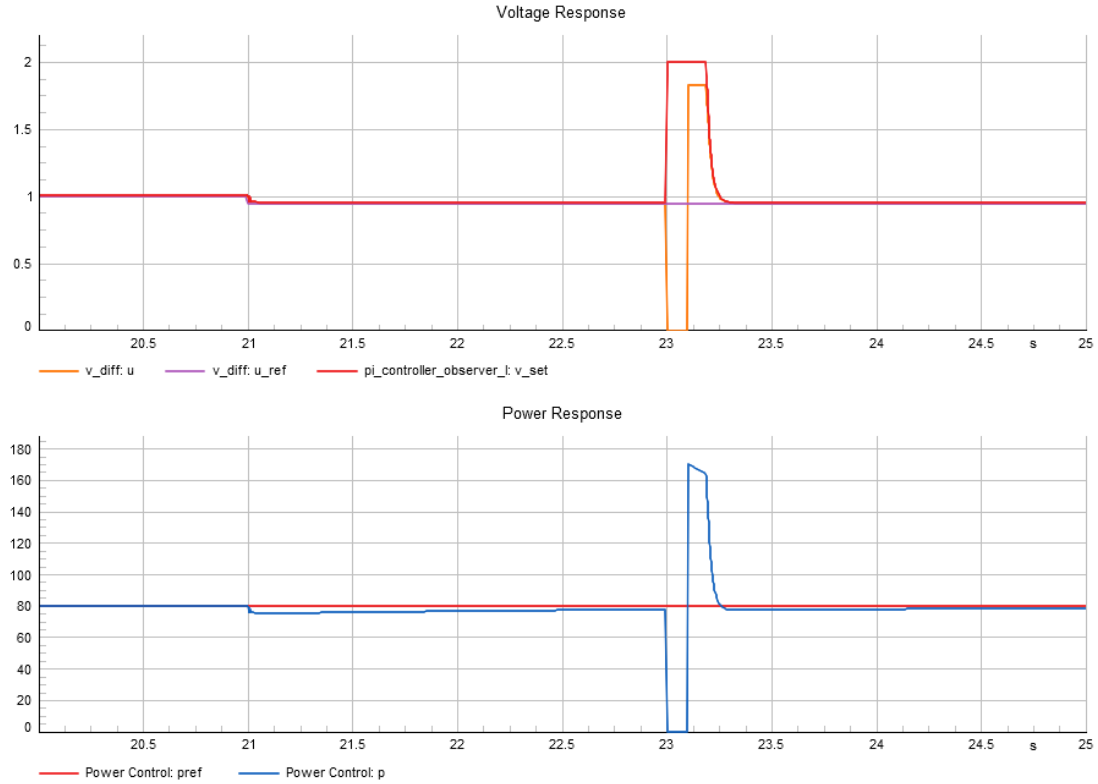


**FIGURE 2.9 VOLTAGE AND POWER RESPONSE UNDER GRID-FORMING CONTROL USING FMU-MODELICA BASED MODEL**

### MMC in grid-forming mode with DSL modelling

As part of this project, one of the models developed is a Modular Multilevel Converter (MMC) and a 2-level Voltage Source Converter (VSC) operating in grid-forming (GFM) mode, implemented using DSL. This contribution combines and extends existing models from the DIgSILENT PowerFactory library.

Specifically, the GFM control scheme was adapted from the DIgSILENT template “Virtual Synchronous Machine” which represents a "Static Generator" (an “ElmGenStat” element) controlled in GFM mode. However, the Static Generator model does not represent the DC side in the network, impeding the connection of arbitrary DC grids. To overcome this limitation, the "Static Generator" has been replaced with a "PWM Converter" (“ElmVsc” element), which allows for the connection of arbitrary DC sources or grids. Additionally, the PWM Converter supports both MMC and 2-level VSC technologies.

Figure 2.10 highlights the modifications made to the original model. To incorporate the PWM Converter, the Static Generator block was replaced. Furthermore, a modulation block and a DC voltage measurement system were adapted from DIgSILENT example models, specifically the “Offshore Wind Farm” application example. Similar to the “Virtual Synchronous Machine” template, the

implementation presented here includes blocks for: Measuring electrical values, AC voltage control, GFM control, a virtual impedance block, and overcurrent and overvoltage protection.

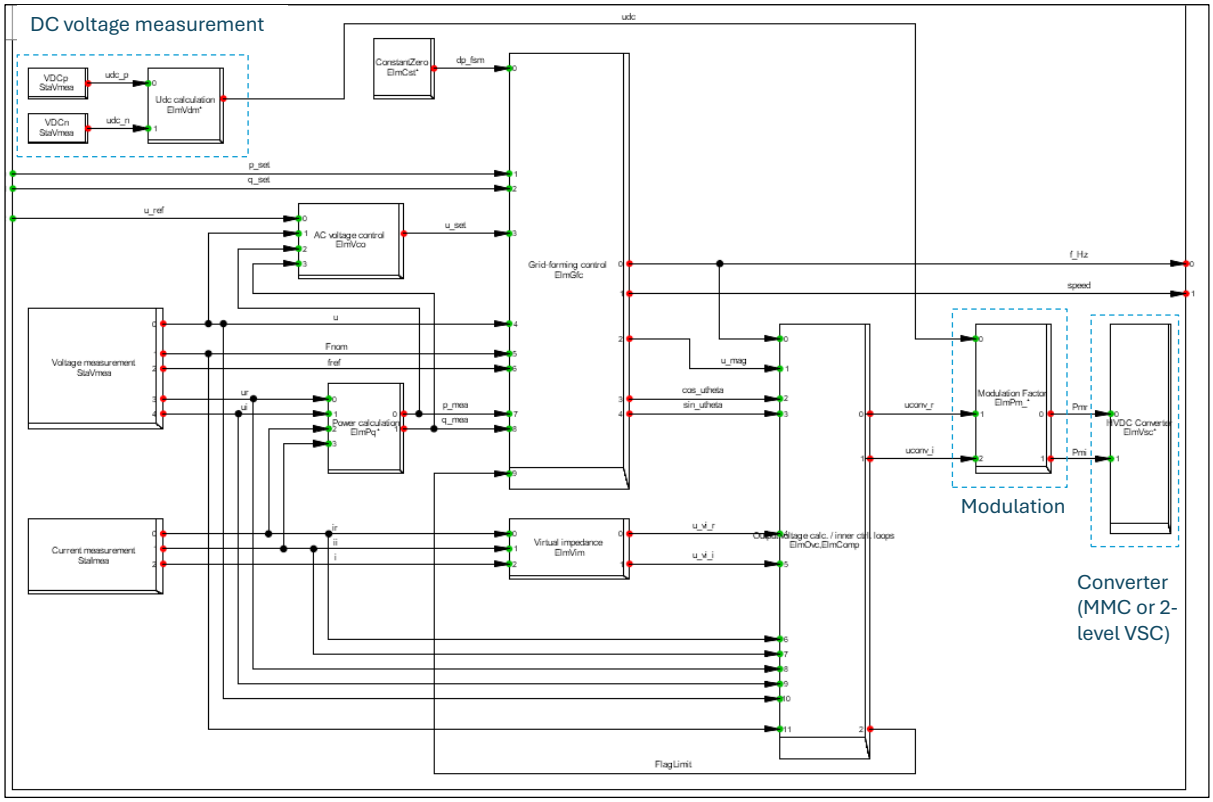


FIGURE 2.10 IMPLEMENTATION OF GFM CONVERTER IN DSL

To test the model, a simple test system, as shown in Figure 2.11, was implemented. The test setup consists of a 1000 MVA MMC operating in GFM mode, connected to a grid via two AC lines. With the modifications introduced, it is now possible to connect arbitrary DC sources or grids on the DC side.

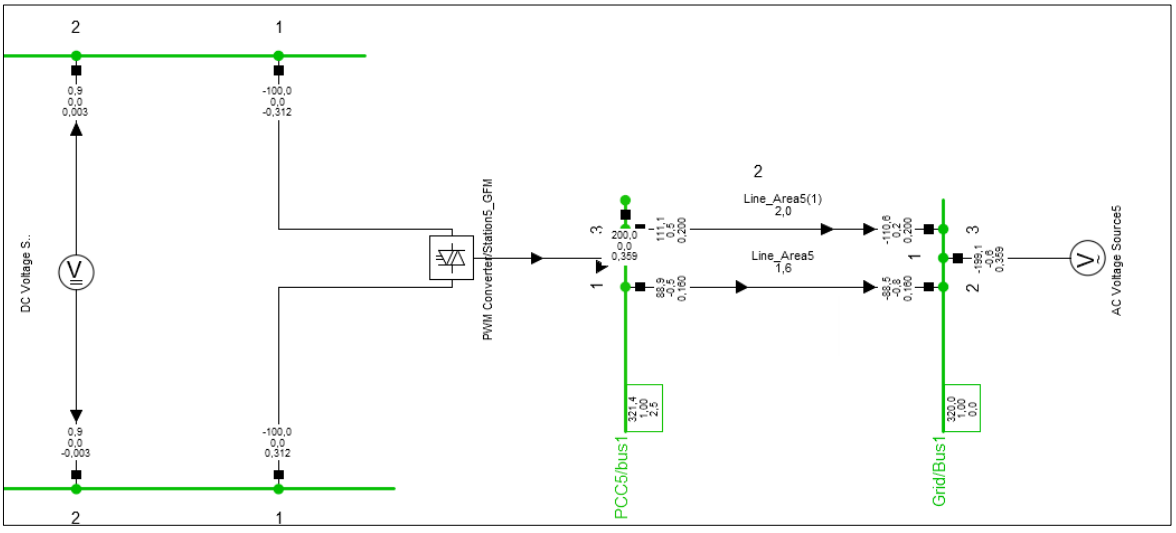


FIGURE 2.11 SIMPLE TEST SYSTEM

Figure 2.12 displays the active and reactive power responses for two scenarios:

1. A change in the active power reference from 200 MW to 300 MW
2. The disconnection of one of the AC lines.

The GFM parameters used for this test are H=2 s and D=50 pu, the other parameters used for this simulation being the default parameters of the reused blocks from the DIgSILENT PowerFactory library.

As expected, the active power follows the step reference with dynamics determined by the control parameters. Upon the disconnection of one AC line, the system exhibits the expected behaviour: the line trip induces an angle jump, prompting the converter—operating as a voltage source—to apply a corresponding active power phase adjustment. For the reactive power, which is controlled by maintaining a constant voltage magnitude, the trip results in a change in the reactive power injected by the converter.

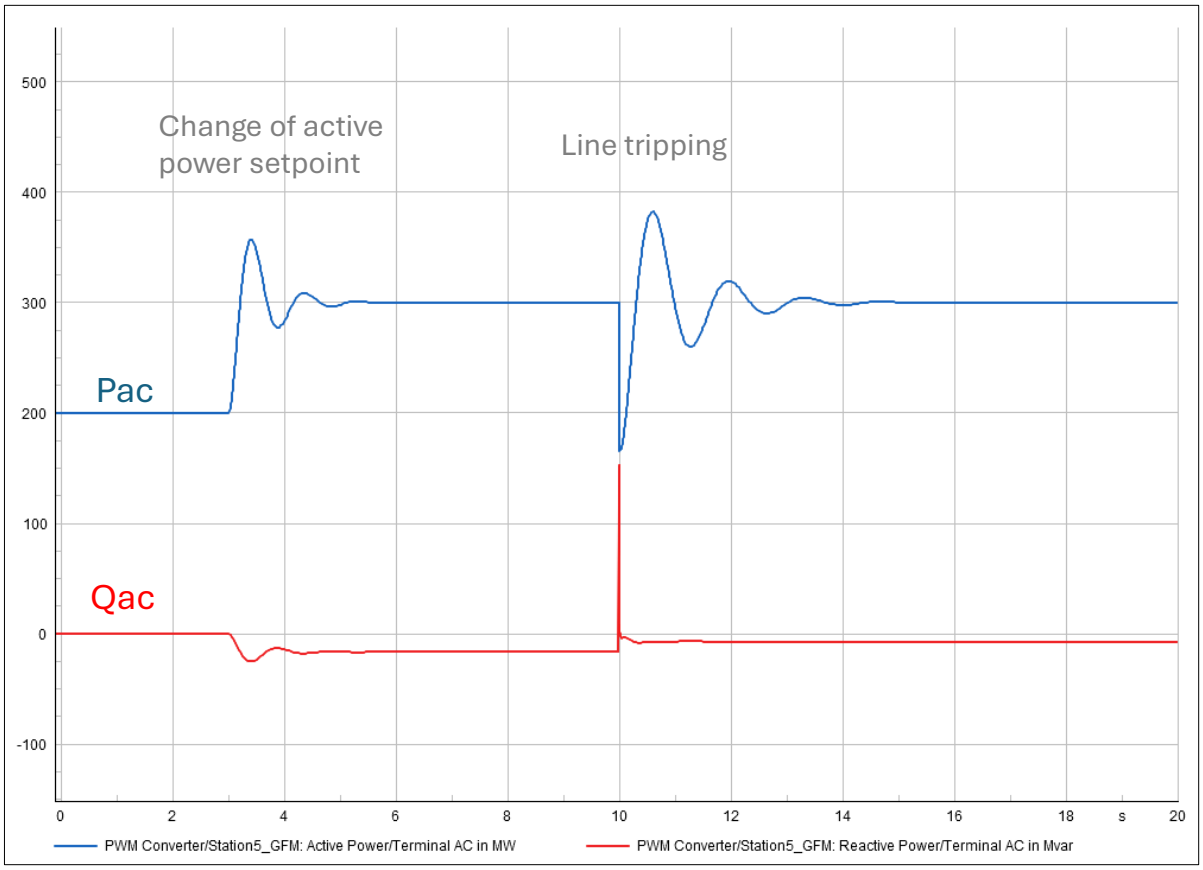


FIGURE 2.12 ACTIVE AND REACTIVE POWER OF THE GFM MMC

## 2.2.2 Energy dissipating system

Energy dissipating systems in HVDC system allow for the dissipation of surplus energy when a fault occurs on the AC side of a receiving terminal. These systems ensure that even if the power generated by the sending terminal remains unchanged for a short time, the continuous rise of the DC bus voltage during the fault is avoided. This section considers DC-connected energy dissipating systems, aka dynamic braking systems (DBSs). The model developed in Matlab Simulink and DIgSILENT PowerFactory is presented and exemplary results are shown.

As a model to be used in RMS simulations, i.e., with time steps in the range of a few ms, the modelling approach aims to represent the dissipated energy but not the fast dynamics (typically the switch commutations). Also, the model can represent the behaviour of the various topologies presented in deliverable D4.1 [1] section 2.4.1. The model of the DBS is presented in Figure 2.13. The device is modelled by a controlled direct-current source to be connected between DC terminals. The current

value is computed to dissipate power according to a characteristic depending on the DC voltage. Basically, the higher the DC voltage, the higher the dissipated power. Further details are given below.

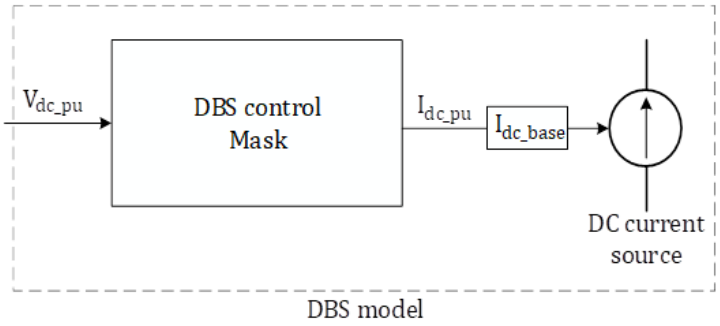


FIGURE 2.13 DBS CONTROL MODEL.

$V_{dc_{pu}}$ represents the voltage between DC terminals and is in per unit (p.u). $I_{dc_{pu}}$ represents the output current of the DBS and is also in per unit (p.u). $I_{dc_{base}}$ represents the nominal current of the HVDC network in international system (SI) i.e., ampere. The mask parameters are shown in Table 2.3.

TABLE 2.3 DBS MASK PARAMETERS

| VARIABLE NAME | DESCRIPTION | DEFAULT VALUES |
|---|---|---|
| $V_{dc_{th}}$ | DC voltage threshold | 1.05 pu |
| $V_{dc_{max}}$ | Maximum acceptable DC voltage | 1.1 pu |
| $P_{dc_{max}}$ | Maximum dissipated power | 1 pu |
| $E_{th}$ | Energy threshold | 1 s |
| $E_{max}$ | Maximum acceptable energy | 1.1 s |
| $tr_{Idc}$ | First-order filter time constant for slowing down the DBS output current | 5 s |
| $t_{reset_E}$ | Delay before dissipated energy is reset to zero | 1 s |

The expected behaviour is as follows. When the DC voltage exceeds $V_{dc_{th}}$, the DBS dissipates power proportionally to the voltage difference between $V_{dc_{th}}$ and $V_{dc_{th}}$(i.e. when the DC voltage reaches $V_{dc_{max}}$, the dissipated power $P_{dc} = P_{dc_{max}}$) as shown in Figure 2.14.

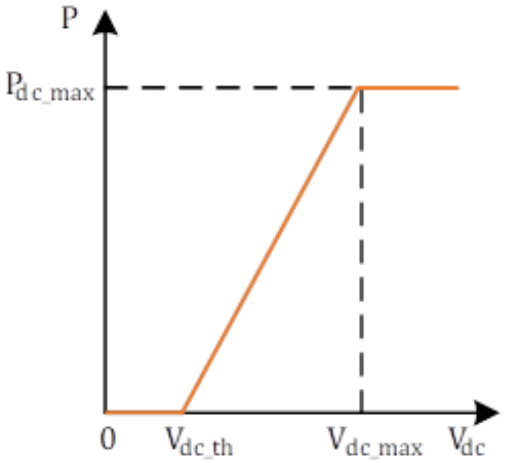


FIGURE 2.14 DBS POWER DISSIPATION ACCORDING TO THE DC VOLTAGE

A first-order filter is placed at the output of the block to model the dynamic of the current absorbed by the DBS. The parameter $\mathrm{tr_{ld}}$ represents the time constant of this filter.

Also, the model represents the limited energy dissipation capability, i.e., a DBS cannot dissipate energy in steady-state operation. To implement this, when the DBS dissipates power, the dissipated energy is calculated and when it crosses the threshold $E_{th}$, the maximal dissipated power is reduced until it reaches zero once the dissipated energy reaches $E_{max}$, as shown in Figure 2.15. This simulates a physical limit (the heating of the system and the energy that can be dissipated). The dissipated energy is then reset to zero after a time $t_{reset_E}$ in seconds when the DBS is no longer dissipating power ($V_{dc_{pu}}$<$V_{dc_{th}}$), representing that the heated elements have cooled down and that the DBS can be used again. The DBS power limitation strategy is done in Matlab Simulink thanks to a MATLAB function *Power_limitation* illustrated in Figure 2.14. The "sampling and hold" block maintains the DBS power limit when the energy exceeds $E_{th}$. It should be noted that the energy threshold is set at 1 s, which means that the DBS can dissipate the rated power for 1 s.

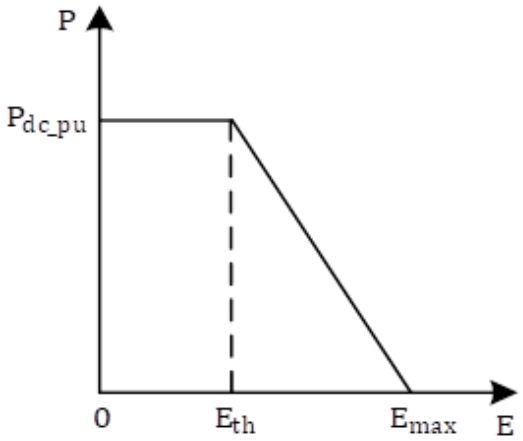


FIGURE 2.15 DBS POWER LIMITATION STRATEGY ACCORDING TO THE ENERGY DISSIPATED

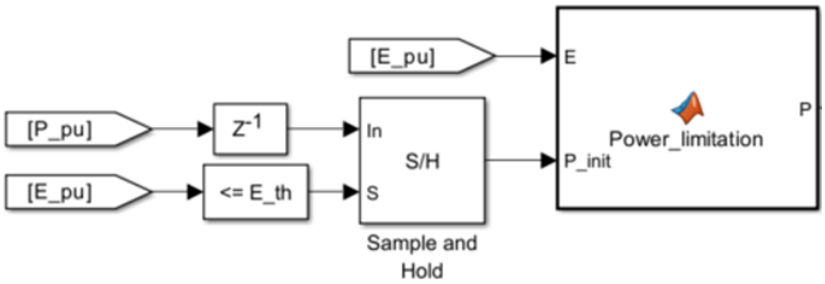


FIGURE 2.16 DBS POWER LIMITATION BLOCK IN SIMULINK

To validate the effectiveness of the DBS control, a simulation was carried out with a $V_{dc_{pu}}$ profile where there is an overvoltage between 2 and 7 s. The results obtained are shown in Figure 2.17. It can be seen that between 0 and 2 s, the DBS dissipates no power because the voltage is below the threshold. However, from 2 s onwards, the DBS starts to dissipate energy when the voltage exceeds the threshold voltage $V_{dc_{th}}$. Between 2 and 6.1 s, the DBS absorbs power proportionally to the difference between $V_{dc_{th}}$ and $V_{dc}$. At 6.1 s, the energy dissipated reaches the threshold energy and the DBS power is reduced to zero when the dissipated energy reaches $E_{max}$ slightly after 7s. After 8 s, the DC voltage drops to 1 p.u, below the threshold. The DBS absorbs no power and the energy dissipated remains at $E_{max}$, until 8 s at which time the dissipated energy is reset to zero.

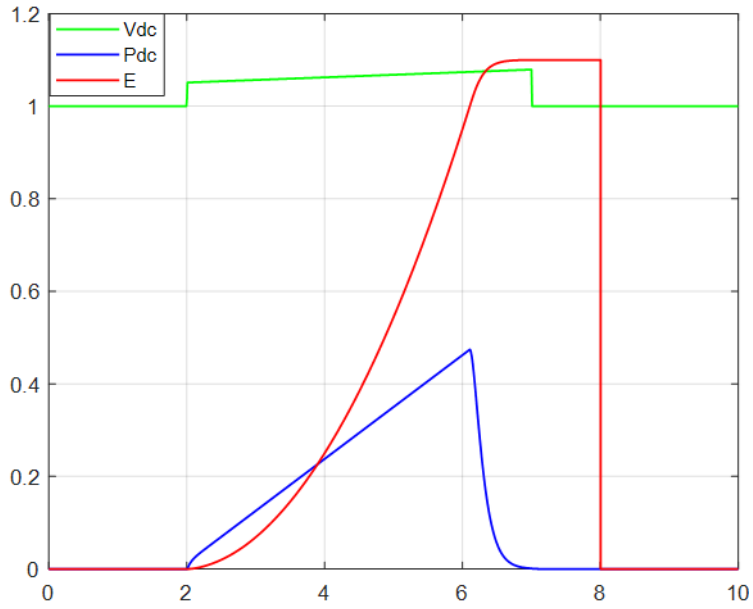


FIGURE 2.17 RESULTS OF THE DBS CONTROL STRATEGY

To implement the above system in DIgSILENT PowerFactory, the FMU based co-simulation has been used. The built dissipating system is exported from Matlab Simulink as an FMU and imported into DIgSILENT PowerFactory as a modelica model. To verify the effectiveness of the built dissipating system in DIgSILENT PowerFactory, a DC grid has been built with two DC voltage sources and one DC transmission line and the dissipating system represented by the DC current source.

Figure 2.18 verifies the FMU-based configuration, showing voltage variations (red and green lines) and current generation from the DC current source (purple line), demonstrating the same performance as shown in the provided Matlab Simulink model.

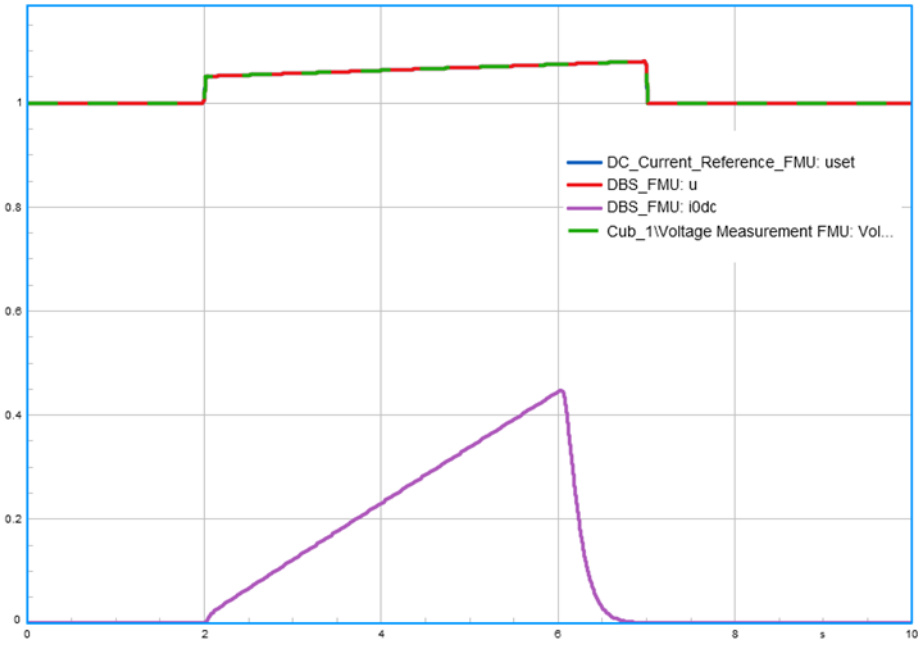


FIGURE 2.18 VOLTAGE AND CURRENT VARIATION IN POWERFACTORY

### 2.2.3 Energy storage

Energy storage systems (ESS) can enhance the reliability of service in power systems with a high share of renewable energy sources. This section considers a converter topology that can integrate ESS directly into an HVDC system. This topology has been selected among the candidate technologies listed in section 2.5.2 of [1]. Its model in DIgSILENT PowerFactory is presented and exemplary results are shown.

The topology (see Figure 2.19) consists of a branch of energy storage submodules (ES-SMs) and an inductor [2]. The ES-SMs are made of half-bridge submodules ($S_1$, $S_2$ and $Csm$) connected to a storage element (battery or supercapacitor) via a DC/DC converter ($S_3$, $S_4$ and $L_{sm}$). Thanks to this DC/DC converter, the energy storage elements can then be used over their complete voltage range. The topology is scalable to different voltage levels and, since the energy storage elements are distributed among all submodules, it provides a high degree of modularity. This topology provides less footprint than using two or more branches, at the cost of requiring the same type of storage in each ES-SMs and no possibility to use AC circulating current to balance the modules. This approach is especially suited for applications in which the storage devices of a single branch are enough to provide the target power and energy.

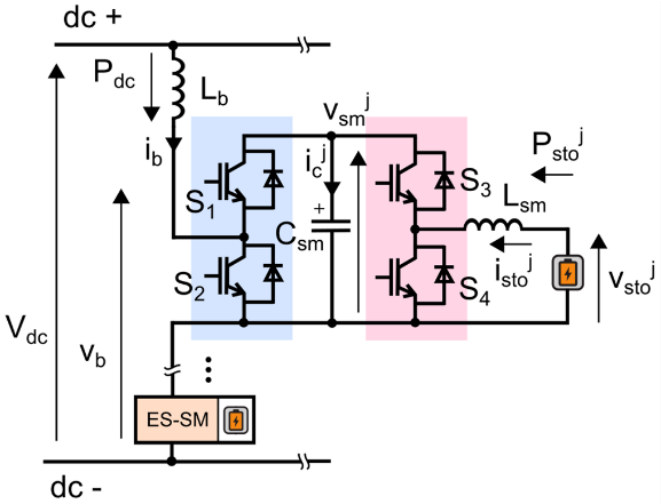


FIGURE 2.19 SCHEME OF THE ES-SM BRANCH DETAILING THE STRUCTURE OF THE ONE OF THE ES-SMS.

The ES-SM branch can inject or absorb power from the HVDC system. The sign of $Pdc$ determines the sign of $i_b$. The control is designed to regulate the injected DC power and the energy in the capacitors of the ES-SMs. In each ES-SM, switches $S_1$ to $S_4$ are turned on or off to control the power exchange between the energy storage elements and the HVDC terminals and to balance $v_{sm}^j$, the voltages of each submodule capacitor.

For system-level simulations, considering each switch, each capacitor, internal balancing algorithms, etc., brings an unnecessary complexity and an *average arm* modelling approach can be used as for MMCs and as shown in Figure 2.20. In such a model, the energy stored in all submodule capacitors is modelled with a single equivalent capacitor ($C_{eq}$) with an equivalent voltage $v_c^b$. The power absorbed by the chain of ESSMs is transferred to the equivalent capacitor ($v_b i_b = v_c^b i_{c,b}$). This is controlled by a modulation index $m_b$, modelling the average state of switches $S_1$ and $S_2$ in all ESSMs, and it can be shown that $v_b = m_b v_c^b$. The energy in the equivalent capacitor $C_{eq}$ evolves also according to the power exchange with the batteries or supercapacitors, which are also modelled as lumped elements.

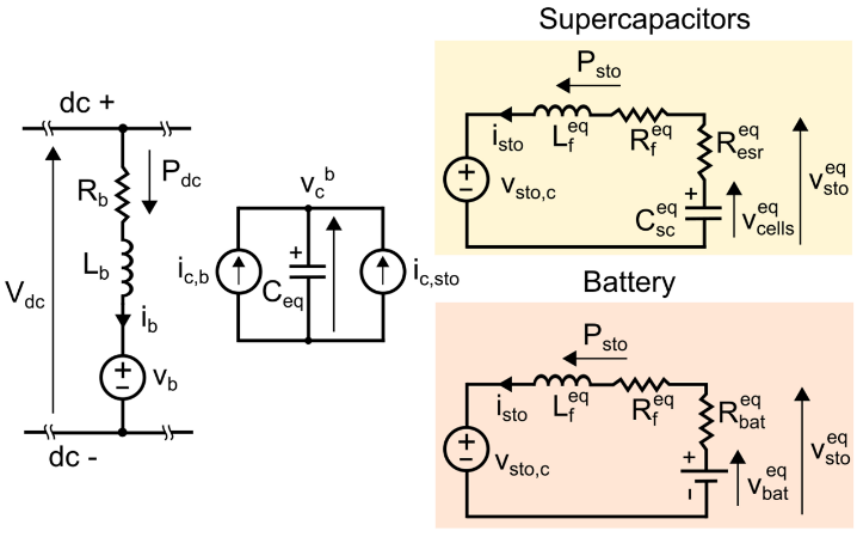


FIGURE 2.20 AVERAGE MODEL OF THE ES-SM BRANCH CONSIDERING SUPERCAPACITORS OR BATTERIES AS STORAGE ELEMENTS

The main assumption in this model is that all the ES-SMs are identical and that their voltage is totally balanced. Then, the voltages of the capacitors in the ESSMs $v_{sm}^{i}$ can be modelled as a single voltage for the whole branch. The same applies to the insertion and bypass orders, they can be averaged in a single variable (modulation index). Based on the previous, the stack of ES-SMs is represented by an equivalent voltage source ($v_b$), which models the voltage generated by the inserted ES-SMs. Separately, in another circuit, the capacitances of the ES-SMs are represented as one equivalent capacitor ($C_{eq}$) with a voltage ($v_c{}^b$) that corresponds to the sum of all the voltages at the terminals of ES-SMs capacitors. On one side, the capacitor is connected to a current source ($i_{c,b}$) that models the charge and discharge from the branch current ($i_b$). The current source is coupled with the voltage source ($v_b$). The model equations and more details on this system are available in [3].

$P_{dc}$ is controlled using switches $S_3$ and $S_4$, which, in the average model, corresponds to the duty cycle of the equivalent DC/DC converter, $D_b$. The energy in the equivalent capacitor is regulated with $S_1$, $S_2$, which corresponds to the modulation index, $m_b$. Both control schemes are described below.

The control scheme to regulate the energy of the equivalent capacitor is depicted in Figure 2.21.

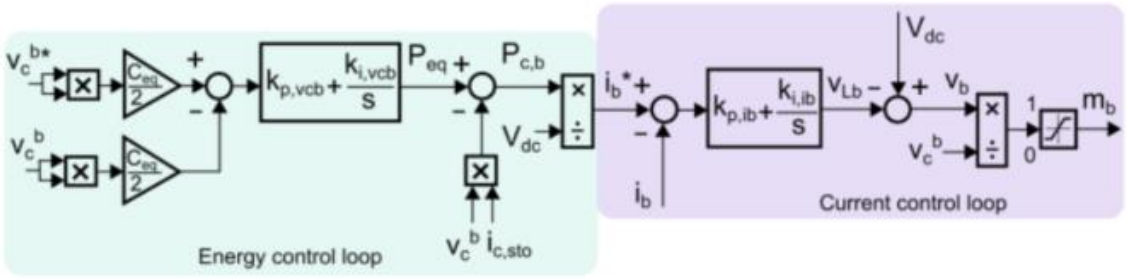


FIGURE 2.21 CONTROL SCHEME TO REGULATE THE ENERGY OF THE EQUIVALENT CAPACITOR WITH TWO CASCADED CONTROLLERS: ENERGY AND CURRENT CONTROLLER

The control scheme to regulate the DC power is shown in Figure 2.22. It is composed of two cascaded control loops: an outer power loop and an inner current loop.

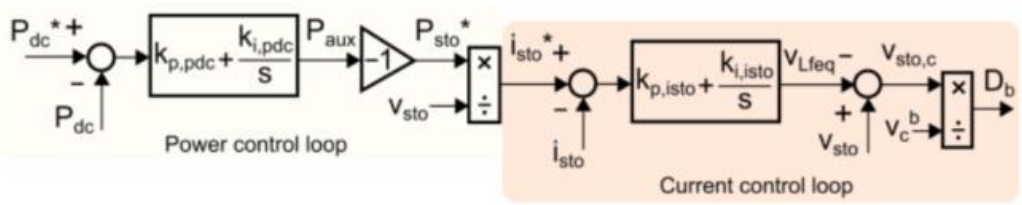


FIGURE 2.22 CONTROL SCHEME TO REGULATE THE DC POWER, WITH TWO CASCADED CONTROLLERS: POWER AND CURRENT CONTROLLER.

The control scheme and the modelling of the ES-SMs have been previously built in the Matlab Simulink environment. To model the ESS in DIgSILENT PowerFactory, it has been decided to use a simple electric model with a branch inductor and a controlled voltage source and import the control and ES-SM model from Matlab SImulink (see Figure 2.23) thanks to the Function Mock-up Unit (FMU) standard. The *fmu* then includes the models for the supercapacitors, current control loops, energy on submodule capacitors, etc.). The DIgSILENT PowerFactory simulations were then compared to the Matlab Simulink simulations.

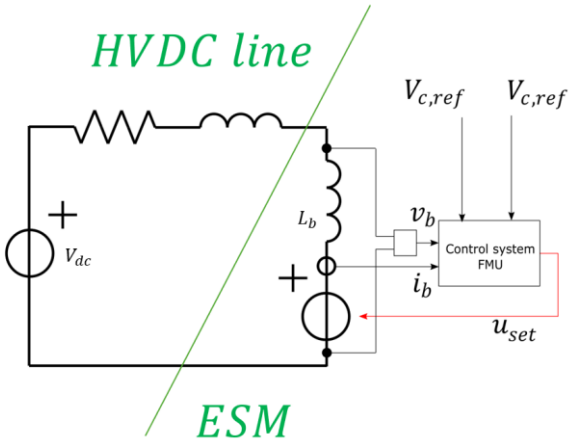


FIGURE 2.23 POWERFACTORY MODEL

The main parameters of the simulated system are listed in Table 2.4.

TABLE 2.4 DC STORAGE MODEL USER DEFINED PARAMETERS

| PARAMETER | DESCRIPTION | DEFAULT VALUE | COMMENT |
|---|---|---|---|
| $P_{branch}$ | Nominal power of the branch of ES-SMs | $200\ MW$ | Impacts most of the values in the model. |
| $V_{dc}$ | DC voltage (between converter terminals) | $640\ kV$ | Impacts mostly the number of submodules. |
| $m_{max}$ | Maximum modulation factor in steady state operation | $0.9$ | Impacts mostly the number of submodules. |
| $V_{ESM}$ | Average voltage of the capacitor in a submodule | $3800\ V$ | Impacts mostly the number of submodules. |
| $\eta_{ind}$ | Efficiency of the inductor in the ESM branch | $0.995$ | Used to calculate $R_b$. |
| $I_{max.switches}$ | Maximum peak current of the switches | $2400\ A$ | Used to calculate $L_b$. |

| | | | |
|---|---|---|---|
| $k_0$ | Ratio between the current threshold after which the blocking is triggered and the nominal branch current | 1.25 | Used to calculate $L_b$. |
| $t_{delay}$ | Time in between the detecting of the fault and the blocking of the branch | $300\ \mu s$ | Used to calculate $L_b$. |
| $V_{ESM,sto}$ | Nominal voltage of the storage device in each submodule | $1450\ V$ | Used for batteries as $V_{bat}^{eq}$. Used for Supercapacitors to calculate $V_{sc,max}$ and $V_{sc,min}$ and then $C_{sc}^{eq}$. |
| | **FOR SUPERCAPACITORS ONLY** | | |
| $W$ | Usable stored energy (energy deviation between $V_{sc,min}$ and $V_{sc,max}$) | $540\ MJ$ | Used to calculate $C_{sc}^{eq}$. |
| $\eta_{sc}$ | Efficiency of the pack of supercapacitors. | 0.98 | Used to calculate $R_{esr}^{eq}$. |
| $k_1$ | Ratio between $V_{sc,max}$ and $V_{sto}$ | 0.9 | Used to calculate $V_{sc,max}$. |
| $k_2$ | Ratio between $V_{sc,min}$ and $V_{sc,max}$ | 0.4 | Used to calculate $V_{sc,min}$. |
| $k_3$ | Ratio between $V_{sc,discharge,lim}$ and $V_{sc,min}$ | 0.7 | Used to calculate $V_{sc,discharge,lim}$. If the energy storage voltage is lower than $V_{sc,discharge,lim}$, the power that can be provided is limited to a value proportional to the energy storage voltage. |
| | **FOR BATTERIES ONLY** | | |
| $P_{loss,pu}$ | Losses in pu at maximum current | $0.015\ pu$ | Used to calculate $R_{bat}^{eq}$. |
| | **STORAGE-SIDE DC/DC CONVERTER** | | |
| $R_{I_{Lf}}$ | Percentage in pu of the allowed current ripple in the inductance Ldcdc. | $0.15\ pu$ | Used for the calculation of $L_{dcdc}$. |
| $f$ | Switching frequency for switches $S_3$ and $S_4$. | $2\ kHz$ | Used for the calculation of $L_{dcdc}$. |
| | **CONTROL LOOP PARAMETERS** | | |
| $\tau_{V_{dc}}$ | Time constant of the $V_c^b$ control loop | $50\ ms$ | Used to calculate the cut of frequency $\omega_c$. |
| $\psi$ | Damping of the $V_c^b$ control loop | 0.707 | Used to calculate $K_{pVc}$ and $K_{iVc}$. |
| $\tau_{b,Ireg}$ | Time constant of the $I_b$ control loop | $10\ ms$ | Used to calculate $K_{pP}$ and $K_{iP}$. |

| | | | |
|---|---|---|---|
| $\tau_{b,Isc}$ | Time constant of the $I_{sto}$ control loop | $2\ ms$ | Used to calculate $K_{pIsc}$ and $K_{iIsc}$. |

TABLE 2.5 DC STORAGE MODEL DERIVED PARAMETERS

| PARAMETER | DESCRIPTION | CALCULATION |
|---|---|---|
| $I_{bn}$ | DC nominal branch current | $P_{branch}/V_{dc}$ |
| $P_{ind,loss}$ | Max losses in the branch inductor | $P_{branch}(1-\eta_{ind})$ |
| $R_b$ | Parasitic resistance of the branch inductor | $P_{ind,loss}/\left(\frac{P_{branch}}{V_{dc}}\right)^2$ |
| $V_c^b$ | DC voltage of the equivalent capacitor of the ES-SMs obtained from fixing a maximum modulation factor in steady state operation | $\frac{V_{dc}+R_bI_{bn}}{m_{max}}$ |
| $I_{th}$ | Current threshold after which the blocking is triggered | $k_0 I_{bn}$ |
| $L_b$ | Inductance of the ESM branch, sized to ensure the DC fault current does not exceed the rating of the switch | $\frac{V_{dc}t_{delay}}{I_{max,switches}-I_{th}}$ |
| $N_{ESM}$ | Number of submodules | $ceil(V_c^b/V_{ESM})$ |
| $V_{sto}$ | Maximum voltage of the pack storage devices | $V_{ESM,sto}N_{ESM}$ |
| | **DC/DC CONVERTER** | |
| $I_{nLf}$ | Nominal current through the inductor in the LV DC/DC | $\frac{P_{branch}}{N_{ESM}/V_{ESM,sto}}$ |
| $L_{dcdc}$ | Inductance to comply with the current ripple in the LV DC/DC | $\frac{V_{ESM}}{4fR_{I_{Lf}}I_{nLf}}$ |
| | **CONTROL SYSTEM LOOP PARAMETERS** | |
| $\omega_c$ | Cut off frequency of the $V_c^b$ control loop | $3/\tau_{Vdc}$ |
| $K_{pP}$ | Proportional gain of the $I_b$ control loop | $R_b/\tau_{bIreg}$ |
| $K_{iP}$ | Integral gain of the $I_b$ control loop | $L_b/\tau_{bIreg}$ |
| $K_{pVc}$ | Proportional gain of the $V_c^b$ control loop | $2\psi\omega_c$ |
| $K_{iVc}$ | Integral gain of the $V_c^b$ control loop | $\omega_c^2$ |
| $K_{pIsc}$ | Proportional gain of the $I_{sto}$ control loop | $L_f^{eq}/\tau_{bIsc}$ |
| $K_{iIsc}$ | Integral gain of the $I_{sto}$ control loop | $R_f^{eq}/\tau_{bIsc}$ |
| | **FOR SUPERCAPACITORS ONLY** (Supercapacitors modelled as a capacitor $C_{sc}^{eq}$ in series with a resistor $R_{esr}^{eq}$) | |

| | | |
|---|---|---|
| $V_{sc,max}$ | Maximum voltage of the pack of supercapacitors to absorb $P_{branch}$ | $k_1 V_{sto}$ |
| $V_{sc,min}$ | Minimum voltage of the pack of supercapacitors to provide $P_{branch}$ | $k_2 V_{sc,max}$ |
| $V_{sc,discharge,lim}$ | If the energy storage voltage is lower than $V_{sc,discharge,lim}$, the power that can be provided is limited to a value proportional to the energy storage voltage. | $k_3 V_{sc,min}$ |
| $V_{sc,charge,lim}$ | If the energy storage voltage is higher than $V_{sc,charge,lim}$, the power that can be absorbed is set to 0W to prevent further charge of the supercapacitors. | $V_{sto}$ |
| $C_{sc}^{eq}$ | Equivalent capacitance of the pack of supercapacitors | $2W/(V_{sc,max}^2 - V_{sc,min}^2)$ |
| $I_{max,sc}$ | Maximum current of the pack of supercapacitors | $P_{branch}/V_{sc,min}$ |
| $P_{sc,loss}$ | Max losses in the pack of supercapacitors at max current | $P_{branch}(1-\eta_{sc})$ |
| $R_{esr}^{eq}$ | Equivalent series resistance of the pack of supercapacitors | $P_{sc,loss}/I_{max,sc}^2$ |
| $R_f^{eq}$ | Parasitic resistance of the inductor | $\frac{P_{ind,loss}}{I_{max,sc}^2}$ |
| $L_f^{eq}$ | Equivalent inductance of the inductor in the average model of the DC/DC converter | $L_{dcdc} N_{ESM}$ |
| **FOR BATTERIES ONLY** (batteries modelled as a voltage source $V_{ESM,sto}$ in series with $R_{series}$) | | |
| $R_{bat}^{eq}$ | Series resistance with the DC source, obtained from the losses | $\frac{P_{loss,pu} P_{branch}}{\left(\frac{P_{branch}}{V_{sto}}\right)^2}$ |
| $R_f^{eq}$ | Parasitic resistance of the inductor | $\frac{P_{ind,loss}}{\left(\frac{P_{branch}}{V_{sto}}\right)^2}$ |
| $L_f^{eq}$ | Equivalent inductance of the inductor in the average model of the DC/DC converter | $L_{dcdc} N_{ESM}$ |

The simulation starts with no power absorption and with a voltage on the equivalent capacitor equal to $700\ kV$. At t=0.5 s the equivalent capacitor voltage reference jumps to $710\ kV$, while at t=1 s the power reference jumps to $10\ MW$ and at t=1.5 s it goes up to $100\ MW$ (in both cases the power flow direction is towards the storage system).

The DIgSILENT PowerFactory simulation results are shown in Figure 2.24 and Figure 2.25, whereas the Matlab SImulink results are presented in Figure 2.26.

It can be observed that the power and voltage waveforms are the same after the initial transients which depends on the initial conditions set in the two software platforms. This validates the FMU control system import from Matlab SImulink to DIgSILENT PowerFactory and its functioning in relation to the DIgSILENT PowerFactory power system model.

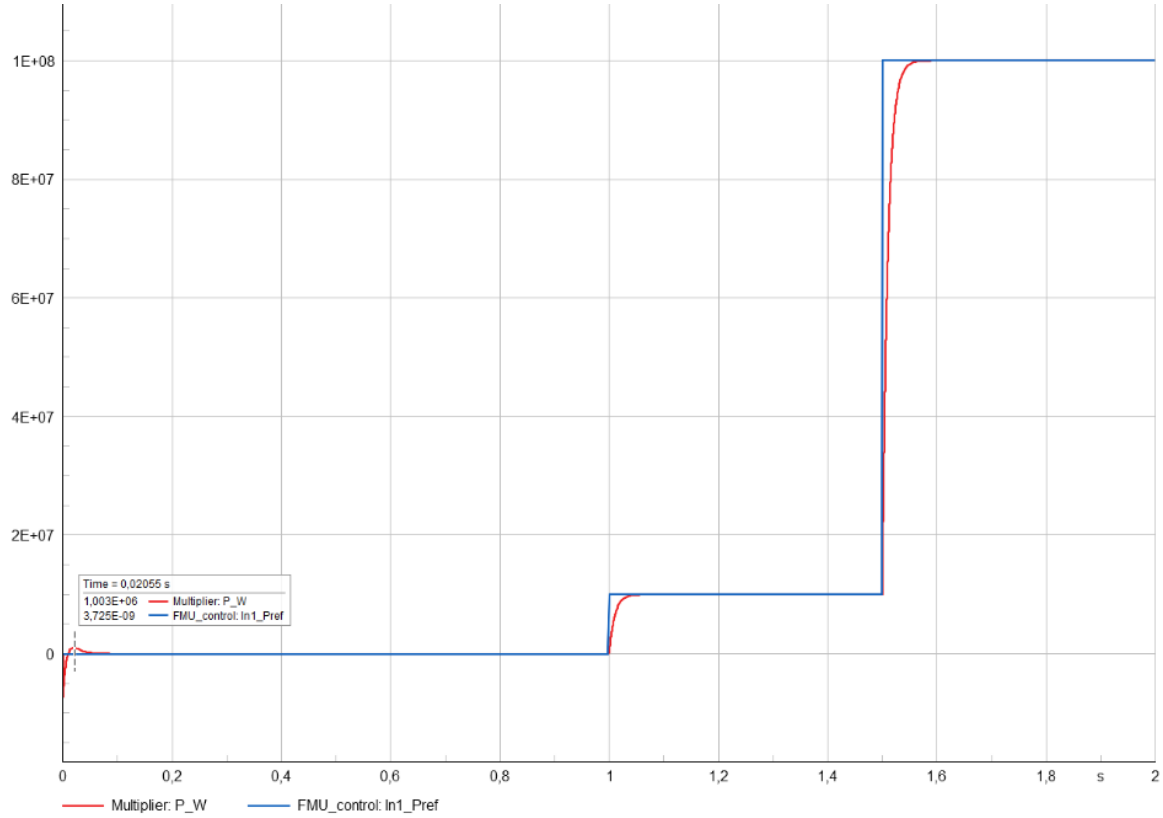


FIGURE 2.24 ESS POWER PROFILE IN POWERFACTORY

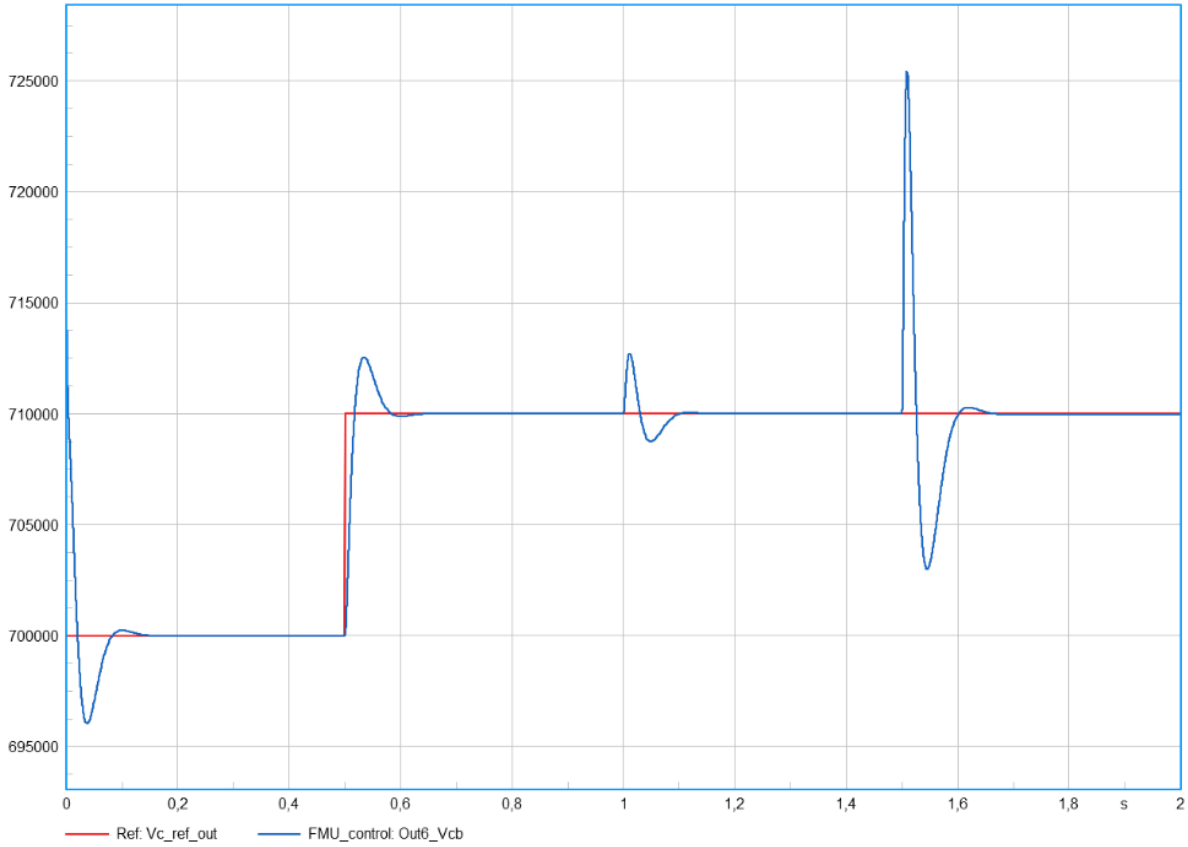


FIGURE 2.25 EQUIVALENT CAPACITOR VOLTAGE IN POWERFACTORY

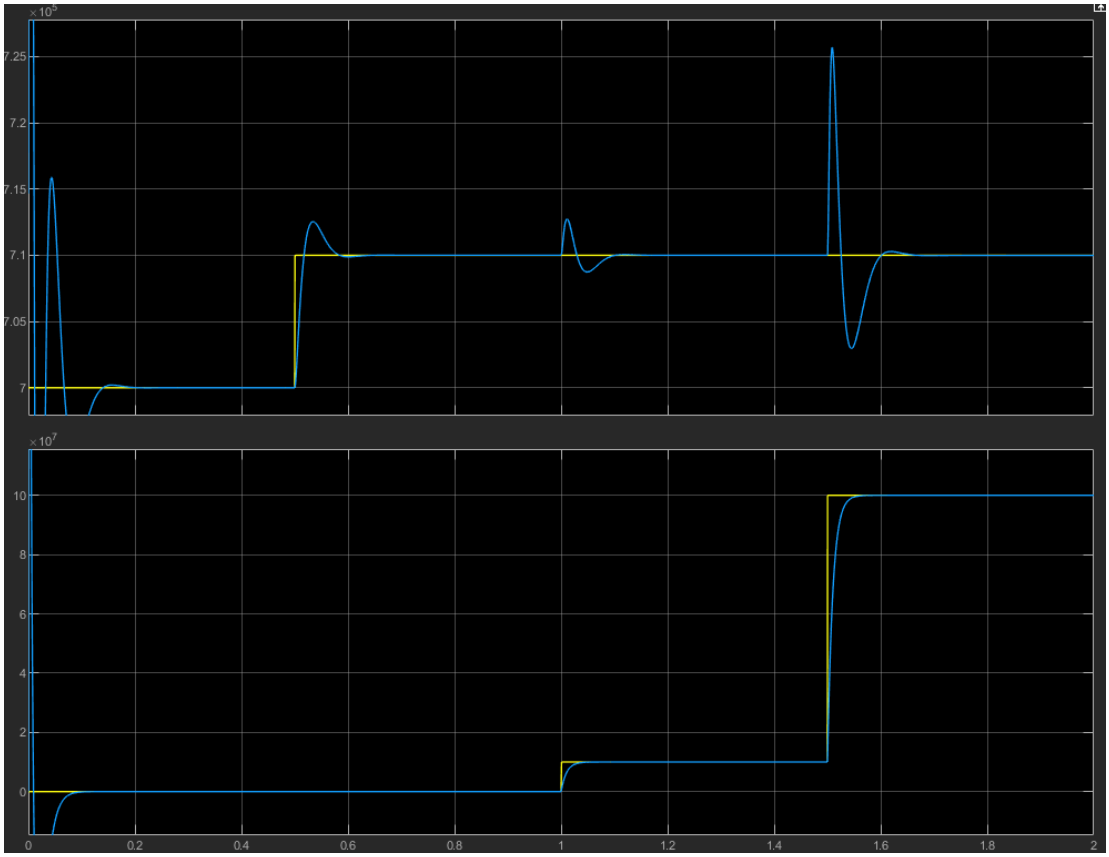


FIGURE 2.26 EQUIVALENT CAPACITOR VOLTAGE IN VOLTS (HIGH PANEL) AND ESM POWER IN WATTS (LOW PANEL)

### 2.2.4 Wind farm with wind turbines in grid-forming mode

Wind farms (WFs) are expected to become major sources of energy globally due to the target of achieving low-carbon development. The grid-forming (GFM) control in wind turbine (WT) converters can establish the offshore grid voltage and frequency while synchronizing the operation of hundreds of WTs. This section considers a wind farm under GFM mode. Its model in DIgSILENT PowerFactory is presented and exemplary results are shown.

An example of a 5 MW WF containing 2 wind turbines (WTs) with fully rated converters under GFM connected to a 20 kV network is shown in Figure 2.27. Each wind turbine has a nominal active power of 2.5 MW. The wind speed is set at 16 m/s.

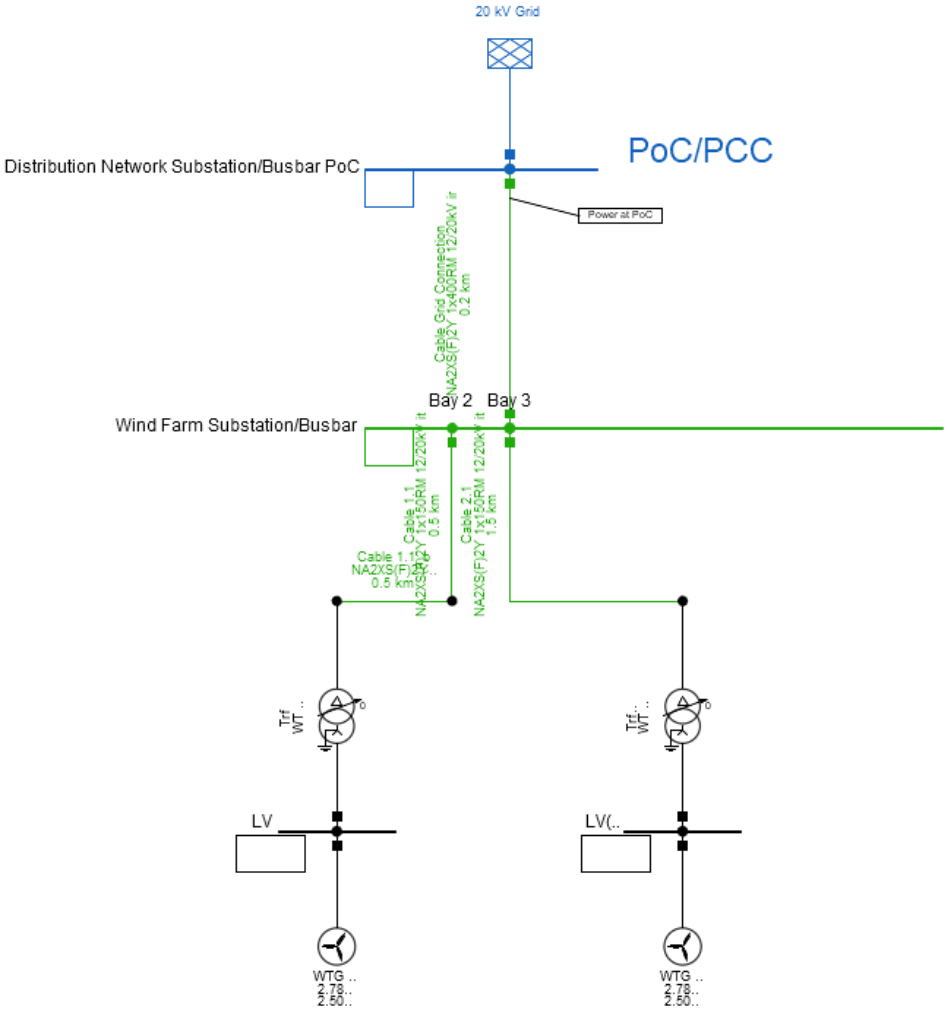


FIGURE 2.27 WIND FARM IN GFM MODE

The wind turbine model used in the example mainly represents the behaviour of the grid-side converter, as this dominates the electric behaviour of the wind turbine. The composite models, shown in Figure 2.28, of the wind turbines link each Static Generator with additional dynamic models which represent the turbine controller, over-frequency power reduction, protection, GFM and PLL. The Park Controller, Turbine Controller, Overfrequency Power Reduction and Protection block are based on the Template of a wind turbine model from the library.

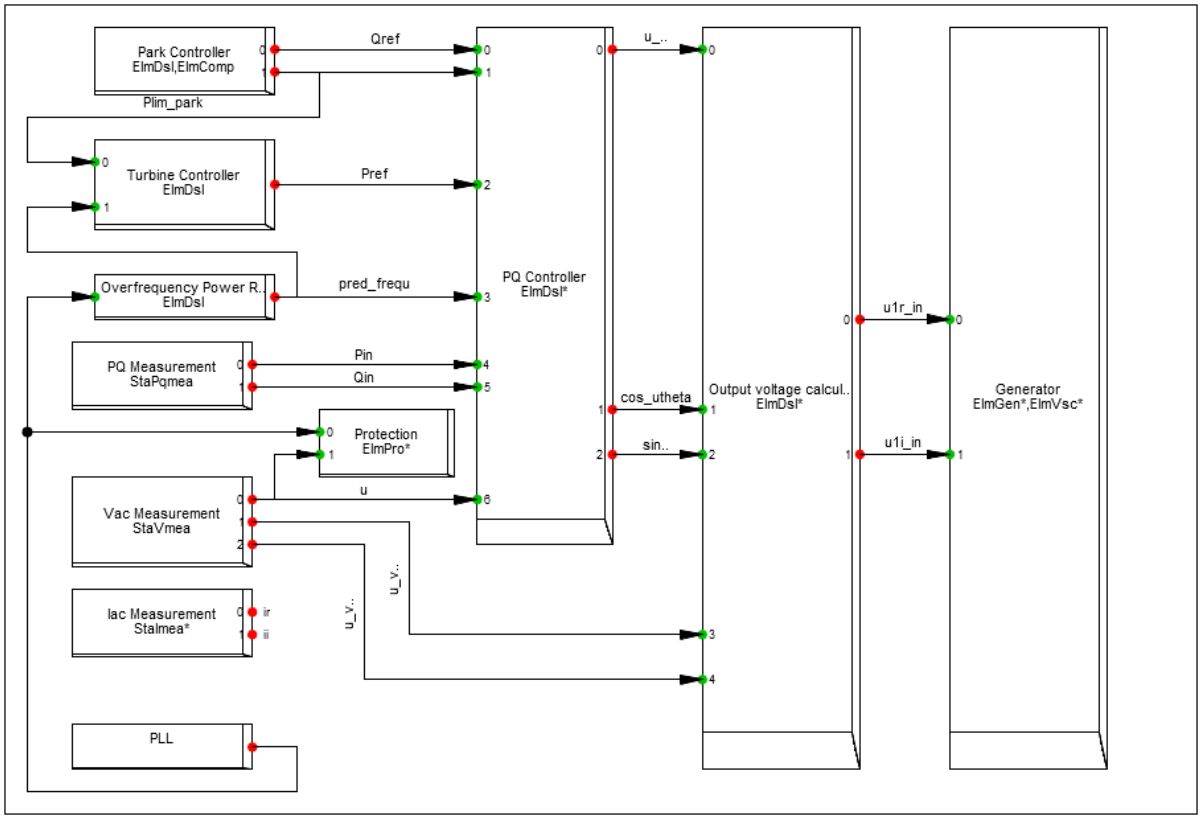


FIGURE 2.28 COMPOSITE MODEL OF A WIND TURBINE

The Park Controller block determines the individual dispatch of wind turbines. The Turbine Controller block determines the active power output of the corresponding wind turbines using their individual wind. The block Overfrequency Power Reduction used to emulate the characteristics of the WTG could reduce the active power during over-frequency conditions. The Protection block allows the wind turbine to be tripped in case of under-frequency or over-frequency conditions or under-voltage or over-voltage conditions.

The PQ block comprises a GFM controller implemented as a virtual synchronous machine (VSM), as shown in Figure 2.29. The active and reactive power references could be limited considering voltage and the wind farm park control signal. The PI controller in the VSM controllers now represents both damping with the proportional term and inertia with the integral term. The PI controller can be tuned accordingly to represent the dynamics of the power swing equation in a very simple form.

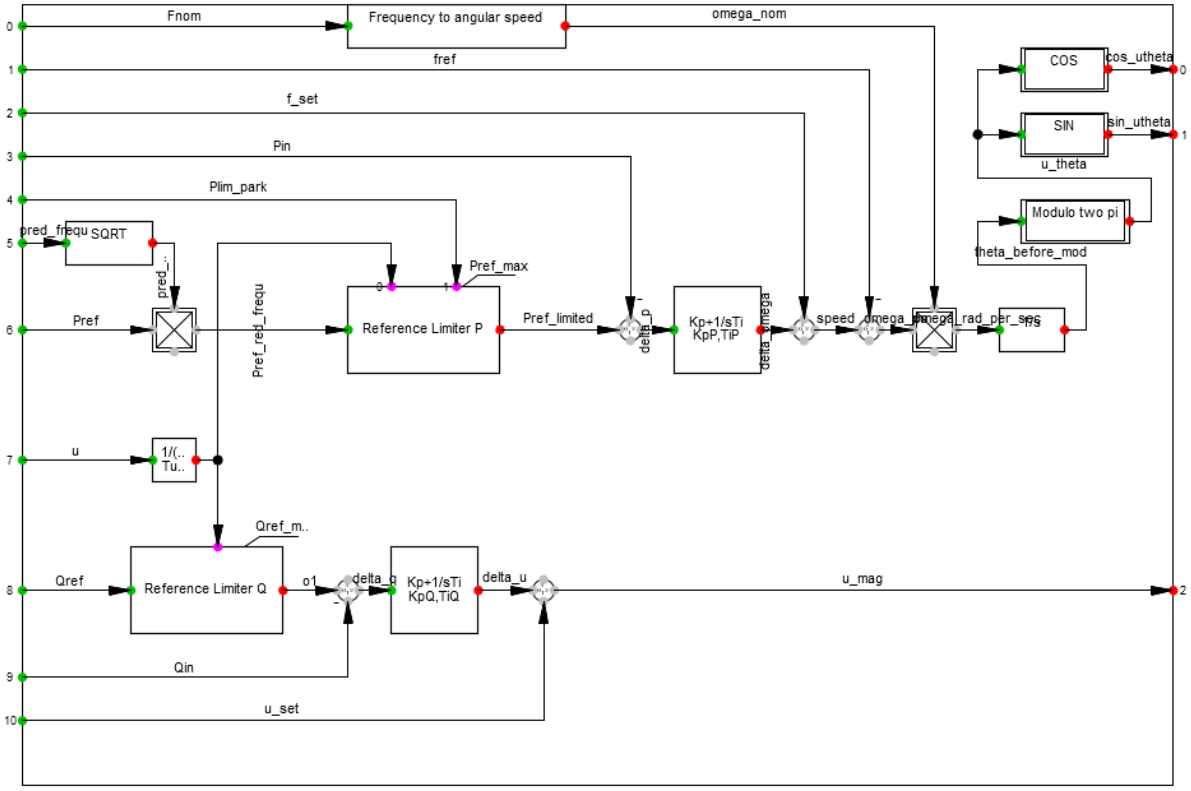


FIGURE 2.29 GFM VSM OUTER CONTROL

The main parameters of the simulated system are listed in Table 2.6.

TABLE 2.6 PARAMETERS OF THE SIMULATED WIND FARM WITH GFM WIND TURBINES

| PARAMETER | DESCRIPTION | DEFAULT VALUE |
|---|---|---|
| | **PARK CONTROLLER** | |
| $u_{min}$ | Minimum operation voltage in park controller | $0.9\ pu$ |
| $T_p$ | Filter time constant P measurement | $0.02\ s$ |
| $T_q$ | Filter time constant Q measurement | $0.02\ s$ |
| $T_u$ | Filter time constant U measurement | $0.01\ s$ |
| $K_{qp}$ | Proportional gain Q control | 2 |
| $K_{qi}$ | Integration gain Q control | 2 |
| $K_{pp}$ | Proportional gain P reduction | 1 |
| $K_{pi}$ | Integration gain P reduction | 0.875 |
| $q_{min}$ | Minimum reactive power | $-0.4\ p.u.$ |
| $q_{max}$ | Maximum reactive power | $0.4\ p.u.$ |
| $dq_{max}$ | Gradient limit of Q reduction | $1\ p.u./s$ |
| $dp_{max}$ | Gradient limit of P reduction | $1\ p.u./s$ |

| | | |
|---|---|---|
| | **TURBINE CONTROLLER** | |
| $T_{aero}$ | Smoothing time constant for wind speed | $1\ s$ |
| $T_{ctrl}$ | Time constant for turbine control and pitch | $3\ s$ |
| $v_{wind}$ | Wind speed | $16\ m/s$ |
| $v_{nom}$ | Wind speed at nominal operation | $16\ m/s$ |
| $v_{max}$ | Maximum operating wind speed | $25\ m/s$ |
| | **OVERFREQUENCY POWER REDUCTION** | |
| $f_{Up}$ | Start of Act. Power Reduction | $50.2\ Hz$ |
| $f_{Low}$ | End of Act. Power Reduction | $50.05\ Hz$ |
| $P_{Hz}$ | Gradient of Act. Power Reduction | $40\ \%/Hz$ |
| $T_{filter}$ | Input Filter Time Constant | $0.1\ s$ |
| $Grad_{neg}$ | Neg. Gradient for Power Change | $-0.25\ p.u./s$ |
| $Grad_{pos}$ | Pos. Gradient for Power Change | $0.25\ p.u./s$ |
| | **PROTECTION** | |
| $u_{max1}$ | Threshold slow overvoltage protection | 1.1 p.u. |
| $t_{Umax1}$ | Tripping delay slow overvoltage protection | 60 s |
| $U_{max2}$ | Threshold fast overvoltage protection | 1.15 p.u. |
| $t_{Umax2}$ | Tripping delay slow undervoltage protection | 0.1 s |
| $U_{min1}$ | Threshold fast undervoltage protection | 0.8 p.u. |
| $t_{Umin1}$ | Tripping delay overfrequency protection | 1.5 s |
| $U_{min2}$ | Threshold underfrequency protection | 0.45 p.u. |
| $t_{Umin2}$ | Tripping delay underfrequency protection | 0.8 s |
| $F_{max}$ | Threshold underfrequency protection | 51.5 Hz |
| $t_{Fmax}$ | Tripping delay overfrequency protection | 0.1 s |
| $F_{min}$ | Threshold underfrequency protection | 47.5 Hz |
| $t_{Fmin}$ | Tripping delay underfrequency protection | 0.1 s |
| $U_{low}$ | Minimum voltage for frequency protection | 0.8 p.u. |
| $T_{fblock}$ | Frequency protection blocking time after UVRT | 0.1 s |
| $T_{fMA}$ | Time window for moving average of frequency | 0.3 s |

| | | |
|---|---|---|
| $T_{max}$ | Maximum simulation time step | 0.02 s |
| | **VSM PQ CONTROLLER** | |
| $T_{umeas}$ | Filter time constant RMS voltage | 0.007 s |
| $K_{pQ}$ | Frequency setpoint | 0.05 p.u. |
| $T_{iQ}$ | Voltage setpoint | 1 p.u. |
| $K_{pP}$ | Nominal Frequency | 0.01 p.u. |
| $T_{iP}$ | Reference frequency (for RMS) | 1 p.u. |
| $P_{refmax}$ | Active power measurement | 1 p.u. |
| $Q_{refmax}$ | Reactive power measurement | 0.4 p.u. |

The dynamic simulation of Under-Voltage-Ride-Through (UVRT) behaviour is essential for validating a wind farm's performance during network faults. The study case demonstrates a scenario where a short circuit in a 20 kV network feeder near the wind farm's Point of Connection (PoC) causes voltage to drop to approximately 0% for 100 ms. The fault is cleared by opening the circuit breaker on the affected network feeder, with the wind farm's response measured through three key parameters at the point of coupling (PoC): voltage, active power, and reactive power. These measurements, illustrated in Figure 2.30, examines the wind turbines' ability to provide dynamic voltage support during short circuits.

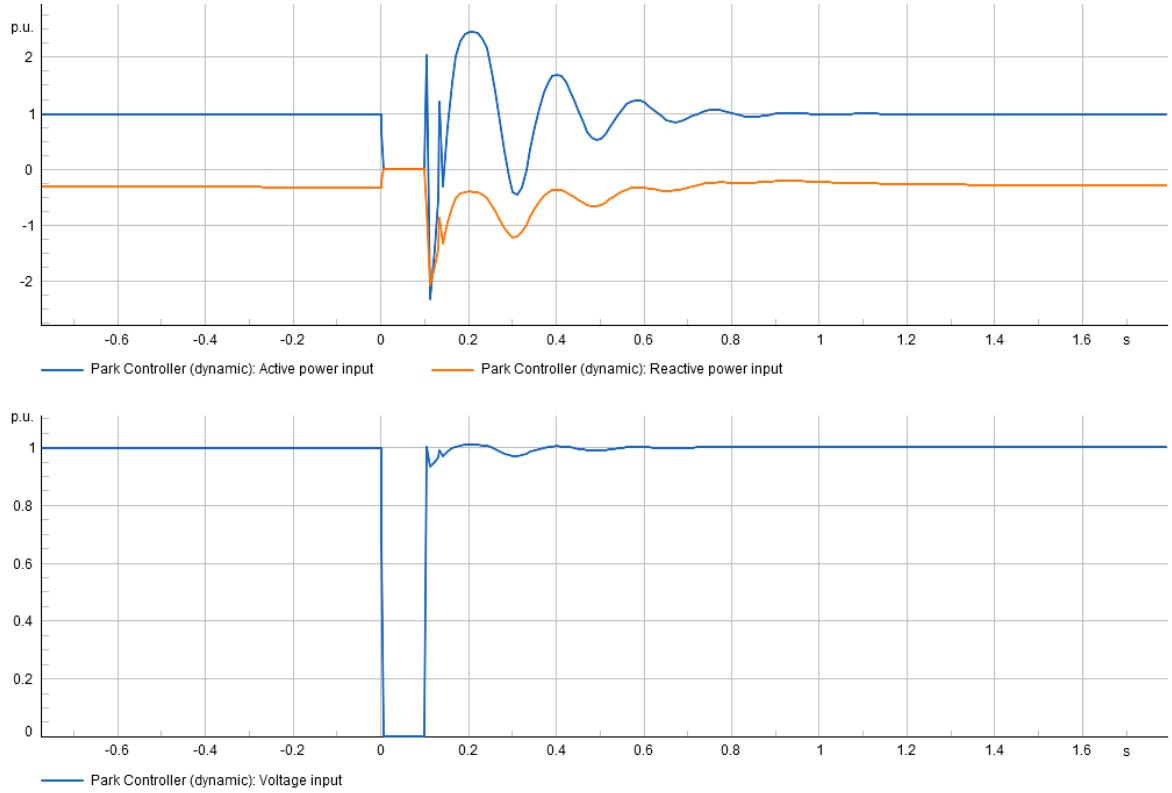


**FIGURE 2.30 VOLTAGE, ACTIVE AND REACTIVE POWER RESPONSE OF POC DURING A SHORT-CIRCUIT FAULT**

The power plant controller performance is also validated and shown in Figure 2.31. The active power reference is decreased to 0.9 p.u. from 1 p.u. at 20 s. The power plant controller monitors and controls the active and reactive power injection of the generating units at the point of connection (PoC) during dynamic simulation.

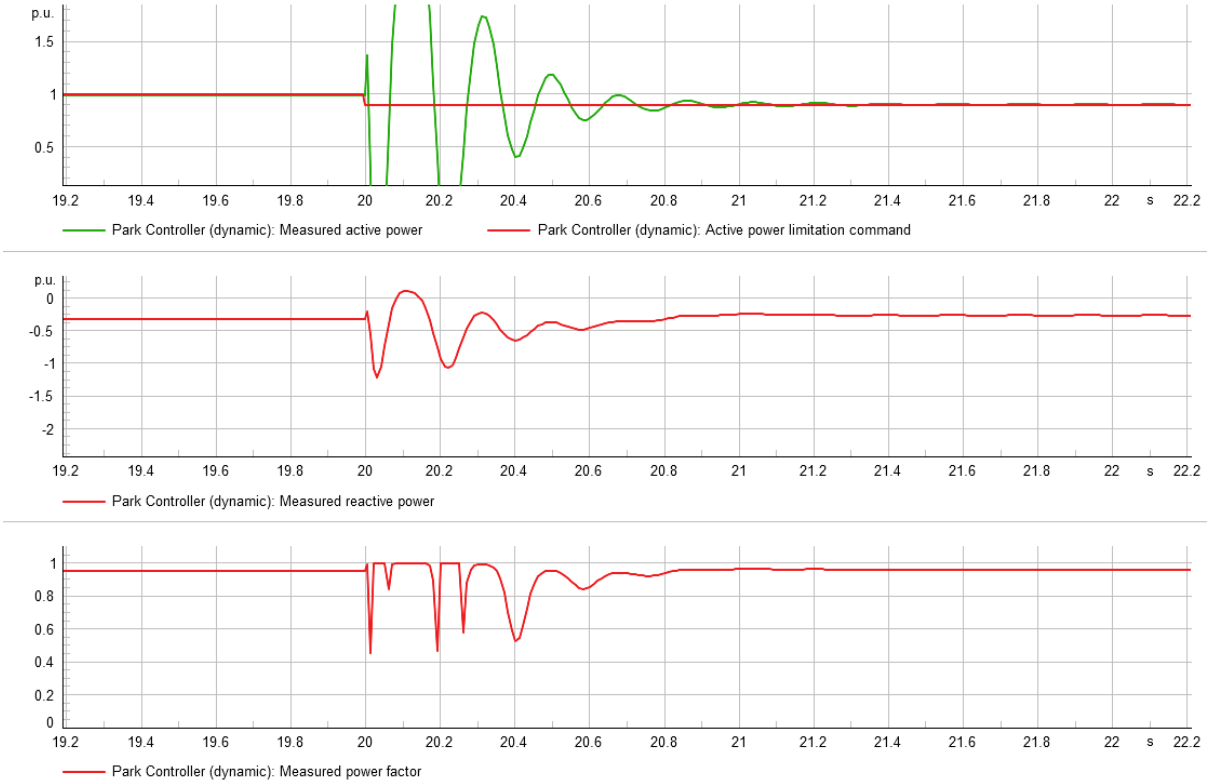


**FIGURE 2.31 ACTIVE AND REACTIVE POWER RESPONSE DURING A STEP CHANGE OF ACTIVE POWER REFERENCE**

# 2.3 Conclusion

This section provides two modelling frameworks in DIgSILENT PowerFactory and highlights the key building blocks necessary for use case studies, considering technological maturity and advancements in related work packages. DSL and Modelica-FMU provide powerful tools for modelling and simulation, yet each has its distinct strengths and challenges. DSL offers a comprehensive framework for dynamic model representation, supporting both linear and non-linear systems in continuous time domains. Its built-in macros and intrinsic functions streamline the development of controllers and models with flexibility, while formal verification procedures such as algebraic loop detection and identification of undefined variables ensure model integrity before simulation. However, its complexity and the requirement for deep familiarity with its syntax may pose challenges for users, particularly when dealing with intricate dynamic models or advanced control algorithms. The language's interface complexity can also hinder rapid prototyping and frequent model updates. On the other hand, Modelica-FMU enhances simulation tool capabilities through seamless integration with DIgSILENT PowerFactory. It supports the export of models from other platforms, such as Matlab Simulink, to FMUs, facilitating interoperability and model reuse across diverse simulation environments. The use of FMUs also ensures IP protection by encapsulating model functionality within black-box structures. Despite these benefits, the Modelica-FMU approach introduces runtime complexities, such as platform-specific dependencies and potential issues with the export of certain components, like

acausal Simscape connectors from MATLAB. Additionally, debugging FMUs is more challenging compared to DSL-based models, as their black-box nature limits access to underlying model details.

Both DSL and Modelica-FMU are used to build the following models:

- MMC in Grid-Following (GFL) Mode:
  - An MMC converter operating in standard GFL mode regulates the active power and voltage at the Point of Common Coupling (PCC), which is modelled using DSL.
  - Validation: The GFL MMC model was validated through step change tests in both active power reference and voltage reference.
- MMC in Droop Grid-Forming (GFM) Mode:
  - The MMC converter in standard droop-GFM mode was modelled using Modelica-FMU, offering enhanced control over voltage regulation.
  - Validation: The effectiveness of this mode was verified through a step change test in voltage reference and a three-phase short circuit fault test.
- MMC as virtual synchronous machine (VSM):
  - Another MMC was developed in GFM VSM mode using DSL, based on existing models from the DIgSILENT PowerFactory library.
  - Validation: The model was validated by assessing its response to a change in active power reference and the disconnection of one of the AC lines.
- Energy Dissipating System in HVDC:
  - An energy dissipating system for an HVDC system was built through Modelica-FMU to manage surplus energy during AC-side faults.
  - Validation: Effectiveness was confirmed using a voltage variation profile, demonstrating that the system prevents the continuous rise of the DC bus voltage during faults, even when the sending terminal's power remains unchanged.
- Energy storage system integration in HVDC:
  - An ESS, directly integrated into the HVDC system was built through Modelica-FMU.
  - Validation: The results from DIgSILENT PowerFactory were compared with MATLAB results, confirming successful integration.
- Wind farm with Grid-Forming Inverters:
  - A WF containing two wind turbines in GFM mode was modelled in DSL to establish offshore grid voltage and frequency.
  - Integration: The GFM (VSM) control was extended to combine with existing WF models from the DIgSILENT PowerFactory library, incorporating park control, turbine control, and protection.
  - Validation: The performance of the Under-Voltage-Ride-Through and power plant controller was validated through short circuit fault and active power reference step change tests, respectively.

These models have been validated under different test conditions, confirming the effectiveness of the grid-following, grid-forming, and energy dissipation control strategies and ESS for HVDC systems, as well as WF in GFM. Each model's capabilities, including fault resilience, active power regulation, and system integration, were thoroughly assessed, ensuring robust performance for future deployment in more realistic use cases.

# 3. Standardized Model Exchange of HVDC Equipment for RMS and EMT Simulations

## 3.1 Introduction

Power system development and operational planning are complex processes involving multiple organizations, with requirements defined by European regulations such as EU Regulation 2017/1485 (guidelines for electricity transmission system operation) and EU Regulation 2022/869 (trans-European energy infrastructure). These processes rely on exchanging detailed models of both AC and DC components of the European power system, including offshore interconnections and multi-terminal HVDC systems. As illustrated in Figure 3.1 , this model exchange integrates various simulation tools used by Transmission System Operators (TSOs) for static analyses, root mean square (RMS), electromagnetic transient (EMT). Given the involvement of over 50 entities, interoperable models are essential.

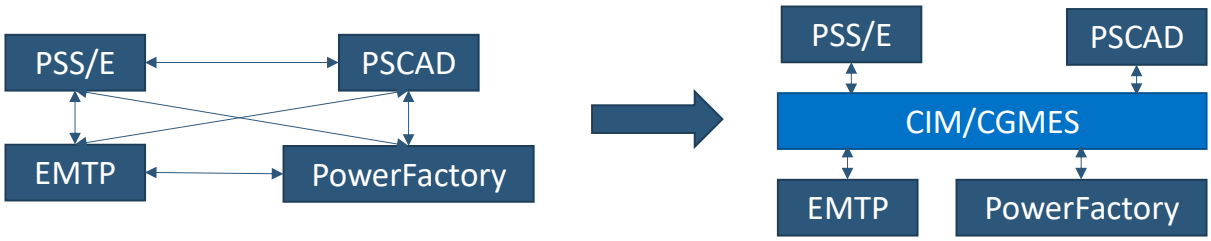


FIGURE 3.1 ILLUSTRATION OF THE NEED FOR STANDARDIZED MODEL EXCHANGE

To address interoperability challenge and model exchange challenges, a common information model (CIM) is indispensable. The IEC published a set of standards that defined the Common Information Model (CIM) and the Common Grid Model Exchange Standard (CGMES, IEC 61970-600-1:2021, IEC 61970-600-2:2021). This family of standards together with IEC CIM for Dynamics standard (IEC 61970-457:2024) provides a solid basis for meeting requirements of power system model data exchange. These standards have been used quite extensively by TSOs and some Distribution System Operators (DSOs) to exchange AC power system model data in system development and operational planning.

However, as discussed in [4], model exchange in modern hybrid AC/DC power systems presents new requirements, including:

1. Supporting exchange of both open-source and compiled (black-box) models for equipment and controls,
2. Enabling HVDC equipment model exchange, including complex configurations and details necessary for modelling of equipment for static (power flow) calculation,
3. Facilitating cross-domain model exchange, e.g., a model from RMS tool to be exchanged with EMT tool.

Along with these requirements, the following use cases are identified as key to enable seamless interactions between entities in the exchange and to be able to cope with confidentiality and intellectual property (IP) rights to which model exchanges need to adhere to:

- An exchange of black box, where the details are part of a compiled model,
- An exchange of open-source model where equations describing the model are defined by code (e.g., in Modelica, Matlab languages or regular mathematical expressions) and the code is referenced,
- An exchange of open-source model where equations are defined and are explicitly exchanged as part of the data exchange.

The new edition of IEC 61970-457 standard addresses the first requirement by supporting exchanges of both black-box and open-source models through a Detailed Model Configuration Profile. To meet the second requirement mentioned above, there is a need to build up the knowledge about which DC equipment models are crucial for model exchange, how they can be modelled with CIM. The HVDC-WISE project contributes to this by developing a public HVDC model library, HVDC-Wise Lib[2] based on IEC CIM standards. This library offers two main benefits:

- Guidance: It provides instructions for populating the library, converting models into the required format, and maintaining it as an open-source resource.
- Standardization: It supports standardizing hybrid AC/DC equipment model exchanges to ensure compatibility across simulation tools.

To successfully standardize control dynamic models for devices, it is crucial to take a comprehensive approach that ensures these models accurately reflect real-world performance. However, when dealing with studies looking at assessing new designs and integrating new technologies, the real-world performance is unknown. Invaluable insights into actual device performance and expected behaviour of models and design concepts can be gained by consulting TSOs and vendors from the outset. Such collaborative effort would help create standardized models that not only adhere to theoretical specifications but also encompass all relevant existing or expected behaviours and capabilities with high accuracy thus avoiding creating overly hypothetical models that may not align with practical implementations. HVDC-Wise Lib helps the community by exposing available models described in a standardised data exchange form which enables TSOs and other entities to perform simulation in their planning studies along with any harmonisation efforts and standardisation activities to fine tune the details of the models. This should be seen as a stepwise approach where available models are used and maintained by the community.

## 3.2 Workflow to build CIM/CGMES models

The following approach has been initially chosen for describing models in HVDC-Wise Lib:

- Static parts: The electrical circuits of HVDC equipment are modelled by extending the canonical CIM and its related data exchange profiles. This involves adding new classes, relationships, and attributes to represent these components.
- Dynamic parts: The control logic of HVDC equipment is modelled using Modelica code within the Detailed Model Configuration profile. In this method, the dynamic model equations are treated as instance data, avoiding the need for extensions to the IEC CIM standards.

The data flow diagram in Figure 3.2 illustrates the workflow to bring a model to HVDC-Wise Lib. The workflow slightly alters from initially selected approach and includes learnings from the work on bringing pilot models to the library. The workflow is discussed in detail in the following.

[2] https://github.com/HVDC-WISE/HVDC-Wise_lib

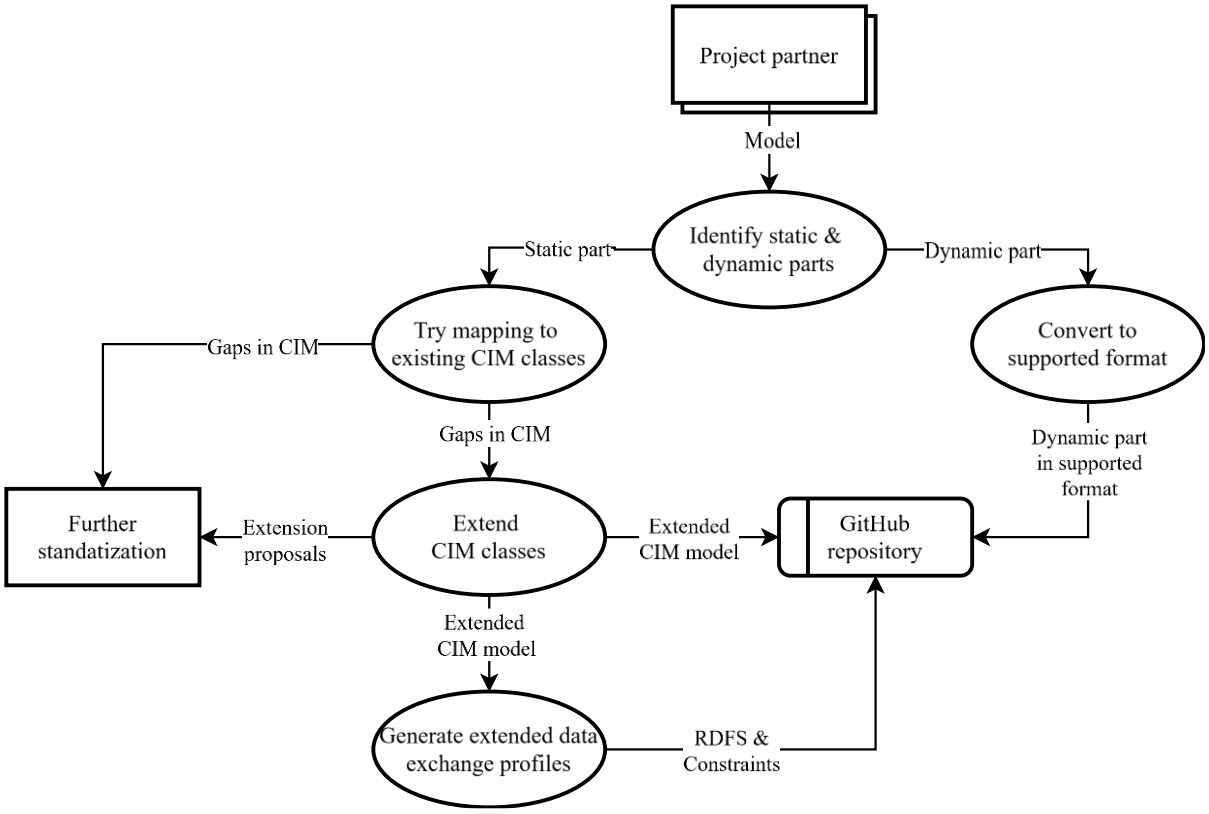


FIGURE 3.2 WORKFLOW TO BRING A MODEL TO HVDC-WISE LIB

### 3.2.1 Identification of Static and Dynamic Parts

The workflow begins with a project partner providing a model for identification of static and dynamic parts, including its full documentation and a demonstration of its implementation in the software. Splitting the model into static and dynamic parts is a practical approach for model exchange and may not correspond directly to how the model operates during simulation. The RMS and EMT simulations handle components of electrical grid differently. RMS uses complex algebraic equations and EMT differential equations to model electrical grids (cables, overhead lines, transformers, and their connectivity). For that reason, in EMT simulations, the distinction between static and dynamic components may be less apparent, but this should not be confused with how models are prepared for exchange. The key criterion for separation of the model into static and dynamic parts is what needs to be exchanged. For static part of the model, the data are exchanged, e.g., model topology and equipment parameters. For dynamic part of model, the actual model equations and optionally even the solver are exchanged.

### 3.2.2 Static part

After splitting a model into static and dynamic parts, the next step for the static part is to try mapping it to existing classes, attributes, and relationships from the latest released version of IEC CIM standards. However, since the IEC CIM standards have historically focused on AC grid data exchange, mapping the DC equipment is not always possible. In this case, gaps in the IEC CIM for representing DC equipment are identified and provided as an input for CIM extension in the HVDC-WISE project and for further recommendations for IEC CIM standardization.

The extension of IEC CIM involves adding new classes, attributes, and relationships that are not present in the standard but are necessary to describe the static part. These changes are added to the

canonical data model describing the IEC CIM and later profiled for the purpose of a structured data exchange based on RDF framework [5] (a Resource Description Framework standardised by W3C) that requires generation of an RDF Schema file to describe the data exchange profile. A custom "HVDCwise" namespace is used to differentiate the extended elements from the elements in canonical IEC CIM. The extended CIM data model is published to a GitHub repository[3], hosting HVDC-Wise Lib.

The result of this effort is an extension of existing data exchange standards (e.g. CGMES or Dynamics profile) to include new equipment. Over time, this work contributes to a new release of the IEC CIM standards, incorporating new classes, attributes, and relationships into the canonical data model.

### 3.2.3 Dynamic part

The identified dynamic part of the equipment model needs to be converted to a serialisation format, supported by IEC CIM standards for dynamic model exchange. As mentioned earlier, the initially defined approach was to exchange dynamic part using Modelica equations in accordance with Detailed Model Configuration profile from the new edition of IEC 61970-457 standard. However, control equations provided by project partners are not always available in Modelica and converting them to Modelica is a non-automated process, that depending on the size of the model can become time-consuming and error prone. To address this challenge, HVDC-WISE decided to utilize several different formats supported by IEC 61970-457 for exchanging dynamic model equations:

- Modelica equations in accordance with Detailed Model Configuration profile,
- Matlab equations in accordance with Detailed Model Configuration profile,
- Compiled models in accordance with FMI standard [6]. In this case, compiled model is exchanged as an artifact.

The dynamic part is converted to one of these formats and stored in a GitHub repository, hosting HVDC-WISE Lib.

## 3.3 Application of the workflow variants for dynamic part

This section describes the application of the workflow to pilot models and reports some lessons learned.

### 3.3.1 Initial set of approaches for dynamic part

Table 3.1 presents models that are considered as part of the initial release of HVDC-Wise Lib along with short model description, project partner and approach envisioned for exchange of dynamic part of the model. In principle, there are two main approaches for the dynamic part: exchange of dynamic part as an artifact (FMU) or CIM XML serialization of dynamic equations. The dynamic equations can be either described by Modelica or Matlab code. In the following, the report discusses in more detail the two approaches: exchange of FMU and CIM XML serialization of Modelica equations, using examples from HVDC-Wise Lib.

[3] https://github.com/HVDC-WISE/HVDC-Wise_lib

TABLE 3.1 CONSIDERED MODELS

| NAME | DESCRIPTION | PROJECT PARTNER | DYNAMIC PART |
|---|---|---|---|
| HVDC MMC | Model includes control modules: synthetic inertia, frequency sensitive mode, power oscillation damping, P/Vdc control, Q/Vac control, voltage unbalance control, fault ride through. | TenneT | CIM XML serialization of Modelica equations |
| DC energy storage | Model of a chain link of energy storage submodules, see [3] | Supergrid Institute | Artifact (FMU) |
| Voltage source converter | Generic voltage source converter HVDC model for power system stability studies, described in [7] | RWTH | CIM XML serialization of Matlab equations |
| Grid-following inverter | Average value inverter model with grid-following control, described in [8]. | RWTH | CIM XML serialization of Modelica equations |

### 3.3.2 FMU

An energy storage system connected to HVDC terminals has been selected as an example of potential future equipment to be included in standardized model exchanges. This kind of equipment is currently at low technology readiness level (TRL), but there is an interest to disseminate models to allow system planners to assess benefits of such technologies. The considered converter is presented in section 2.2.3.

For the model implementation, it has been decided to include in the static part only the elements connected to the DC terminals in Figure 2.20 i.e., the equivalent resistance and inductance of the inductor, and the controlled voltage source. The dynamic part defines $v_b$ and includes the modelling of the energy storage elements and the energy stored in submodule capacitors.

Dynamic part of the model includes the control of the converter. As a control has been previously developed in Matlab Simulink (for EMT simulations), and as it would be time consuming to re-describe the scheme with Modelica equations, it has been decided to test the possibility of representing the dynamic part by referencing a black box (compiled) model according to the FMI standard [6]. This way the dynamic equations do not need to be described explicitly in CIM XML. The FMU file used is the same the one created for DIgSILENT PowerFactory in section 2.2.3.

The connectivity of the artifact to the static part of the model (*Equipment* class in CIM) can be described using CIM Detailed Model Configuration Profile, introduced in IEC 61970-457:2024.

### 3.3.3 CIM XML serialisation of Modelica equations

A MMC HVDC model is used to demonstrate the exchange of user defined model that originated in DIgSILENT PowerFactory. Initial analysis of the model is performed, and the equations are extracted and represented by Modelica code. This Modelica code that describes the behaviour of the different functional blocks part of the model needs to be serialised using the specification for an exchange of detailed dynamics models described in IEC 61970-457:2024.

There are different ways how to serialise the information in a form compatible with IEC 61970-457:

- Using DIgSILENT PowerFactory export function (under develoment by DIgSILENT),

- Using Python API by DIgSILENT – this requires development of scripts to expand CIM Archive in DIgSILENT PowerFactory and from there export the user defined model,
- Using external tools.

For the purpose of HVDC-WISE prototype of user defined models and creation of the open-source library, the 3rd way of translation was utilized. Therefore, after the extraction of the equations from DIgSILENT PowerFactory user defined model, an Excel template was filled in order to prepare for the serialisation described in IEC 61970-457. This standard is currently using CIM XML type of serialisation which is defined in IEC 61970-552:2016. In the future, different RDF based serialisations (e.g. JSON-LD) can be utilised as well.

The Detailed Model Configuration (DMC)-based approach included modelling control frames as different DMC profile instances, e.g. DMC_POD.xml as a model of the power oscillation control. Moreover, all DMC instances having the same *DetailedModelTypeDynamics* were assumed to belong to the same RMS model. For example, the controls of the HVDC MMC model contained the same instance of *DetailedModelTypeDynamics* class, as can be seen in Figure 3.3.

DMC_POD.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:cim="http://cim.ucaiug.io/ns#"
    xmlns:eumd="http://entsoe.eu/ns/Metadata-European#"
    xmlns:prov="http://www.w3.org/ns/prov#"
    xmlns:md="http://iec.ch/TC57/61970-552/ModelDescription/1#"
    xmlns:skos="http://www.w3.org/2004/02/skos/core#"
    xmlns:dcat="http://www.w3.org/ns/dcat#"
    xmlns:dcterms="http://purl.org/dc/terms/#"
  xml:base="http://iec.ch/TC57/CIM100" >
  <cim:DetailedModelTypeDynamics rdf:ID="_e7eb2061-25b7-4840-abfd-be2cde9c1d2c">
    <cim:IdentifiedObject.description>HVDC MMC control frame for balanced and unbalanced RMS and EMT simulation</cim:IdentifiedObject.description>
    <cim:IdentifiedObject.mRID>e7eb2061-25b7-4840-abfd-be2cde9c1d2c</cim:IdentifiedObject.mRID>
    <cim:IdentifiedObject.name>HVDC MMC Control (TenneT)</cim:IdentifiedObject.name>
  </cim:DetailedModelTypeDynamics>
```

FIGURE 3.3 A DMC INSTANCE

The approach also consisted of modelling blocks as instances of *FunctionDescriptor* or *OperatorDescriptor* class, depending on the block type. It is worth noting that the operators and limiters were modelled as *FunctionDescriptors* if the corresponding dynamic equations were explicitly defined in the original models. Furthermore, model parameters were modelled by utilizing *ParameterDescriptor* class, the interconnections between the control blocks as *InputOutputDescriptor* class instances, and input and output signals were represented by instances of *SignalDescriptor* class.

In certain cases, subframes (such as Measurement and Protection frames inside the main frame of the HVDC MMC RMS model) were modelled in the same way as the blocks (i.e., as instances of *FunctionDescriptor*) to reduce complexity and enable nesting. However, this has led to challenges due to increased complexity, the uncertainty of ID persistency and the concept of a *Model in a Slot in a Frame in a Frame*. As a solution, recommendations on improving the modelling of the frame concept were issued towards the standardization bodies.

The approach of modelling a control block may be illustrated in the following section for a POD control model of HVDC MMC given in Figure 3.4.

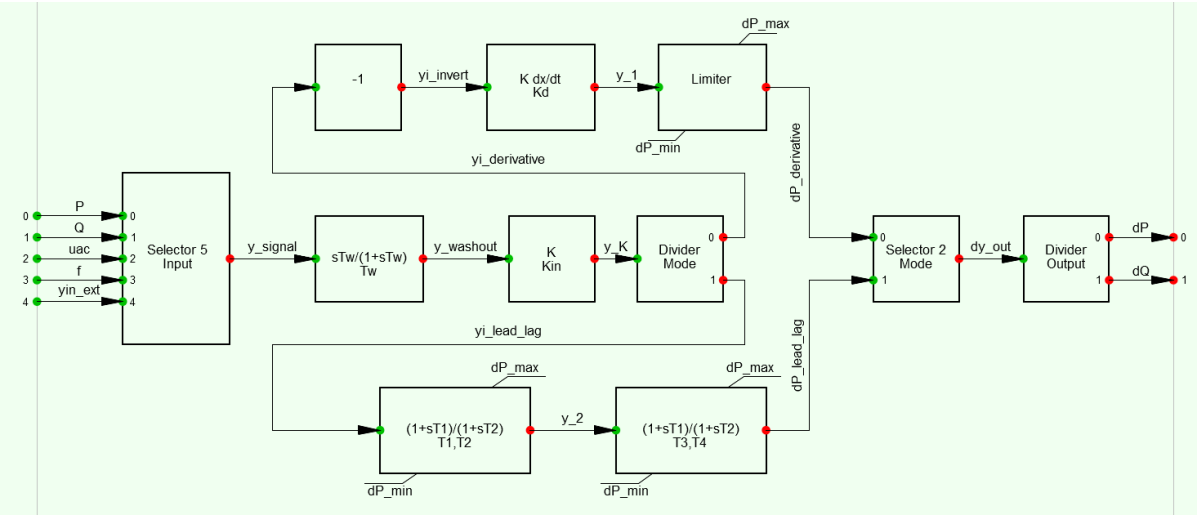


FIGURE 3.4 POD CONTROL MODEL OF HVDC MMC

The first step is to define the blocks inside the frame and assign them the mandatory attributes of *FunctionDescriptor* class, as shown in Table 3.2 for two of the blocks.

TABLE 3.2 FUNCTIONDESCRIPTOR SAMPLES

| FUNCTIONDESCRIPTOR.NAME | POD SELECTOR 5 | POD WASHOUT FILTER |
|---|---|---|
| FunctionDescriptor ID | _638fa29e-ce7a-495c-a52c-86de585b69e0 | _5bfb424d-33ef-4019-a456-e65186b47e5b |
| FunctionDescriptor.Description | Selector with 5 input signals | Washout filter |
| InputOutputDescriptor | InputOutputDescriptor for POD Selector 5 block | InputOutputDescriptor for POD Washout filter block |
| InputOutputDescriptor ID | _af0814f3-a203-4c2b-b23a-27b2ee4534ef | _a56a6607-285a-408a-b652-59b985b9b13c |

The next step is converting and mapping the input and output signals to all blocks, as shown in Table 3.3, and connecting them to the respective *InputOutputDescriptors*.

TABLE 3.3 SIGNALDESCRIPTOR SAMPLES

| SIGNAL TO THE FRAME / SLOT | P | Q | Y_SIGNAL |
|---|---|---|---|
| SIGNAL IN THE TYPE | yi1 | yi2 | yo |
| SIGNAL.ID | _62dda81c-5feb-4186-a024-487f1e8a4c73 | _ac10b613-4d36-40c0-9ced-a257e3482311 | _b4983f6c-070d-47c2-8d4b-f9cd4cba597f |
| SIGNALDESCRIPTOR.DESCRIPTION | Input signal for POD Selector 5 block | Input signal for POD Selector 5 block | Output signal for POD Selector 5 block |
| SIGNALDESCRIPTOR.ISSIGNALINVERTED | FALSE | FALSE | FALSE |

Connection between blocks is made by assigning a signal to different instances of *InputOutputDescriptor* class, e.g., since the signal *y_signal* is at the same time an output of POD Selector 5 (the left-most block in Figure 3.4) and an input of POD Washout filter (the block to the right of POD Selector 5 in Figure 3.4), it should be associated to the *InputOutputDescriptor* of POD Selector 5 as *OutputSignal*, and to the of *InputOutputDescriptor* the POD Washout filter as *InputSignal*.

Afterwards, for each *FunctionDescriptor*, parameters are defined as attributes of the CIM class called *ParameterDescriptor*, as shown in Table 3.4.

TABLE 3.4 PARAMETERDESCRIPTOR SAMPLES

| FUNCTIONDESCRIPTOR.NAME | PARAMETERDESCRIPTOR.NAME | PARAMETERDESCRIPTOR.ID | PARAMETERDESCRIPTOR.KIND |
|---|---|---|---|
| POD SELECTOR 5 | Input | _89d48d37-7a2b-4713-b246-872fc7aa3af0 | regular |
| POD WASHOUT FILTER | xw | _d53f468c-226a-4f19-9cb5-e500565e325a | stateVariable |
| POD WASHOUT FILTER | Tw | _e3fb1fea-92ff-4306-ae29-2f4f455da476 | regular |

Finally, the corresponding equations are expressed using Modelica language and at least all required attributes and association are populated in the Excel template. If the native language of the model is not Modelica, equations (not describing connectivity) must be converted accordingly.

Finally, an open-source application CimPal [9] is used to import the template and serialise the data. Figure 3.5 illustrates the function *Generation of instance data based in xls template* in CimPal.

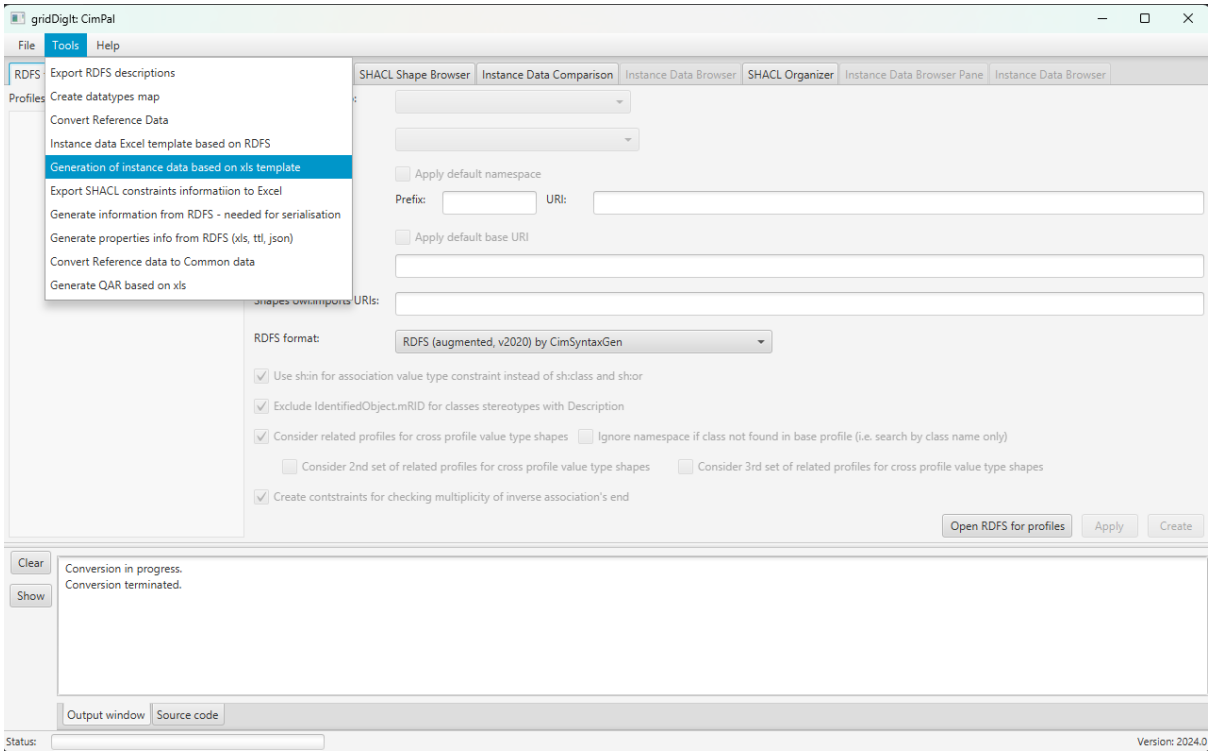


FIGURE 3.5 GENERATION OF INSTANCE DATA BASED IN XLS TEMPLATE IN CIMPAL

As a result, a set of CIM XML files are being generated. A snippet of the CIM XML file describing the data given in Figure 3.4. is given in Figure 3.6.

```
  <cim:InputOutputDescriptor rdf:ID="_af0814f3-a203-4c2b-b23a-27b2ee4534ef">
    <cim:DetailedModelDescriptor.DetailedModelTypeDynamics rdf:resource="#_e7eb2061-25b7-4840-abfd-be2cde9c1d2c"/>
    <cim:IdentifiedObject.description>InputOutputDescriptor for POD Selector 5 block</cim:IdentifiedObject.description>
    <cim:IdentifiedObject.mRID>af0814f3-a203-4c2b-b23a-27b2ee4534ef</cim:IdentifiedObject.mRID>
    <cim:InputOutputDescriptor.InputSignal rdf:resource="#_62dda81c-5feb-4186-a024-487f1e8a4c73"/>
    <cim:InputOutputDescriptor.InputSignal rdf:resource="#_409bc51e-2250-4d48-9fb2-c42824a75dee"/>
    <cim:InputOutputDescriptor.InputSignal rdf:resource="#_b71ea0fc-8cb2-40bd-8f4d-6c2a29f05d2f"/>
    <cim:InputOutputDescriptor.InputSignal rdf:resource="#_ac10b613-4d36-40c0-9ced-a257e3482311"/>
    <cim:InputOutputDescriptor.InputSignal rdf:resource="#_d634b11e-ff51-4a3e-aff5-6bf0d71edd19"/>
    <cim:InputOutputDescriptor.OutputSignal rdf:resource="#_b4983f6c-070d-47c2-8d4b-f9cd4cba597f"/>
  </cim:InputOutputDescriptor>
  <cim:InputOutputDescriptor rdf:ID="_a56a6607-285a-408a-b652-59b985b9b13c">
    <cim:DetailedModelDescriptor.DetailedModelTypeDynamics rdf:resource="#_e7eb2061-25b7-4840-abfd-be2cde9c1d2c"/>
    <cim:IdentifiedObject.description>InputOutputDescriptor for POD Washout filter block</cim:IdentifiedObject.description>
    <cim:IdentifiedObject.mRID>a56a6607-285a-408a-b652-59b985b9b13c</cim:IdentifiedObject.mRID>
    <cim:InputOutputDescriptor.InputSignal rdf:resource="#_b4983f6c-070d-47c2-8d4b-f9cd4cba597f"/>
    <cim:InputOutputDescriptor.OutputSignal rdf:resource="#_a9961755-a1db-4ab4-bc08-8e08b376fbd6"/>
  </cim:InputOutputDescriptor>
  <cim:FunctionDescriptor rdf:ID="_5bfb424d-33ef-4019-a456-e65186b47e5b">
    <cim:DetailedModelDescriptor.DetailedModelTypeDynamics rdf:resource="#_e7eb2061-25b7-4840-abfd-be2cde9c1d2c"/>
    <cim:FunctionDescriptor.InputOutputDescriptor rdf:resource="#_a56a6607-285a-408a-b652-59b985b9b13c"/>
    <cim:FunctionDescriptor.describedByArtifact>false</cim:FunctionDescriptor.describedByArtifact>
    <cim:FunctionDescriptor.equation>model POD_Washout_filter
equation
    dx=Select((Tw&amp;gt;0),((yi-x)/Tw),0);
    der(x)=dx;
    yo=Select((Tw&amp;gt;0),(Tw*dx),yi);
end POD_Washout_filter;

function Select
    input Boolean boolexpr;
    input Real x;
    input Real y;
    output Real varA;
algorithm
    if boolexpr then
        varA = x;
    else
        varA = y;
    end if;
end Select;</cim:FunctionDescriptor.equation>
    <cim:FunctionDescriptor.equationLanguageKind rdf:resource="#http://cim.ucaiug.io/ns#EquationLanguageKind.modelica"/>
    <cim:IdentifiedObject.description>Washout filter</cim:IdentifiedObject.description>
    <cim:IdentifiedObject.mRID>5bfb424d-33ef-4019-a456-e65186b47e5b</cim:IdentifiedObject.mRID>
    <cim:IdentifiedObject.name>POD Washout filter</cim:IdentifiedObject.name>
  </cim:FunctionDescriptor>
  <cim:FunctionDescriptor rdf:ID="_638fa29e-ce7a-495c-a52c-86de585b69e0">
    <cim:DetailedModelDescriptor.DetailedModelTypeDynamics rdf:resource="#_e7eb2061-25b7-4840-abfd-be2cde9c1d2c"/>
    <cim:FunctionDescriptor.InputOutputDescriptor rdf:resource="#_af0814f3-a203-4c2b-b23a-27b2ee4534ef"/>
    <cim:FunctionDescriptor.describedByArtifact>false</cim:FunctionDescriptor.describedByArtifact>
    <cim:FunctionDescriptor.equation>model POD_Selector_5
    <cim:FunctionDescriptor.equationLanguageKind rdf:resource="#http://cim.ucaiug.io/ns#EquationLanguageKind.modelica"/>
    <cim:IdentifiedObject.description>Selector with 5 input signals</cim:IdentifiedObject.description>
    <cim:IdentifiedObject.mRID>638fa29e-ce7a-495c-a52c-86de585b69e0</cim:IdentifiedObject.mRID>
    <cim:IdentifiedObject.name>POD Selector 5</cim:IdentifiedObject.name>
  </cim:FunctionDescriptor>
  <cim:ParameterDescriptor rdf:ID="_89d48d37-7a2b-4713-b246-872fc7aa3af0">
    <cim:DetailedModelDescriptor.DetailedModelTypeDynamics rdf:resource="#_e7eb2061-25b7-4840-abfd-be2cde9c1d2c"/>
    <cim:IdentifiedObject.description>POD input: 1=P, 2=Q, 3=Vac, 4=f, 5= external signal, 0=none</cim:IdentifiedObject.description>
    <cim:IdentifiedObject.mRID>89d48d37-7a2b-4713-b246-872fc7aa3af0</cim:IdentifiedObject.mRID>
    <cim:IdentifiedObject.name>Input</cim:IdentifiedObject.name>
    <cim:ParameterDescriptor.FunctionDescriptor rdf:resource="#_638fa29e-ce7a-495c-a52c-86de585b69e0"/>
    <cim:ParameterDescriptor.FunctionDescriptor rdf:resource="#_c77bb257-62be-4bf1-b61a-8402382ce784"/>
    <cim:ParameterDescriptor.kind rdf:resource="#http://cim.ucaiug.io/ns#ParameterKind.regular"/>
    <cim:ParameterDescriptor.sequenceNumber>5.0</cim:ParameterDescriptor.sequenceNumber>
    <cim:ParameterDescriptor.valueXSDdatatype rdf:resource="#http://cim.ucaiug.io/ns#XSDDatatypeKind.integer"/>
  </cim:ParameterDescriptor>
  <cim:ParameterDescriptor rdf:ID="_d53f468c-226a-4f19-9cb5-e500565e325a">
    <cim:DetailedModelDescriptor.DetailedModelTypeDynamics rdf:resource="#_e7eb2061-25b7-4840-abfd-be2cde9c1d2c"/>
    <cim:IdentifiedObject.description>stateVariable</cim:IdentifiedObject.description>
    <cim:IdentifiedObject.mRID>d53f468c-226a-4f19-9cb5-e500565e325a</cim:IdentifiedObject.mRID>
    <cim:IdentifiedObject.name>xw</cim:IdentifiedObject.name>
    <cim:ParameterDescriptor.FunctionDescriptor rdf:resource="#_5bfb424d-33ef-4019-a456-e65186b47e5b"/>
    <cim:ParameterDescriptor.FunctionDescriptor rdf:resource="#_c77bb257-62be-4bf1-b61a-8402382ce784"/>
    <cim:ParameterDescriptor.kind rdf:resource="#http://cim.ucaiug.io/ns#ParameterKind.stateVariable"/>
    <cim:ParameterDescriptor.sequenceNumber>1.0</cim:ParameterDescriptor.sequenceNumber>
    <cim:ParameterDescriptor.valueXSDdatatype rdf:resource="#http://cim.ucaiug.io/ns#XSDDatatypeKind.float"/>
  </cim:ParameterDescriptor>
  <cim:SignalDescriptor rdf:ID="_ac10b613-4d36-40c0-9ced-a257e3482311">
    <cim:DetailedModelDescriptor.DetailedModelTypeDynamics rdf:resource="#_e7eb2061-25b7-4840-abfd-be2cde9c1d2c"/>
    <cim:IdentifiedObject.description>Input signal for POD Selector 5 block</cim:IdentifiedObject.description>
    <cim:IdentifiedObject.mRID>ac10b613-4d36-40c0-9ced-a257e3482311</cim:IdentifiedObject.mRID>
    <cim:IdentifiedObject.name>Q</cim:IdentifiedObject.name>
    <cim:SignalDescriptor.isSignalInverted>False</cim:SignalDescriptor.isSignalInverted>
  </cim:SignalDescriptor>
  <cim:SignalDescriptor rdf:ID="_62dda81c-5feb-4186-a024-487f1e8a4c73">
    <cim:DetailedModelDescriptor.DetailedModelTypeDynamics rdf:resource="#_e7eb2061-25b7-4840-abfd-be2cde9c1d2c"/>
    <cim:IdentifiedObject.description>Input signal for POD Selector 5 block</cim:IdentifiedObject.description>
    <cim:IdentifiedObject.mRID>62dda81c-5feb-4186-a024-487f1e8a4c73</cim:IdentifiedObject.mRID>
    <cim:IdentifiedObject.name>P</cim:IdentifiedObject.name>
    <cim:SignalDescriptor.isSignalInverted>False</cim:SignalDescriptor.isSignalInverted>
  </cim:SignalDescriptor>
  <cim:SignalDescriptor rdf:ID="_b4983f6c-070d-47c2-8d4b-f9cd4cba597f">
    <cim:DetailedModelDescriptor.DetailedModelTypeDynamics rdf:resource="#_e7eb2061-25b7-4840-abfd-be2cde9c1d2c"/>
    <cim:IdentifiedObject.description>Input signal for POD Washout filter block</cim:IdentifiedObject.description>
    <cim:IdentifiedObject.mRID>b4983f6c-070d-47c2-8d4b-f9cd4cba597f</cim:IdentifiedObject.mRID>
    <cim:IdentifiedObject.name>y_signal</cim:IdentifiedObject.name>
    <cim:SignalDescriptor.isSignalInverted>False</cim:SignalDescriptor.isSignalInverted>
  </cim:SignalDescriptor>
```

**FIGURE 3.6 SAMPLE CIM XML SERIALIZATION**

These files/datasets can be used for the purpose of an exchange of user defined models. It is expected that other applications will be able to parse the information from these files and extract the equations as well as the information related to connectivity of the different function blocks, i.e. input and output signals as well as the parameters that are used by the model. As the dynamic part of the model exchanged using IEC 61970-457 contains references to the static part of the model, an application that consumes both the static and dynamic part would be able to link the dynamic model with the static model. This is a prerequisite for performing any future dynamic (RMS or EMT) simulations on the complete dataset – model.

# 3.4 Identified gaps and proposed extensions

This section highlights gaps identified in IEC CIM for exchange of static and dynamic parts of equipment models when building HVDC-Wise Lib. Furthermore, propositions to address these gaps are made. For the gaps identified for exchange of static part, details of the proposed extensions are given in Appendix A. This includes new required classes and attributes. The extensions to enhance exchange of dynamic part of equipment models will be proposed as recommendation of the project in a further deliverable (D8.3).

## 3.4.1 Static part

As CIM-based data exchange currently requires concrete modelling of the static part of equipment models, several gaps in canonical CIM are identified.

First, relevant classes and attributes to model electrical circuit of DC energy storage are not available in current editions of the IEC standards. It was then necessary to propose new classes including a class *DCStorage* which has attributes like rated DC voltage, maximal power, number of submodules, series inductance. The submodules can be of two types (with batteries or supercapacitors) which are also described in proposed extensions with attributes including maximal energy, no-load voltage, capacitance.

The second identified gap is the limited capability of the standard to report DC state variables of DC terminals and nodes, i.e. in general there is a gap to report power flow results of the DC part of the model. Therefore, this gap was also covered by an extension.

## 3.4.2 Dynamic part

The dynamic part is relying on the IEC 61970-457:2024 that was published recently. Working on HVDC-Wise Lib resulted in identifying two gaps in modelling of dynamic part. Extensions to IEC 61970-457:2024, addressing those gaps, can lead to more convenient modelling of dynamic part, especially composite dynamic models, like control systems, that may involve multiple groups of controls. HVDC-WISE is working to propose a set of CIM extensions to cover this part. The proposal will be presented in another deliverable (D8.3).

The first gap is related to modelling of relationships inside of composite dynamic models, i.e., containing several dynamic submodules. HVDC-WISE finds it useful to be able to model a composite model as a kind of a framework structure that has different frame parts that will host functional blocks of various model types of part of the frame. One essential functionality to be covered by the framework is to be able to express the connectivity between different frame parts in a form of signals.

The second gap is related to a possibility of modelling multiple *InputOutputDescriptor*s associated with the single *FunctionDescriptor*. An example use case is illustrated in Figure 3.7 and described in the

following. In Modelica implementation of grid-following inverter control system, there are two different instances of the same class *abc_to_DQ* used within the model. The two instances of *abc_to_DQ* are applying the same transformation to the signals and do not have state variables.

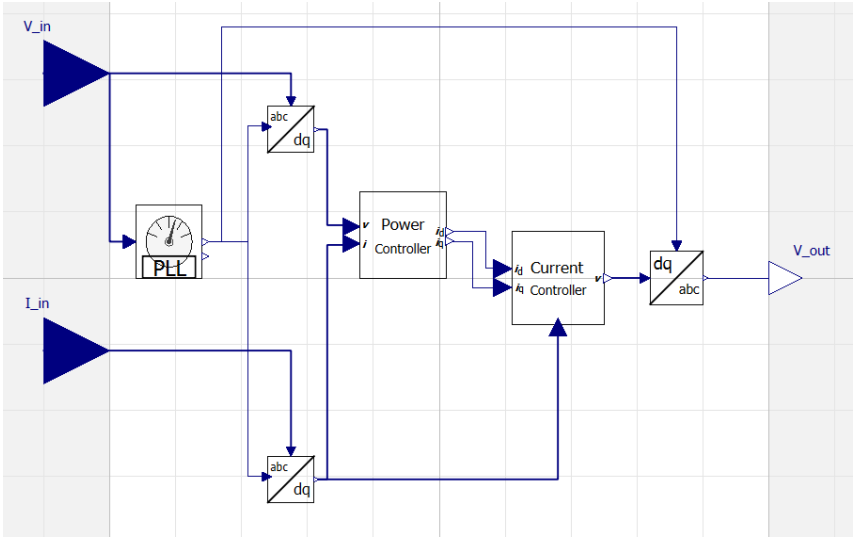


FIGURE 3.7 OPENMODELICA IMPLEMENTATION OF GRID-FOLLOWING INVERTER CONTROL SYSTEM

Currently, a separate *FunctionDescriptor* needs to be created to model each of the *abc_to_DQ* transformations, as a *FunctionDescriptor* can be associated just one *InputOutputDescriptor* (see Figure 3.8). In the example both *FunctionDescriptors* include the same equations and lead to code duplication. Allowing a *FunctionDescriptor* to have association with multiple *InputOutpuDescriptors* could help reduce the code duplication and therefore complexity of dynamic model in CIM XML. The concrete suggestion to IEC could be to change cardinality of FunctionDescriptor.InputOutDescriptor from *0..1* to *0..**.

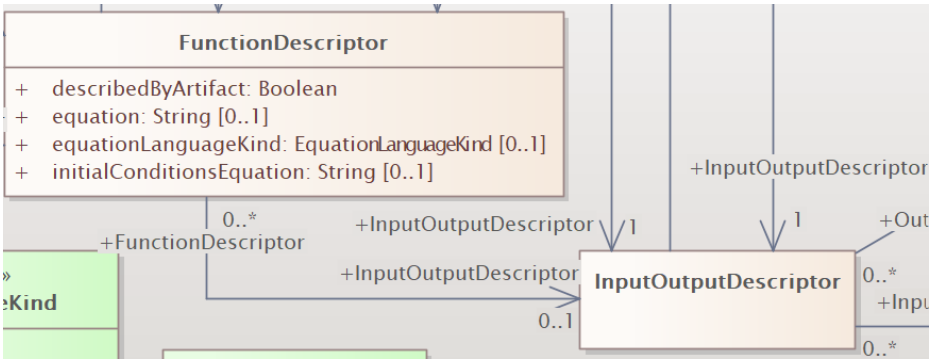


FIGURE 3.8 RELATIONSHIP BETWEEN FUNCTIONDESCRIPTOR AND INPUTOUTPUTDESCRIPTOR

# 3.5 Results and next steps

This section describes the lessons learned, main results and further steps, related to IEC CIM-based model exchange of HVDC equipment models.

## 3.5.1 Lessons learned

The following lessons learned have been recorded while working on the library of models.

- There should be proactive efforts and communication to facilitate the sharing of models developed by different organisations. The intellectual property rights (IPR) and licensing conditions could be obstacles, but in most of the cases parties are willing to share for the purpose to standardise the model exchange.
- Multiple project partners contribute to HVDC-Wise Lib and the models are originally available in different modelling and simulation tools. There is a lack of reliable tools to automatically convert a model from one format to another, i.e. from Matlab SImulink to Modelica. The resources to convert dynamic parts of the model to a particular form of equations can be time consuming, error-prone and would create a bottleneck for the project if the data exchange would support only one language for describing the models. HVDC-WISE partners welcomed the setup of the CIM for dynamics standards that have sufficient support for open-source code exchange and exchange of compiled models. Another learning is that model import/export capabilities can be an important factor for selection of modelling/simulation tool, when developing a new model. For example, it is an advantage if the tool supports compilation of the model into FMU.
- When preparing different forms of the models in the library there is a need to utilise different tooling. Therefore, experts that are proposing a model need to get familiar with the structure of the library and various possibilities for contribution. Close interactions with HVDC-WISE project partners are advisable in order to optimise resources. Some of the activities could be very time consuming if not approached in the right way.
- The approach for exchange of dynamic part of the model in HVDC-Wise Lib is based on the very recent standard IEC 61970-457:2024. Widely used commercial power system modelling/simulation tools have not implemented the support of this standard yet. Therefore, complete and automatic CIM import of the models, including the dynamic parts is not yet possible. Same goes for the static parts of equipment models, that required extensions.

### 3.5.2 Standardization

The following proposals have been made during the piloting work on HVDC-Wise Lib:

- Extensions to describe the static part of the DC storage model which includes the DC battery storage submodule and DC super capacitor storage submodule.
- A set of extensions to enable reporting on state variables for the DC part of the model i.e., to report power and voltage at the DC terminals and nodes. The polarity for the DC terminals was also added which is essential for modelling the VSC-based HVDC systems. Figure 3.9 illustrates the proposed extensions that are prepared for inclusion in the next release of IEC CIM standards.
- An ability to explicitly describe a frame of the dynamic part of the model would facilitate modelling of the relationship between different modules of a control system.
- Allowing a *FunctionDescriptor* to have association with multiple *InputOutpuDescriptors i*n order to reduce the complexity of dynamic model.

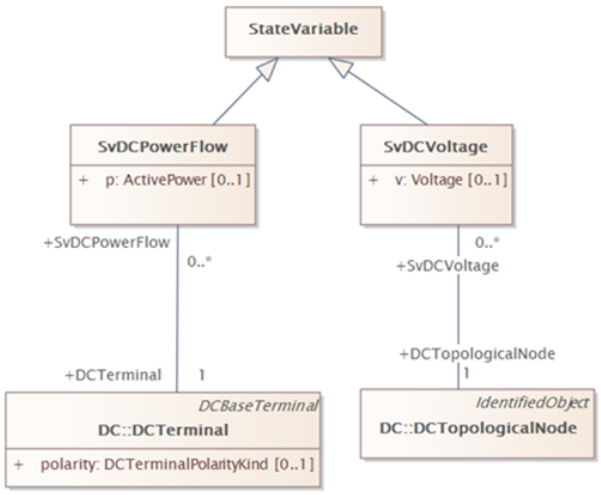


FIGURE 3.9 EXTENSIONS REGARDING DC STATE VARIABLES

Because of the project, most of these extensions were discussed in the IEC and CIM open community and adopted for inclusion in the next release of the IEC CIM standards.

## 3.5.3 HVDC-Wise Lib

The results of the work on standardized exchange of models for hybrid AC/DC grids are disseminated in HVDC-Wise Lib[4], a GitHub repository corresponding to Deliverable 4.3 of the HVDC-WSIE project. The repository hosts a library of HVDC equipment models for model exchange, based on the IEC CIM/CGMES standard format and is publicly available.

## 3.5.4 Potential synergies

As HVDC-Wise Lib is publicly available, further synergy is possible through collaboration with Cresym's collaborative open-source dynamic library "Colib" [5]. The focus of Colib is on hosting model documentation and equations, while the focus of HVDC-Wise Lib is rather on model exchange based on the IEC CIM/CGMES standard format. Both libraries can be enriched and benefit from the exchange.

## 3.5.5 Further work

The presented methodology and analysis work gives confidence that a model exchange of HVDC systems for performing complex studies is feasible. As mentioned in the section 3.5.1 of this report, there are still organizational, IP, interoperability and standardization challenges to be addressed. One of the key added values of the effort is to identify standardization gaps and propose CIM extensions to overcome the issues.

Models initially included in HVDC-Wise Lib satisfy the outlined scope of the project and are valuable inputs for standardization work. HVDC-WISE Lib being released, apart from further extending the library, it is recommended to have an interoperability effort. This effort would involve testing of model import and cross-validation of simulation results with the models imported to different power system simulation tools. Any findings should lead to updating the versions of the models included in the library. This would enable TSOs and organisations like ENTSO-E to recognize the set of models in the library and recommend the usage in the planning or operational studies.

[4] https://github.com/HVDC-WISE/HVDC-Wise_lib
[5] https://github.com/CRESYM/colib0.github.io

# 4. Inverter modelling in dynamic phasor domain

## 4.1 Dynamic phasors

Dynamic phasors, also referred to as envelope functions, emerged in the 1980s as a robust method to model and analyze electrical systems featuring oscillatory behaviors. Initially introduced as a generalized state-space averaging method in power electronics [10], this approach allowed engineers to handle switching circuits with high-frequency dynamics efficiently. The late 1990s and 2000s saw the start of integration of dynamic phasors into power system simulation platforms. Studies like [11] demonstrated the hybrid use of electromagnetic transient (EMT) and dynamic phasor techniques to simulate large interconnected systems. Theory behind application of dynamic phasors for modelling and simulation of power systems in EMTP-type programs is described in [12]. The modelling approach, that relies on dynamic phasors instead of real signals to represent power system model variables, like voltages and currents, is called by the authors of [12] a "Shifted-Frequency Analysis" (SFA). In this case, the power system model is said to be represented in shifted frequency or alternatively in dynamic phasor modeling domain. Finally, recent works like [13] focused on application of dynamic phasor modelling for real-time simulation of large power systems.

Dynamic phasors transform oscillatory time-domain signals into a complex-valued representation, encapsulating both amplitude and phase variations over time. This approach leverages concepts from signal processing, particularly bandpass signal decomposition and baseband representation. To give a definition of dynamic phasor, let's refer to Figure 4.1. A bandpass signal $u(t)$ (amplitude modulated signal in Figure 4.1) can be split into bandpass signal and carrier signal, and represented as following:

$$\boldsymbol{u(t) = u_I(t)cos(\omega t) + ju_Q(t)sin(\omega t)} \quad (1)$$

where the low-pass signals $\boldsymbol{u_I(t)}$ and $\boldsymbol{u_Q(t)}$ are respectively in-phase and quadrature components of the bandpass signal and $\boldsymbol{\omega}$ is an angular frequency of a carrier signal [12].

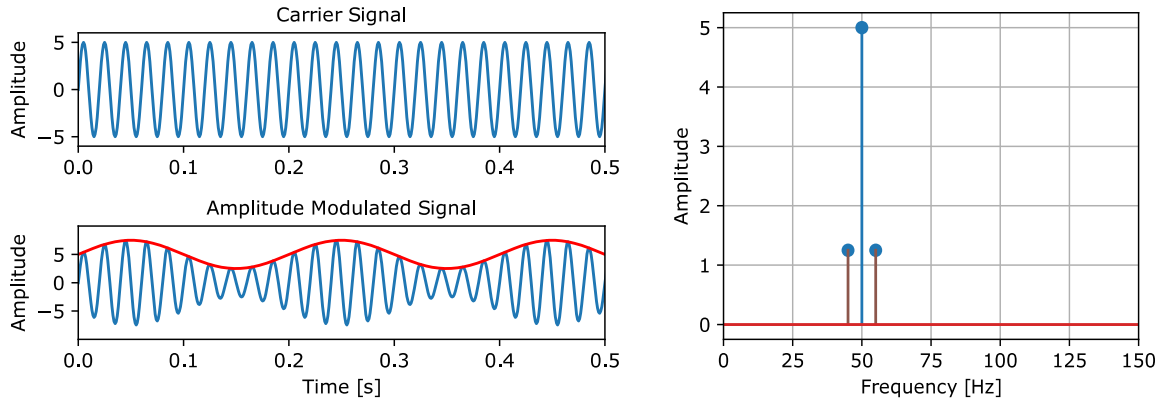


**FIGURE 4.1 AMPLUTUDE-MODULATED SIGNAL WITH CARRIER SIGNAL (LEFT) AND SPECTRUM OF THE MODULATED SIGNAL (RIGHT) [13]**

Then, the dynamic phasor $\langle u \rangle(\mathrm{t})$ is defined as a complex envelope of a signal $\boldsymbol{u(t)}$:

$$\boldsymbol{\langle u \rangle(t) = u_I(t) + ju_Q(t) = (u(t) + jH[u(t)])e^{-j\omega t}} \quad (2)$$

where $\boldsymbol{H}[\boldsymbol{u}(\boldsymbol{t})]$ denotes the Hilbert transform of the signal $\boldsymbol{u}(\boldsymbol{t})$**.** The real part of the dynamic phasor is a red envelope line of an amplitude modulated signal in Figure 4.1. The spectrum of a dynamic phasor $\langle u\rangle(\mathrm{t})$ is a spectrum of bandpass signal $\boldsymbol{u}(\boldsymbol{t})$, shifted by a frequency $\omega$. For power system signals, rated frequency can be selected as frequency shift $\omega$. For intuitive understanding, a three-phase power system in shifted-frequency domain can be compared to modelling each phase of the system in its own Park (d-q) reference frame, rotating with frequency $\omega$.

Table 4.1 compares representation of electrical grid signals and dynamics with an example of inductor equation in each of the modelling domains: electromagnetic transient (EMT), dynamic phasor (DP) and root-mean-square (RMS).

TABLE 4.1 COMPARISON OF EMT, DP, RMS MODELLING DOMAINS

| MODELING DOMAIN | EMT | DP | RMS |
|---|---|---|---|
| SIGNAL | $\mathrm{x(t)}$ | $\langle \mathrm{x}\rangle(\mathrm{t})$ | $\langle \mathrm{x}\rangle$ |
| DYNAMICS | $\frac{dx(t)}{dt}$ | $\langle\frac{dx(\mathrm{t})}{dt}\rangle = \frac{d\langle \mathrm{x}\rangle(\mathrm{t})}{dt} + j\omega\langle \mathrm{x}\rangle(\mathrm{t})$ | $\langle\frac{dx}{dt}\rangle = j\omega\langle \mathrm{x}\rangle$ |
| INDUCTOR EQUATION | $v(t) = L\frac{di(t)}{dt}$ | $\langle v\rangle(\mathrm{t}) = \mathrm{L}\frac{d\langle i\rangle(\mathrm{t})}{dt} + j\omega L\langle i\rangle(\mathrm{t})$ | $\langle v\rangle = j\omega L\langle i\rangle$ |

There are two main motivations behind the application of dynamic phasor modeling:

- The first motivation is the application of higher time steps for dynamic power system simulation, while maintaining good numerical accuracy. The good numerical accuracy with higher time step is achieved in case a frequency band of simulated signals is shifted near to 0 Hz by the frequency shift. The possibility of applying higher time steps is specifically appealing for the purpose of hard real-time simulation. In hard real-time simulations, the process of solving the system of equations for a given discrete time step must be completed within the duration of that time step as measured by the real-time clock. The computation effort to solve the system of equations is independent of time step size. Therefore, application of larger time step gives more time to solve the system of equations and allows simulating larger power systems in hard real-time.
- The second motivation is to bridge the gap between RMS and EMT modeling domains. Results of power system simulation in RMS and EMT modelling domains differ, due to a different mathematical representation of power system equations and, in some cases, due to different modelling depth of power system equipment or control systems. By discretizing the EMT and DP equations in Table 4.1, it can be demonstrated, that when discrete time step grows indefinitely, the DP model converges to RMS representation, while when discrete time step tends towards small values the DP model tends to EMT representation. This way, adjusting the size of time step is expected to allow switching between RMS and EMT power system mathematical representation using DP modelling domain.

# 4.2Inverter modelling in DPsim

DPsim[6] is an open-source tool consisting of solver and model libraries for offline and real-time power system simulation, implemented in C++ programming language. The tool supports EMT, DP and RMS

[6] https://github.com/sogno-platform/dpsim

power system modelling domains. There are two types of inverter models in DPsim: average value model and inverter model including harmonics.

### 4.2.1 Average value inverter model

Average value inverter model is a simplified, computationally efficient representation of an inverter that captures its overall performance and control characteristics without modelling the high frequency switching events of power electronics. DPsim adopts a "modular" philosophy towards average value modelling of inverters. The average value inverter model, as shown in Figure 4.2, consists of modules: electrical grid, control system and interface.

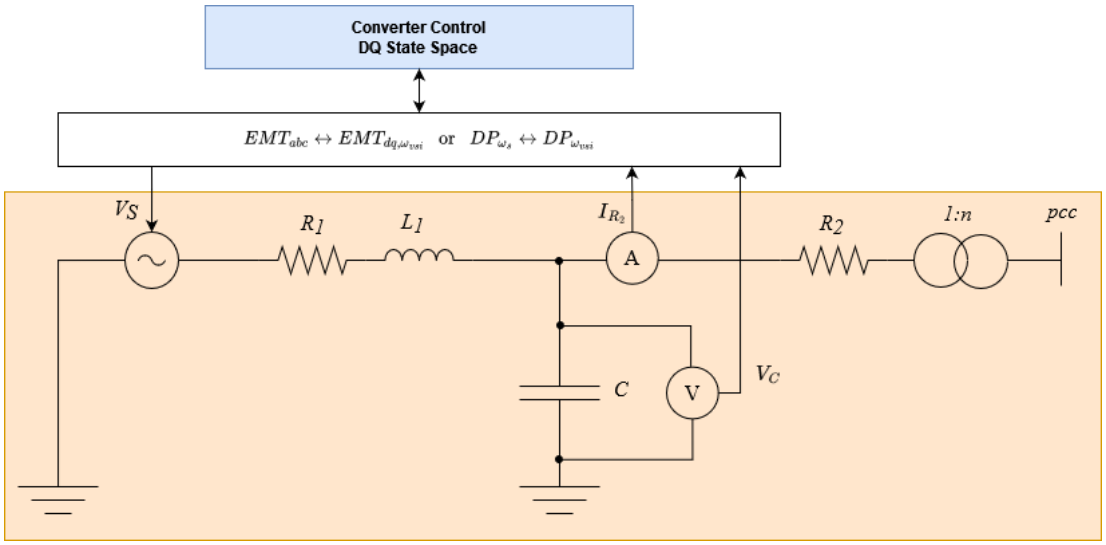


FIGURE 4.2 AVERAGED INVERTER MODELS WITH CONTROLS

The electrical grid model equations are included together with the rest of the power system equations into Modified Nodal Analysis (MNA) power system model. The power system model, including inverter's AC filter and transformer, can be represented in any of the modelling domains: EMT, DP or RMS. This way the modular philosophy allows combining the same control system with different power system modelling domains. The control system can be of grid-following (see Figure 4.3) or grid-forming (see Figure 4.4) type and is modelled in "dq" (Park) reference frame. The grid-following type controls active and reactive power injection and consists of outer power and inner current control loops, implemented as PI controllers. The synchronization to grid voltage is performed using phase-locked loop (PLL). The control architecture has been validated against open-source implementation in Modelica[7].

[7] https://github.com/ModPowerSystems/ModPowerSystems

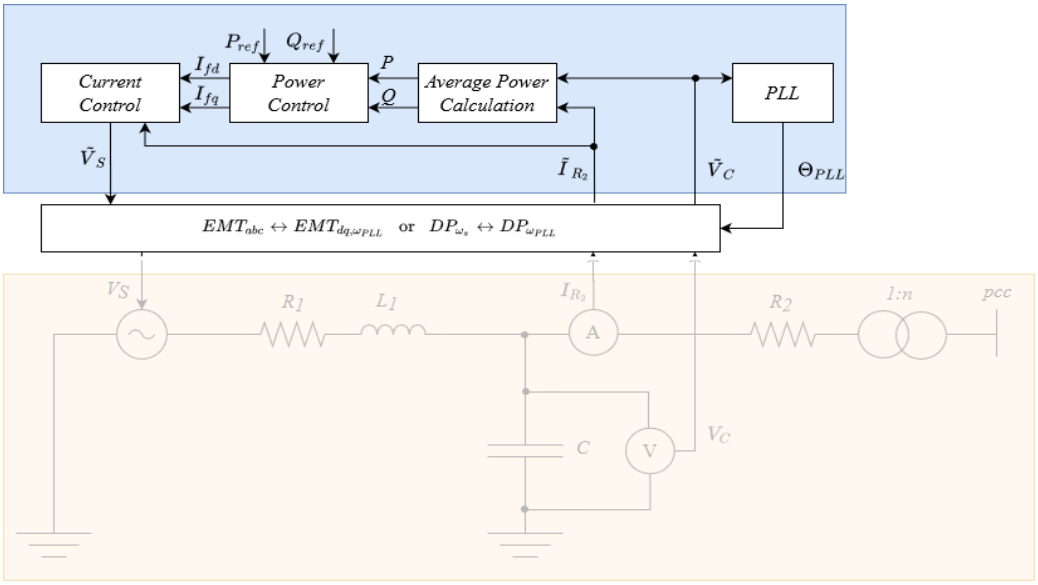


FIGURE 4.3 GRID FOLLOWING CONTROL ARCHITECTURE

The grid-forming type controls voltage magnitude and phase, consists of outer voltage and inner current control loops. The model can operate in isolated and grid-connected (with a droop control) modes.

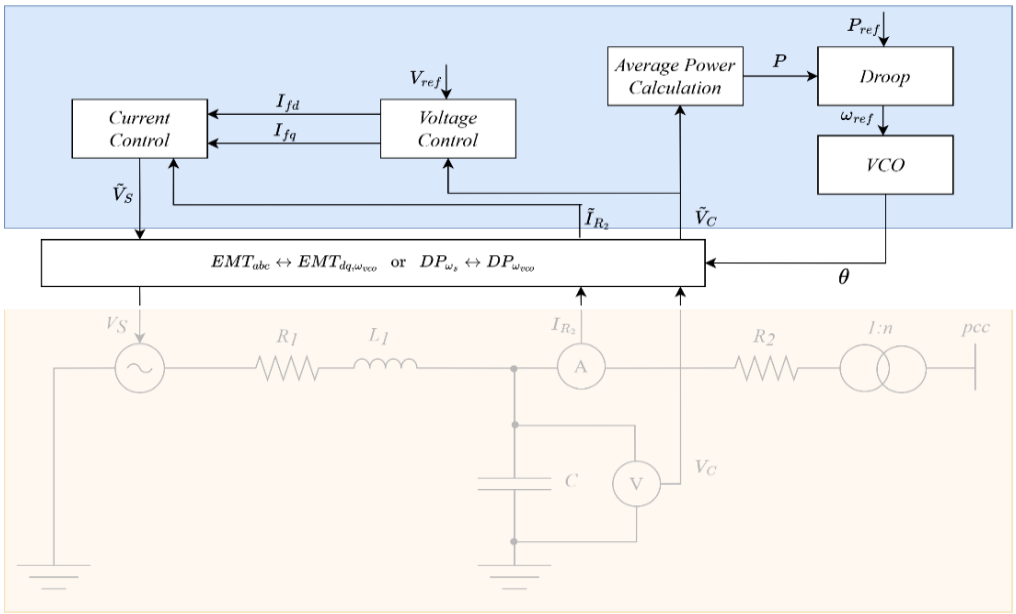


FIGURE 4.4 GRID FORMING CONTROL ARCHITECTURE

The interface between the control system and the electrical grid is established using a voltage source and measurements on LC filer. The interface currently implemented for average value inverter model in DPsim follows a partitioned solution approach, meaning that the power system equations, including inverter electrical grid equations, are solved separately from control system equations. The partitioning is achieved by a time delay, which can be also viewed as numerical discretization of control system equations with Euler forward method.

### 4.2.2 Inverter model including harmonics

The inverter model including harmonics has been developed in DPsim as a part of PhD dissertation [13]. The model has a grid-following control architecture. The inverter model including harmonics approximates high frequency oscillations from power electronic switching events using harmonic analysis and parallel simulation with multiple dynamic phasors. Through harmonic analysis, several dynamic phasors are computed from inverter model's voltage signal. Multiple inverter voltage dynamic phasors are "superposed" on the same power system through replicating a power system model in multiple shifted frequency domains (each shifted at its own harmonic) and injecting each harmonic's dynamic phasor using voltage source (see schematic in Figure 4.5). The application of the model is limited to DP (or, better say "multi-DP") modelling domain. The challenges of this approach include increased computation effort in case multiple harmonics are simulated and the need to know the expected harmonics before the simulation. The research question of application of multiple DPs for simulation of inverter switching harmonics is well covered in [13], further research and development in this particular area is currently not continued within the DPsim development team.

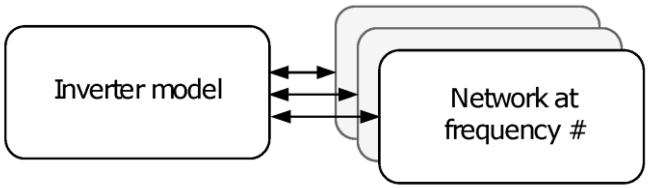


FIGURE 4.5 INTERACTION OF HARMONIC INVERTER MODEL WITH MULTI-DP REPRESENTATION OF POWER SYSTEM [14]

## 4.3 Comparison between simulation results in EMT, DP and RMS

The comparison is presented for average value inverter model, because it is supported in all power system modelling domains: EMT, DP and RMS. The grids, used for comparison of grid-following and grid forming inverter simulation results are shown in Figure 4.6 and Figure 4.8 respectively.

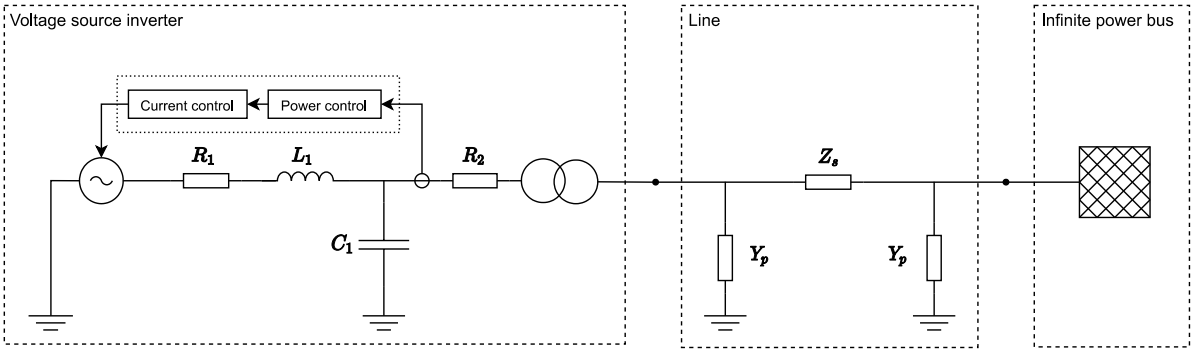


FIGURE 4.6 GRID FOR COMPARISON OF GRID-FOLLOWING INVERTER SIMULATION RESULTS

Figure 4.7 shows voltage at the point of coupling of inverter to the grid as a result of simulation in three modelling domains: EMT, DP and RMS (RMS modelling domain is denoted as "SP" – static phasor in the legend of the Figure 4.7). The simulation scenario includes a voltage frequency ramp from transmission system, starting at 5 seconds. The frequency ramp is simulated by varying the frequency of infinite power bus's voltage source. The simulation duration is selected large enough to include the steady state before the frequency ramp, transient period due to frequency ramp and steady state

after the frequency ramp. EMT simulation with a very small time step (1µs) is considered as the reference. As expected, DP simulation converges to EMT simulation in case of small time steps (10 µs), while still keeps reasonably accurate simulation results, especially for slower transients, when larger time step is applied (1 ms). The RMS simulation result is different from EMT even with very small step size (10 µs). The computational complexity for solution of system of equations, assuming Gaussian elimination (worst case) is O(n^3), where n is the number of equations. DP and RMS models are expressed using complex number equations, meaning that the number of equations is doubled compared to EMT. For the same time step size, the computational complexity for DP and RMS is 8 times higher than for EMT. However, increase of time step in DP simulation saves the computational effort. For example, simulation of the model in DP domain with 1 ms time step is expected to be at least 125 times faster than simulation of EMT domain model with 1 µs time step.

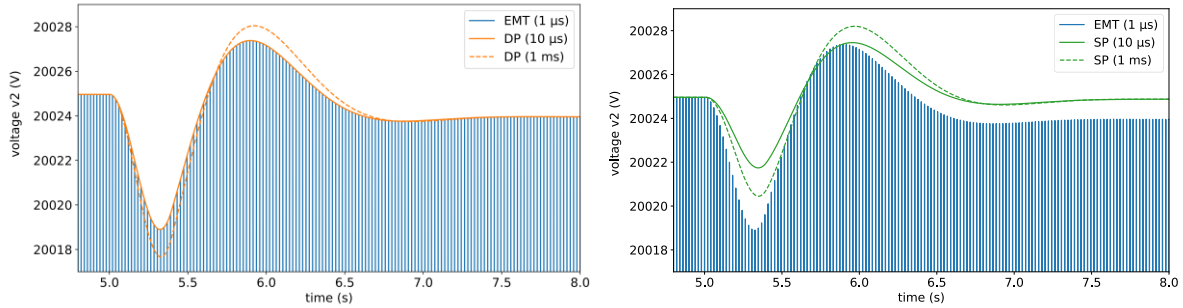


FIGURE 4.7 COMPARISON OF GRID-FOLLOWING INVERTER SIMULATION RESULTS

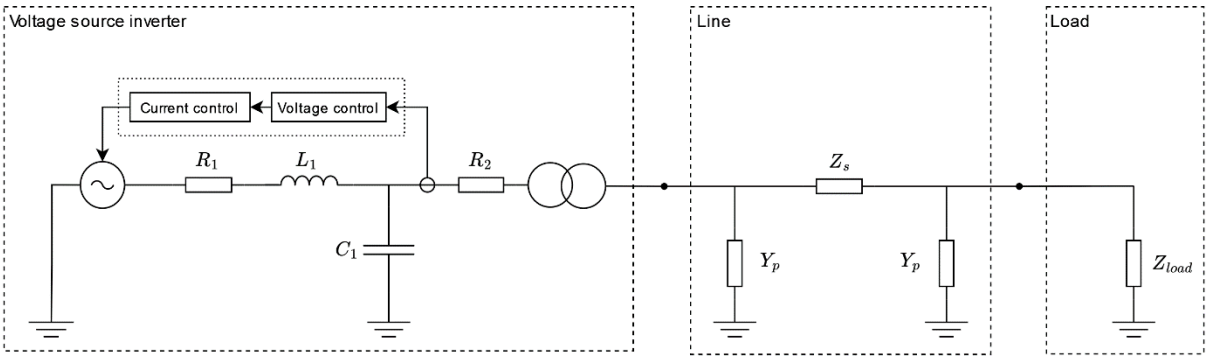


FIGURE 4.8 GRID FOR COMPARISON OF GRID-FORMING INVERTER SIMULATION RESULTS

The Figure 4.9 shows voltage at the point of coupling of inverter to the grid as a result of simulation in two modelling domains: EMT and DP. The simulation scenario includes a large step change of load. This scenario is chosen to illustrate grid-forming inverter operation isolated from the power system. RMS simulation results are not included in the Figure 4.9, as the simulation in RMS modelling domain is numerically unstable. The DPsim developers assume that numerical instability is caused by the partitioned solution approach in the implemented interface between control and electrical grid equations. Real part of simulated DP is an accurate envelope of simulated EMT signal.

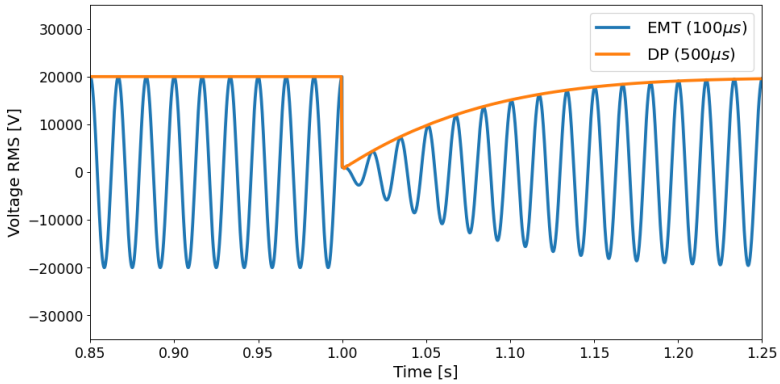


FIGURE 4.9 COMPARISON OF GRID-FORMING INVERTER SIMULATION RESULTS

## 4.4 Current work, limitations, and challenges

The DP inverter model including harmonics and DP average value inverter model with grid-following controls were available in DPsim before the start of the Task 4.2 in HVDC-Wise. Currently, dynamic phasor inverter modelling work in DPsim is further focused on average value inverter model. As a part of the Task 4.2 the average value grid-forming inverter model was implemented in DPsim in EMT, DP and RMS modelling domains. The model is currently available in the feature branch of DPsim on GitHub, it is planned that the model will be merged with the main branch. The implementation includes development of grid-forming controls and creation of modular inverter models for each of the modelling domains. Furthermore, the developed models were tested by comparing simulation results against benchmark implementation in an EMT commercial software and against each other. There are several further research and development topics outlined in this area thanks to the work in HVDC-Wise:

1. Development of a numerically stable interface to couple electrical grid equations with control equations. In order that the DP modelling domain can truly bridge the gap between EMT and RMS modelling domains for simulation of inverters, it is required that DP simulation is numerically stable both with typical EMT time step sizes (in the range of µs) and RMS step sizes (in the range of ms). Partitioned solution approach, although computationally efficient, leads to numerical instability when applied with larger time steps. This limits the advantages of DP modelling domain and makes grid-forming inverter model not applicable with RMS modelling domain.
2. Enhancement of control system modelling depth. For example, the grid following control architecture currently implemented in DPsim relies on a single PLL, that allows to correctly synchronize only to a balanced set of three phase voltages. Therefore, the implemented control architecture is not adequate for simulation of unbalanced grid conditions, e.g. unsymmetrical faults, that can be of interest for EMT studies.
3. Modelling of DC part of the grid, interfaced by a converter to the AC grid. Unlike AC grid, the DC part of the grid does not have a rated frequency, that can be interpreted as “carrier” frequency and used for frequency shift in DP modelling domain. For that reason, it makes most sense to represent DC part of the grid with EMT domain equations even for DP domain power system model. The challenge and effort mostly lie in implementation of a solver that would handle both the DP and EMT parts of the model in one simulation.

As has been outlined in further research and development topics, there are limitations in existing capabilities of DPsim to model and simulate hybrid AC/DC grids for validation of control and protection concepts. Despite these limitations, DPsim can be potentially used in studies in AC part of the grid, where the focus is set on comparing the simulation results achieved with the same models in different modelling domains (EMT, DP, RMS), such as converter interaction studies or stability studies in inverter-dominated grids. The outcome of these studies can be understanding of the influence of modelling domain on simulation results: for instance, if a simulation instability is caused by an actual power system instability or by numerical problems, associated with modelling domain, numerical integration or interfacing of inverter control equations.

# 5. Conclusion

Two modelling approaches for DIgSILENT PowerFactory have been presented. The first approach uses the DIgSILENT Simulation Language (DSL) and is quite convenient for simple models although it requires the users to be familiar with this specific framework and its complex interface. The second approach relies on model exchange facilitated by the FMI standard. It utilizes compiled models from potentially other software platforms (Matlab Simulink in this deliverable). It enables the sharing of models without disclosing detailed information, thereby preserving intellectual property. It fosters the reuse of previously developed control schemes. Nevertheless, this approach has certain export limitations and the debugging process can be challenging. Several models have been developed according to one of these modelling approaches. In one case (MMC GFM) both approaches have been used, and similar results have been obtained. These models are shared with the project partners for potential further use for use cases (WP6).

The feasibility of standardized model exchange for HVDC power systems' static data has been demonstrated by leveraging the IEC CIM/CGMES standards. Moreover, a workflow has been proposed with variants for enabling the standardized exchange of the dynamic parts of the HVDC models described by either Modelica equations, Matlab equations or compiled file according to FMI standard. This workflow has been applied to build the models in deliverable D4.3 (HVDC-Wise Lib) and provide a structured approach to facilitate interoperability between different modelling and simulation tools. Moreover, key contributions include the identification of gaps in existing standards, the development of CIM extensions for improved static and dynamic model exchanges, and the establishment of methodologies to support both open-source and compiled model sharing. This deliverable has also underscored the need for collaboration between TSOs, vendors, and standardization bodies to address challenges related to intellectual property, model conversion, and interoperability. Moving forward, efforts should focus on further refining the library, conducting interoperability testing, and ensuring cross-validation of simulation results to enhance model accuracy and usability. The advancements which resulted from this deliverable will enable industry stakeholders to integrate HVDC technologies more effectively, optimize system planning and operational studies, and support ongoing standardization initiatives.

The principle and current status of power electronics modelling in DP domain has been presented. The new developments regarding modelling of grid-forming inverter in DP domain and open-source implementation in DP, EMT and RMS domains has been discussed. The first tests and simulation in DP domain in open-source tool DPsim demonstrate the potential advantage of DP modelling domain. Firstly, simulation in DP domain can include the electrical grid dynamics, that are not available from RMS simulation but relevant for study of converter interaction phenomena. Secondly, application of higher time step in DP modelling domain can reduce computation time compared to simulation in EMT modelling domain. Besides the new development results, the report as well addresses the limitations and further research and development questions. These are: (i) the need to develop and implement novel interfaces between power electronics models and solver to increase numerical stability of simulation, (ii) the necessity to increase modelling depth of converter controls implemented in DPsim to support unbalanced grid conditions, and (iii) the requirement to model DC side of the grid. The grid-forming and grid-following inverter models, implemented in DPsim, will be used in further steps of the project (WP7) to investigate the relevance of assessing converters interactions in DP domain.

# A. Appendix: Proposed CIM Extensions

This section provides detailed information on extensions related to DC modelling proposed by HVDC WISE project. This content is based on an export form Enterprise Architect.

## 1. Package ExtDCEquipment

This section contains extensions related to the DC equipment. Figure A.1 shows extensions related to the DC storage.

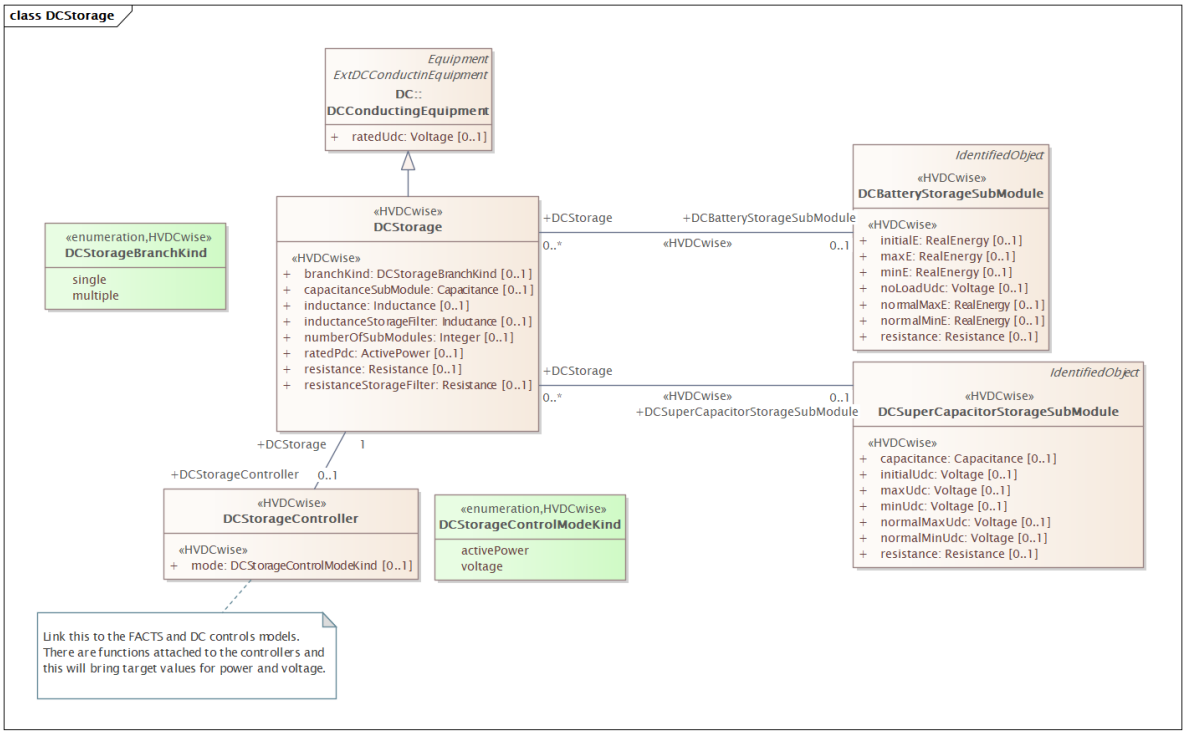


FIGURE A.1 CLASS DIAGRAM EXTDCEQUIPMENT::DCSTORAGE

### 1.1 (HVDCwise) DCStorage

*DCStorage* is for direct current storage. It is a two-terminal device. *DCStorage* has been primarily defined for DC-connected single branch of series-connected submodules with energy storage but can be used for models with several branches in parallel. Table A.1 shows all attributes of *DCStorage*. Table A.2 shows all association ends of *DCStorage* with other classes.

TABLE A.1 ATTRIBUTES OF EXTDCEQUIPMENT::DCSTORAGE

| name | mult | type | description |
|---|---|---|---|
| ratedPdc | 0..1 | ActivePower | (HVDCwise) Rated DC power of the converter. The attribute shall be positive value. |
| inductance | 0..1 | Inductance | (HVDCwise) Inductance of the branch inductor. The attribute shall be positive value. |
| resistance | 0..1 | Resistance | (HVDCwise) Resistance of the branch inductor. The attribute shall be positive value. |

| name | mult | type | description |
|---|---|---|---|
| capacitanceSubModule | 0..1 | Capacitance | (HVDCwise) Capacitance of the sub-module capacitor. The attribute shall be positive value. |
| numberOfSubModules | 0..1 | Integer | (HVDCwise) Number of submodules. |
| inductanceStorageFilter | 0..1 | Inductance | (HVDCwise) Storage filter inductance for each sub-module (Lsm). Used for calculation of the inductance of the equivalent circuit. The attribute shall be positive value. |
| resistanceStorageFilter | 0..1 | Resistance | (HVDCwise) Storage filter inductor resistance for each sub-module (Rsm). Used for calculation of the resistance of the equivalent circuit. The attribute shall be positive value. |
| branchKind | 0..1 | DCStorageBranchKind | (HVDCwise) Kind of DC storage in terms of number of branches. |
| ratedUdc | 0..1 | Voltage | inherited from: DCConductingEquipment |
| aggregate | 0..1 | Boolean | inherited from: Equipment |
| inService | 0..1 | Boolean | inherited from: Equipment |
| networkAnalysisEnabled | 0..1 | Boolean | inherited from: Equipment |
| normallyInService | 0..1 | Boolean | inherited from: Equipment |
| aliasName | 0..1 | String | inherited from: IdentifiedObject |
| description | 0..1 | String | inherited from: IdentifiedObject |
| mRID | 0..1 | String | inherited from: IdentifiedObject |
| name | 0..1 | String | inherited from: IdentifiedObject |
| energyIdentCodeEic | 0..1 | String | (European) inherited from: ExtEuIdentifiedObject |
| shortName | 0..1 | String | (European) inherited from: ExtEuIdentifiedObject |
| ratedCurrent | 0..1 | CurrentFlow | inherited from: ExtDCConductinEquipment |

**TABLE A.2 ASSOCIATION ENDS OF EXTDCEQUIPMENT::DCSTORAGE WITH OTHER CLASSES**

| mult from | name | mult to | type | description |
|---|---|---|---|---|
| 0..* | DCBatteryStorageSubModule | 0..1 | DCBatteryStorageSubModule | (HVDCwise) The DC battery storage submodule for this DC storage. |
| 0..* | DCSuperCapacitorStorageSubModule | 0..1 | DCSuperCapacitorStorageSubModule | (HVDCwise) The DC super capacitor submodule for this DC storage. |
| 1..1 | DCStorageController | 0..1 | DCStorageController | The DC storage controller for this DC storage device. |
| 1..1 | SvDCStorage | 0..* | SvDCStorage | (HVDCwise) The state variables of a DC storage. |
| 1..1 | DCTerminals | 0..* | DCTerminal | inherited from: DCConductingEquipment |
| 0..* | AggregatedEquipment | 0..1 | Equipment | (NC) inherited from: Equipment |
| 0..1 | OperationalLimitSet | 0..* | OperationalLimitSet | inherited from: Equipment |
| 1..1 | ContingencyEquipment | 0..* | ContingencyEquipment | inherited from: Equipment |
| 0..* | EquipmentContainer | 0..1 | EquipmentContainer | inherited from: Equipment |
| 0..1 | Faults | 0..* | Fault | inherited from: Equipment |
| 0..* | AdditionalEquipmentContainer | 0..* | EquipmentContainer | inherited from: Equipment |
| 0..1 | DetailedModelDynamics | 0..* | DetailedModelDynamics | inherited from: Equipment |

| mult from | name | mult to | type | description |
|---|---|---|---|---|
| 0..1 | DetailedEquipment | 0..* | Equipment | (NC) inherited from: Equipment |
| 0..* | PSRType | 0..1 | PSRType | inherited from: PowerSystemResource |
| 0..1 | Controls | 0..* | Control | inherited from: PowerSystemResource |
| 0..1 | Measurements | 0..* | Measurement | inherited from: PowerSystemResource |
| 1..1 | OperatingShare | 0..* | OperatingShare | inherited from: PowerSystemResource |
| 0..* | ReportingGroup | 0..* | ReportingGroup | inherited from: PowerSystemResource |
| 0..1 | DiagramObjects | 0..* | DiagramObject | inherited from: IdentifiedObject |
| 1..1 | Names | 0..* | Name | inherited from: IdentifiedObject |
| 0..1 | ParameterEvent | 0..* | ParameterEvent | inherited from: IdentifiedObject |
| 0..1 | AlternativeIdentifier | 0..* | Name | (NC) inherited from: IdentifiedObject |
| 0..1 | Name | 0..* | Name | (NC) inherited from: IdentifiedObject |

## 1.2 (HVDCwise) DCStorageBranchKind enumeration

*DCStorageBranchKind* enumeration is for kind of DC storage in terms of number of branches. Table A.3 shows all literals of *DCStorageBranchKind*.

TABLE A.3 LITERALS OF EXTDCEQUIPMENT::DCSTORAGEBRANCHKIND

| literal | value | description |
|---|---|---|
| single | | DC storage is a single branch of series-connected submodules with energy storage. |
| multiple | | DC storage is made of multiple branches of series-connected submodules with energy storage. |

## 1.3 (HVDCwise) DCSuperCapacitorStorageSubModule

*DCSuperCapacitorStorageSubModule* is for supercapacitor storage submodule of the DC storage device. Table A.4 shows all attributes of *DCSuperCapacitorStorageSubModule*. Table A.5 shows all association ends of *DCSuperCapacitorStorageSubModule* with other classes.

TABLE A.4 ATTRIBUTES OF EXTDCEQUIPMENT::DCSUPERCAPACITORSTORAGESUBMODULE

| name | mult | type | description |
|---|---|---|---|
| capacitance | 0..1 | Capacitance | (HVDCwise) Capacitance of supercapacitor per sub-module (Csc). Used for calculation of the capacitance of the equivalent circuit. The attribute shall be positive value. |
| resistance | 0..1 | Resistance | (HVDCwise) Resistance of supercapacitor per sub-module (Rsc). Used for calculation of the resistance of the equivalent circuit. The attribute shall be positive value. |
| normalMaxUdc | 0..1 | Voltage | (HVDCwise) Normal maximum permissible supercapacitor voltage. When the supercapacitor voltage is superior to this value, |

| name | mult | type | description |
|---|---|---|---|
| | | | the power that the device can absorb is reduced. The attribute shall be positive value. |
| maxUdc | 0..1 | Voltage | (HVDCwise) Maximum permissible supercapacitor voltage. When the supercapacitor voltage is superior to this value, the power that the device can absorb is set to zero. The attribute shall be positive value. |
| normalMinUdc | 0..1 | Voltage | (HVDCwise) Normal minimum permissible supercapacitor voltage. When the supercapacitor voltage is inferior to this value, the power that the device can provide is reduced. The attribute shall be positive value. |
| minUdc | 0..1 | Voltage | (HVDCwise) Minimum permissible supercapacitor voltage. When the supercapacitor voltage is inferior to this value, the power that the device can provide is set to zero. The attribute shall be positive or zero value. |
| initialUdc | 0..1 | Voltage | (HVDCwise) Initial voltage of the capacitance of the super capacitor (Csc). The attribute shall be positive or zero value. |
| aliasName | 0..1 | String | inherited from: IdentifiedObject |
| description | 0..1 | String | inherited from: IdentifiedObject |
| mRID | 0..1 | String | inherited from: IdentifiedObject |
| name | 0..1 | String | inherited from: IdentifiedObject |
| energyIdentCodeEic | 0..1 | String | (European) inherited from: ExtEuIdentifiedObject |
| shortName | 0..1 | String | (European) inherited from: ExtEuIdentifiedObject |

**TABLE A.5 ASSOCIATION ENDS OF EXTDCEQUIPMENT::DCSUPERCAPACITORSTORAGESUBMODULE WITH OTHER CLASSES**

| mult from | name | mult to | type | description |
|---|---|---|---|---|
| 0..1 | DCStorage | 0..* | DCStorage | (HVDCwise) The DC storage that has this DC super capacitor submodule. |
| 1..1 | SvDCSuperCapacitor | 0..* | SvDCSuperCapacitor | (HVDCwise) The state variavles for this DC super capacitor. |
| 0..1 | DiagramObjects | 0..* | DiagramObject | inherited from: IdentifiedObject |
| 1..1 | Names | 0..* | Name | inherited from: IdentifiedObject |
| 0..1 | ParameterEvent | 0..* | ParameterEvent | inherited from: IdentifiedObject |
| 0..1 | AlternativeIdentifier | 0..* | Name | (NC) inherited from: IdentifiedObject |
| 0..1 | Name | 0..* | Name | (NC) inherited from: IdentifiedObject |

## 1.4 (HVDCwise) DCBatteryStorageSubModule

*DCBatteryStorageSubModule* is for battery storage submodule of the DC storage device. Table A.6 shows all attributes of *DCBatteryStorageSubModule*. Table A.7 shows all association ends of *DCBatteryStorageSubModule* with other classes.

TABLE A.6 ATTRIBUTES OF EXTDCEQUIPMENT::DCBATTERYSTORAGESUBMODULE

| name | mult | type | description |
|---|---|---|---|
| resistance | 0..1 | Resistance | (HVDCwise) Internal battery resistance per sub-module (Rbat). Used for calculation of the resistance of the equivalent circuit. The attribute shall be positive value. |
| noLoadUdc | 0..1 | Voltage | (HVDCwise) No load battery voltage per sub-module (Vbat). Used for calculation of the equivalent battery voltage. The attribute shall be positive value. |
| normalMaxE | 0..1 | RealEnergy | (HVDCwise) Normal maximum battery energy. When the battery energy is superior to this value, the power that the device can absorb is reduced. The attribute shall be positive value. |
| maxE | 0..1 | RealEnergy | (HVDCwise) Maximum battery energy. When the battery energy is superior to this value, the power that the device can absorb is set to zero. The attribute shall be positive value. |
| normalMinE | 0..1 | RealEnergy | (HVDCwise) Normal minimum battery energy. When the battery energy is inferior to this value, the power that the device can provide is reduced. The attribute shall be positive value. |
| minE | 0..1 | RealEnergy | (HVDCwise) Minimum battery energy. When the battery energy is inferior to this value, the power that the device can provide is set to zero. The attribute shall be positive value. |
| initialE | 0..1 | RealEnergy | (HVDCwise) Initial energy of the battery storage. The attribute shall be positive value. |
| aliasName | 0..1 | String | inherited from: IdentifiedObject |
| description | 0..1 | String | inherited from: IdentifiedObject |
| mRID | 0..1 | String | inherited from: IdentifiedObject |
| name | 0..1 | String | inherited from: IdentifiedObject |
| energyIdentCodeEic | 0..1 | String | (European) inherited from: ExtEuIdentifiedObject |
| shortName | 0..1 | String | (European) inherited from: ExtEuIdentifiedObject |

TABLE A.7 ASSOCIATION ENDS OF EXTDCEQUIPMENT::DCBATTERYSTORAGESUBMODULE WITH OTHER CLASSES

| mult from | name | mult to | type | description |
|---|---|---|---|---|
| 0..1 | DCStorage | 0..* | DCStorage | (HVDCwise) The DC storage that has this DC battery storage. |
| 1..1 | SvDCBattery | 0..* | SvDCBattery | (HVDCwise) The state variable of the DC battery submodule. |
| 0..1 | DiagramObjects | 0..* | DiagramObject | inherited from: IdentifiedObject |
| 1..1 | Names | 0..* | Name | inherited from: IdentifiedObject |
| 0..1 | ParameterEvent | 0..* | ParameterEvent | inherited from: IdentifiedObject |
| 0..1 | AlternativeIdentifier | 0..* | Name | (NC) inherited from: IdentifiedObject |
| 0..1 | Name | 0..* | Name | (NC) inherited from: IdentifiedObject |

## 1.5 (HVDCwise) DCStorageController root class

*DCStorageController* is for Controller of the DC storage. Table A.8 shows all attributes of *DCStorageController*. Table A.9 shows all association ends of *DCStorageController* with other classes.

TABLE A.8 ATTRIBUTES OF EXTDCEQUIPMENT::DCSTORAGECONTROLLER

| name | mult | type | description |
|---|---|---|---|
| mode | 0..1 | DCStorageControlModeKind | (HVDCwise) Direct current storage control mode. |

TABLE A.9 ASSOCIATION ENDS OF EXTDCEQUIPMENT::DCSTORAGECONTROLLER WITH OTHER CLASSES

| mult from | name | mult to | type | description |
|---|---|---|---|---|
| 0..1 | DCStorage | 1..1 | DCStorage | The DC storage device that has this controller. |

## 1.6 (HVDCwise) DCStorageControlModeKind enumeration

*DCStorageControlModeKind* enumeration is for kinds of control modes of the DC storage. Table A.10 shows all literals of *DCStorageControlModeKind*.

TABLE A.10 LITERALS OF EXTDCEQUIPMENT::DCSTORAGECONTROLMODEKIND

| literal | value | description |
|---|---|---|
| activePower | | In this control mode, the reference value is an active power to be provided or absorbed by the device. |
| voltage | | In this control mode, the reference value corresponds to the DC voltage at the device terminals. |

# 2. Package ExtDCTerminal

This section contains extensions related to the DC Terminal. Figure A.2 shows extensions related to the *DCTerminal*.

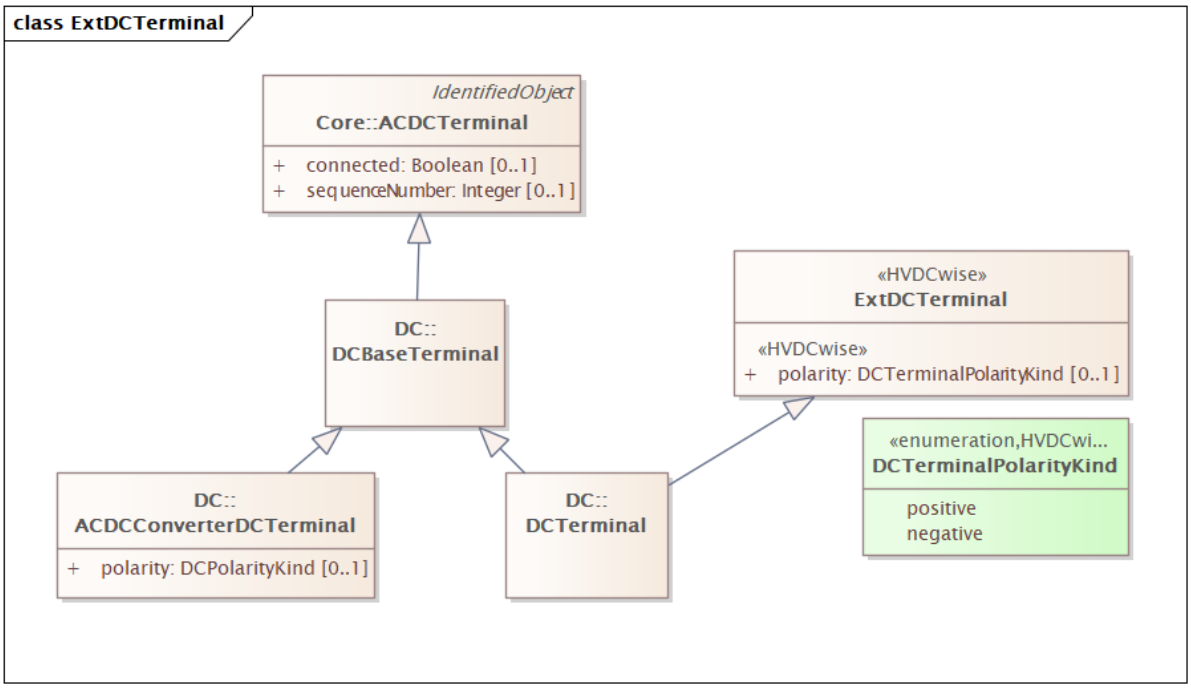


FIGURE A.2 CLASS DIAGRAM EXTDCTERMINAL::EXTDCTERMINAL

## 2.1 (HVDCwise) ExtDCTerminal root class

This is an extended class. Table A.11 shows all attributes of *ExtDCTerminal*.

TABLE A.11 ATTRIBUTES OF EXTDCTERMINAL::EXTDCTERMINAL

| name | mult | type | description |
|---|---|---|---|
| polarity | 0..1 | DCTerminalPolarityKind | (HVDCwise) Represents the normal network polarity condition. Used in DC system configurations that have explicit polarity of the terminals, e.g., voltage source converter (VSC) technology. |

## 2.2 (HVDCwise) DCTerminalPolarityKind enumeration

*DCTerminalPolarityKind* is for polarity for DC terminal. It is used in DC system configurations that have explicit polarity of the terminals, e.g., voltage source converter (VSC) technology. Table A.12 shows all literals of *DCTerminalPolarityKind*.

TABLE A.12 LITERALS OF EXTDCTERMINAL::DCTERMINALPOLARITYKIND

| Literal | value | description |
|---|---|---|
| Positive | | Positive terminal. |
| Negative | | Negative terminal. |

# 3. Package ExtDCStateVariables

This section contains extensions to the state variables. Figure A.3 shows extensions to the state variables.

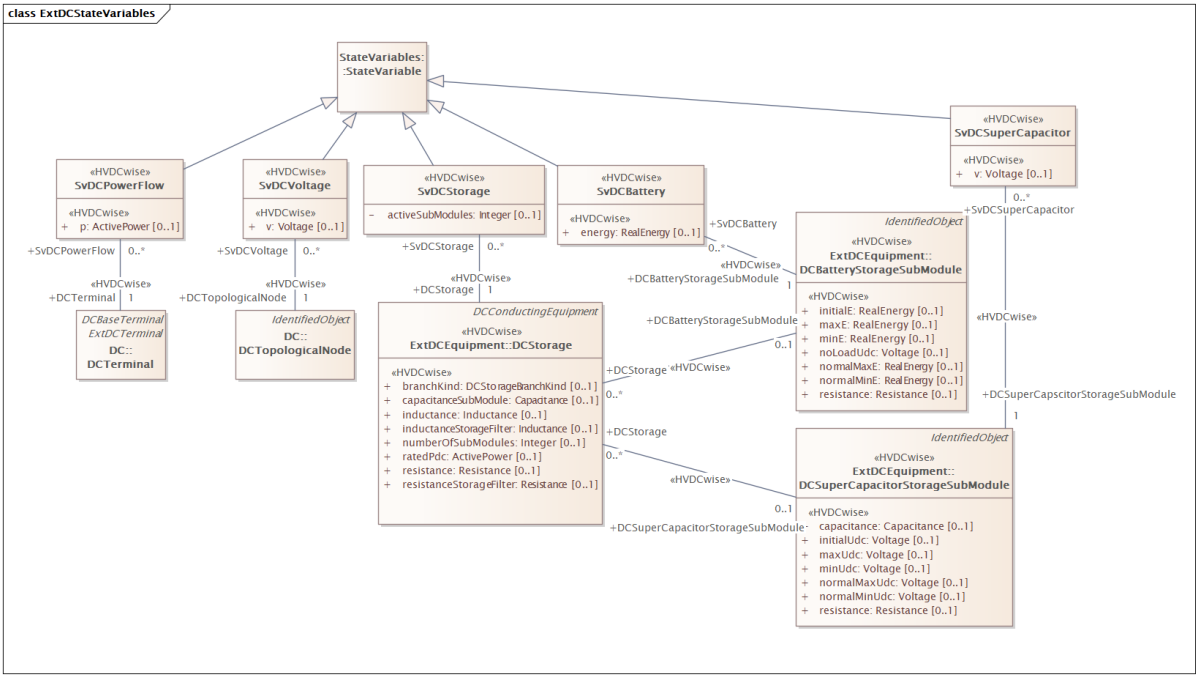


FIGURE A.3 CLASS DIAGRAM EXTDCSTATEVARIABLES::EXTDCSTATEVARIABLES

## 3.1 (HVDCwise) SvDCVoltage

*SvDCVoltage* is for state variable for direct current voltage. Table A.13 shows all attributes of *SvDCVoltage*. Table A.14 shows all association ends of *SvDCVoltage* with other classes.

TABLE A.13 ATTRIBUTES OF EXTDCSTATEVARIABLES::SVDCVOLTAGE

| name | mult | type | description |
|---|---|---|---|
| v | 0..1 | Voltage | (HVDCwise) State variable for direct current voltage. |

TABLE A.14 ASSOCIATION ENDS OF EXTDCSTATEVARIABLES::SVDCVOLTAGE WITH OTHER CLASSES

| mult from | name | mult to | type | description |
|---|---|---|---|---|
| 0..* | DCTopologicalNode | 1..1 | DCTopologicalNode | (HVDCwise) The DC topological node associated with the DC voltage state. |

## 3.2 (HVDCwise) SvDCPowerFlow

*SvDCPowerFlow* is for state variable for power flow. Load convention is used for flow direction. This means flow out from the *DCTopologicalNode* into the equipment is positive. Table A.15 shows all attributes of *SvDCPowerFlow*. Table A.16 shows all association ends of *SvDCPowerFlow* with other classes.

TABLE A.15 ATTRIBUTES OF EXTDCSTATEVARIABLES::SVDCPOWERFLOW

| name | Mult | type | Description |
|---|---|---|---|
| p | 0..1 | ActivePower | (HVDCwise) The active power flow. Load sign convention is used, i.e. positive sign means flow out from a DCTopologicalNode (bus) into the conducting equipment. |

TABLE A.16 ASSOCIATION ENDS OF EXTDCSTATEVARIABLES::SVDCPOWERFLOW WITH OTHER CLASSES

| mult from | Name | mult to | type | Description |
|---|---|---|---|---|
| 0..* | DCTerminal | 1..1 | DCTerminal | (HVDCwise) The DC power flow state variable associated with the DC terminal. |

## 3.3 (HVDCwise) SvDCStorage

*SvDCStorage* is for state variable for direct current storage. Table A.17 shows all association ends of *SvDCStorage* with other classes.

TABLE A.17 ASSOCIATION ENDS OF EXTDCSTATEVARIABLES::SVDCSTORAGE WITH OTHER CLASSES

| mult from | name | mult to | type | description |
|---|---|---|---|---|
| 0..* | DCStorage | 1..1 | DCStorage | (HVDCwise) The DC storage that has this state variables. |

## 3.4 (HVDCwise) SvDCBattery

*SvDCBattery* is for state variable for direct current battery. Table A.18 shows all attributes of *SvDCBattery*. Table A.19 shows all association ends of *SvDCBattery* with other classes.

TABLE A.18 ATTRIBUTES OF EXTDCSTATEVARIABLES::SVDCBATTERY

| name | mult | type | description |
|---|---|---|---|
| energy | 0..1 | RealEnergy | (HVDCwise) Battery energy. The attribute shall be positive value or zero. |

TABLE A.19 ASSOCIATION ENDS OF EXTDCSTATEVARIABLES::SVDCBATTERY WITH OTHER CLASSES

| mult from | name | mult to | type | description |
|---|---|---|---|---|
| 0..* | DCBatteryStorageSubModule | 1..1 | DCBatteryStorageSubModule | (HVDCwise) The DC battery submodule that has this state variables. |

## 3.5 (HVDCwise) SvDCSuperCapacitor

*SvDCSuperCapacitor* is for state variable for direct current super capacitor. Table A.20 shows all attributes of *SvDCSuperCapacitor*. Table A.21 shows all association ends of *SvDCSuperCapacitor* with other classes.

TABLE A.20 ATTRIBUTES OF EXTDCSTATEVARIABLES::SVDCSUPERCAPACITOR

| name | mult | type | description |
|---|---|---|---|
| v | 0..1 | Voltage | (HVDCwise) State variable for direct current voltage of the super capacitor. |

TABLE A.21 ASSOCIATION ENDS OF EXTDCSTATEVARIABLES::SVDCSUPERCAPACITOR WITH OTHER CLASSES

| mult from | name | mult to | type | description |
|---|---|---|---|---|
| 0..* | DCSuperCapscitorStorageSubModule | 1..1 | DCSuperCapacitorStorageSubModule | (HVDCwise) The DC super capacitor submodule that has this state variables. |

# References


[1] "HVDC-WISE Deliverable 4.1: Identification of key technologies, potential benefits and restrictions," 2023.

[2] F. Errigo, J. Sau Bassols, H. Bekkouri, F. Morel, J. C. Gonzalez-Torres, A. Benchaib, P. Rault and X. Bourgeat, "Design of a Single Branch of Energy Storage Submodules Connected to HVDC Systems to Support AC Grids," *MDPI Electronics,* vol. 13, no. 17, 2024.

[3] J. Sau Bassols, F. Morel, F. Errigo, H. Bekkouri and J. C. Gonzalez-Torres, "Single branch of energy storage submodules to integrate energy storage devices in HVDC systems," in *19th International Conference on AC and DC Power Transmission (ACDC 2023)*, Glasgow, United Kingdom, Mar 2023.

[4] "Towards Complete HVDC Model Exchange based on IEC CIM Standards," in *IET ACDC*, Amsterdam, 2024.

[5] "Resource Description Framework. RDF - Semantic Web Standards," 2024. [Online]. Available: https://www.w3.org/standards/techs/rdf. [Accessed 01 04 2024].

[6] "Functional Mock-up Interface," 2024. [Online]. Available: https://fmi-standard.org/. [Accessed 01 04 2024].

[7] "A Generic VSC HVDC Model for Power System," in *EPRI HVDC and FACTS Conference*, Palo Alto, 2015.

[8] "DPsim—Advancements in Power Electronics Modelling Using Shifted Frequency Analysis and in Real-Time Simulation Capability by Parallelization," *Energies,* vol. 13, no. 15, 2020.

[9] gridDigIt, “CimPal,” [Online]. Available: https://github.com/griddigit-ci/CimPal. [Accessed 2024].

[10] S. R. Sanders, J. M. Noworolski, X. Z. Liu and G. C. Verghese, "Generalized averaging method for power conversion circuits," *IEEE Transactions on Power Electronics,* vol. 6, no. 2, pp. 251-259, 1991.

[11] K. Strunz, R. Shintaku and F. Gao, "Frequency-adaptive network modeling for integrative simulation of natural and envelope waveforms in power systems and circuits," *IEEE Transactions on Circuits and Systems I: Regular Papers,* vol. 53, no. 12, pp. 2788-2803, 2006.

[12] P. Zhang, J. Marti and H. W. Dommel, "Shifted-Frequency Analysis for EMTP Simulation of Power-System Dynamics," *IEEE Transactions on Circuits and Systems I: Regular Papers,* vol. 57, no. 9, pp. 2564-2574, 2010.

[13] M. Mirz, A dynamic phasor real-time simulation based digital twin for power systems, Aachen: RWTH Aachen University, 2020.

[14] M. Mirz, J. Dinkelbach and A. Monti, "DPsim—Advancements in Power Electronics Modelling Using Shifted Frequency Analysis and in Real-Time Simulation Capability by Parallelization," *Energies ,* vol. 13, no. 15, 2020.