HVDC Stability Simulator

Developing Guide

Aleksandra Lekić and Philippe De Rua

ELECTA and EnergyVille,

KU Leuven, Leuven, Belgium

aleksandra.lekic@kuleuven.be

December 13, 2019

Contents

1 Introduction

At present, new power electronics-based active components are being introduced in the power system at an astonishing rate. When considering High Voltage Direct Current (HVDC) systems, the most important circuit topology is the voltage source converters (VSC), and more in particular Modular Multilevel Converters (MMCs), a.o. because of its lower losses compared to traditional 2-level topologies. The MMC is characterised by its fast operation and control possibilities [1]. The integration of these devices are bringing new requirements for the system operation and the stability when considering their interactions, e.g. with each other or with the surrounding passive network components. In general, it has to be ensured that the power system operates in a stable way with the desired performance under the desired operation conditions, and that it can be stabilised in the case of the disturbances [2]. The active nature of the components makes that significant research and development efforts have historically been related to the protection and the controller design, in particular when considering multi-terminal HVDC-based systems [3, 4].

Compared to traditional AC system stability, converter controllers interact with the surrounding power system, and hence also with one another, over an extended frequency region when considering small-signal stability. The resulting negative interactions have been referred to in recent years as 'harmonic stability' or 'electromagnetic stability' [5]. Harmonic stability assessment can be undertaken in the Laplace-domain or frequency-domain using either a state-space representation or an impedance-based (admittance-based) analysis [6]. Basic electromagnetic transient (EMT) time domain simulation tools, such as PSCAD™ and EMTP-RV™, on the contrary give rise to long simulation times by checking the stability of each operating condition, which renders it less suitable for a fast system performance check. Motivated by the work initially introduced by Middlebrook [7], related to the input filter design, the so-called impedance-based stability assessment has been further investigated and implemented for the purpose of the power system stability analysis. Impedance-based system stability is a promising approach still gaining popularity especially for VSC HVDC systems, possibly connected in a multi-terminal configuration [5, 8–11].

In order to capture the influence of every passive and active component in the power system over an wide frequency range, it is necessary to model components with as much mathematical details as possible, including their frequency-dependent behaviour (e.g. lines, cables), as well as the impact of converter controls (e.g. MMCs). EMT tools such as PSCAD do offer frequency scanning routines for the passive network components (e.g. lines and cables). However, since traditional harmonic studies are not concerned with the steady-state harmonic amplification through the network, rather than the small-signal stability, these built-in scanning routines do not allow for including a small-signal representation of the converter and its control [5]. Therefore, in the framework of the Neptune project, the decision was made to start the development of a software tool dedicated towards the small-signal stability screening using an impedance-based approach in the frequency domain, including both active and passive network components. In order to obtain the interactions using highly detailed component descriptions, a new mathematical methodology for the modelling of the network using ABCD parameters is proposed and developed. The next sections describe the development of the methodology.

2 Multiport ABCD representation

Classical circuit theory applied to power systems relies on the description of the system using admittance matrices or hybrid matrices [2, 12]. Commonly, the circuit equations are solved using admittance/hybrid matrices applying Kirchhoff's laws and Ohm's law relying on the Modified Nodal Analysis (MNA) approach for the components desribed using linear models.

The admittance based representation is also gaining popularity for the assessment of harmonic stability in systems with power electronic components [13, 14].

Although the admittance representation of the system and its components has a simple definition and a physical dimension, system and component configurations exist without impedance or admittance parameters defined. In Fig. 1a, where the port voltage and current have subscript “i” for an input port and “o” for the output port, a case is depicted where the series impedance cannot be represented using impedance parameters because of the open connection at the input and at the output port. Similarly, the example from Fig. 1b shows the case of a shunt admittance. It cannot be described using admittance parameters, since the short connection between the ports would give an infinite value for the interconnection admittance. A hybrid port representation could be applied in these cases, but its usage for determining input and output impedance of the network is everything but intuitive.

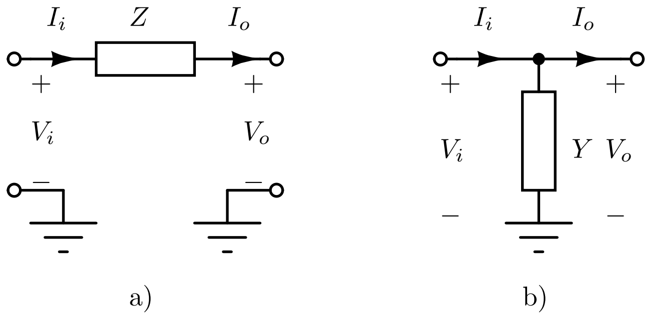


Figure 1: Examples of the circuit in which is not defined: a) impedance matrix; b) admittance matrix.

To overcome the challenge of nonexistent Z or Y parameter representation, we propose a generalised algorithm for representing the power system, and its constituting components using multiport ABCD parameters instead (Fig. 2). Multiport networks can include polyphase AC networks, multi-pole DC networks, etc. The motivation for the choice of the ABCD parameters’ system representation stems from the fact that ABCD parameters provide a direct connection between the voltages and currents at the input ports, and voltages and currents at the output ports.

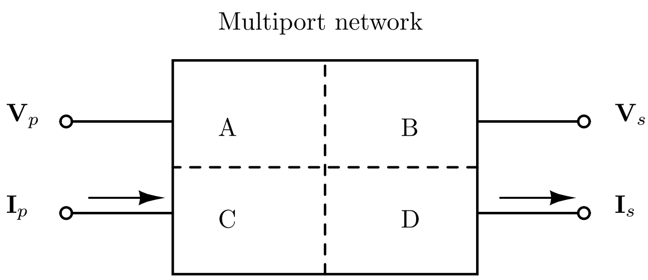


Figure 2: Polyphase power system using multiport ABCD parameters.

As explained, Admittance-based representations of networks are becoming a promising approach gaining popularity to be used in impedance-based stability assessments to investigate harmonic stability with power electronic converters [15], e.g. for Voltage Source Converter High Voltage Direct Current (VSC HVDC) systems [5, 8]. The use of ABCD parameters for such a stability assessment, however, is only recently starting to get attention in literature. Besides [5], where the method was proposed to build small network models for analyzing high-frequency interactions in the kHz-range, recently, also [16] assessed interactions, but within the bandwidth of the converter controllers (up to several 100 Hz), building on the work from [17] to create a network equivalent containing a frequency-dependent model of a single overhead line.

What has been missing so far, however, is a generalised modeling framework that allows automatically constructing an equivalent impedance, including both active and passive components, at any node in the network. Therefore, a systematically derived modeling framework using an ABCD representation for the multiport power system and its components has been developed. Each component is represented using ABCD parameters, and its corresponding model is presented in detail in this report. The model of the DC side impedance of a state-of-the-art VSC HVDC-based MMC is given and the complete system modeling is presented for a two-terminal MMC-based VSC HVDC system. The report also summarises how to use the ABCD parameters for a harmonic system stability analysis.

2.1 ABCD parameters basics

The system is represented as an interconnection of components. To simplify the calculation of the transfer functions and/or input and output impedances, each component is modeled as a multiport network as depicted in Fig. 2. The input voltages and currents are vectors denoted as $\mathbf{V}_p$ and $\mathbf{I}_p$, while the output voltages and currents are $\mathbf{V}_s$ and $\mathbf{I}_s$. Generally, a multiport network has the same number of input and output ports, and thus, the dimensions of the vectors are the same, denoted as n.

A multiport network can be represented with ABCD parameters, where each of parameters $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ and $\mathbf{D}$ represent $n \times n$ matrices and

$$\begin{bmatrix} \mathbf{V}_p \\ \mathbf{I}_p \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} \times \begin{bmatrix} \mathbf{V}_s \\ \mathbf{I}_s \end{bmatrix}. \tag{1}$$

As in electrical power systems, the components are interconnected. Two possible connections are series and parallel connections.

- Series connection of two multiport networks is depicted in Fig. 3. The ABCD multiport representation is especially desirable for this type of connection because the new parameters are determined in a simple matter as follows.

$$\begin{bmatrix} \mathbf{V}_p \\ \mathbf{I}_p \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{A}_1 & \mathbf{B}_1 \\ \mathbf{C}_1 & \mathbf{D}_1 \end{bmatrix} \times \begin{bmatrix} \mathbf{A}_2 & \mathbf{B}_2 \\ \mathbf{C}_2 & \mathbf{D}_2 \end{bmatrix}}_{\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix}} \times \begin{bmatrix} \mathbf{V}_s \\ \mathbf{I}_s \end{bmatrix} \tag{2}$$

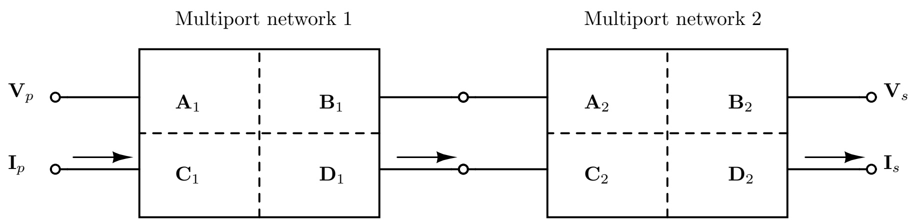


Figure 3: Series connected multiport networks.

- Parallel connection, depicted in Fig. 4, is more complex for calculation. In the case of nonzero matrices $\mathbf{B}_1$ and $\mathbf{B}_2$, the parallel connection is represented as:

$$\begin{bmatrix} \mathbf{V}_p \\ \mathbf{I}_p \end{bmatrix} = \underbrace{\begin{bmatrix} (\mathbf{B}_1^{-1} + \mathbf{B}_2^{-1})^{-1} (\mathbf{B}_1^{-1}\mathbf{A}_1 + \mathbf{B}_2^{-1}\mathbf{A}_2) & (\mathbf{B}_1^{-1} + \mathbf{B}_2^{-1})^{-1} \\ \mathbf{C}_1 + \mathbf{C}_2 + (\mathbf{D}_2 - \mathbf{D}_1)(\mathbf{B}_1 + \mathbf{B}_2)^{-1}(\mathbf{A}_1 - \mathbf{A}_2) & \mathbf{D}_1 + (\mathbf{D}_2 - \mathbf{D}_1)(\mathbf{B}_1 + \mathbf{B}_2)^{-1}\mathbf{B}_1 \end{bmatrix}}_{\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix}} \times \begin{bmatrix} \mathbf{V}_s \\ \mathbf{I}_s \end{bmatrix}. \tag{3}$$

The formula is also valid for networks whose matrices are of dimension 1×1, i.e. two port networks. If some of the matrices cannot be inverted, then the previous equation becomes:

$$\begin{bmatrix} \mathbf{V}_p \\ \mathbf{I}_p \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{A}_i & \mathbf{0} \\ \mathbf{C}_1 + \mathbf{C}_2 + (\mathbf{D}_2 - \mathbf{D}_1)\,\mathbf{B}_j^{-1}\,(\mathbf{A}_1 - \mathbf{A}_2) & \mathbf{D}_i \end{bmatrix}}_{\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix}} \times \begin{bmatrix} \mathbf{V}_s \\ \mathbf{I}_s \end{bmatrix}, \tag{4}$$

where $i, j \in \{1, 2\}$ and i denotes the invertible matrix $\mathbf{B}_i$ with $j \neq i$.

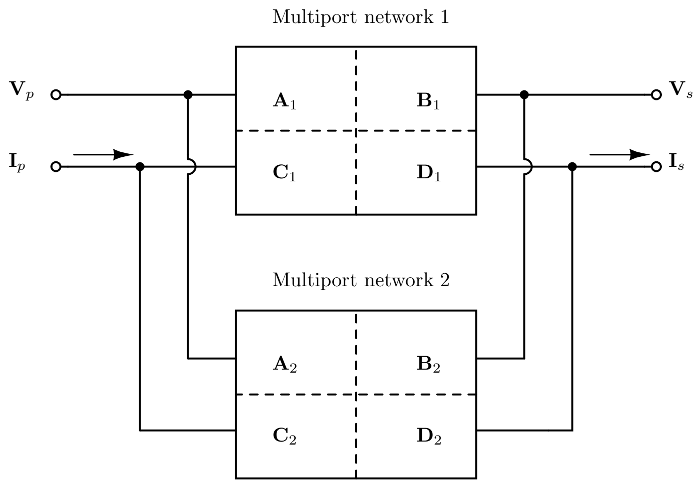


Figure 4: Parallel connected multiport networks.

It should be noted that with the term multiport network is considered both one component with one or more input and output nodes, and a subnetwork with defined input and output nodes.

2.2 Determining the input/output impedance of the network

Let us assume that every output port, represented with the voltage V_{si} and the current I_{si}, is closed with an impedance Z_{ti}. Then we can write the equation $V_{si} = Z_{ti}\, I_{si}$, or in matrix form:

$$\mathbf{V}_s = \mathbf{Z}_t \odot \mathbf{I}_s = \widetilde{\mathbf{Z}}_t \times \mathbf{I}_s, \tag{5}$$

for $\odot$ denoting hadamard product, and $\mathbf{Z}_t$ being the corresponding closing impedance column vector, and $\widetilde{\mathbf{Z}}_t = \mathrm{diag}\{\mathbf{Z}_t\}$, see Fig. 5.

The input impedance can be then estimated from:

$$\begin{bmatrix} \mathbf{V}_p \\ \mathbf{I}_p \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} \times \begin{bmatrix} \widetilde{\mathbf{Z}}_t \times \mathbf{I}_s \\ \mathbf{I}_s \end{bmatrix},$$

as

$$\mathbf{Z}_p = (\mathbf{A} \times \widetilde{\mathbf{Z}}_t + \mathbf{B}) \times (\mathbf{C} \times \widetilde{\mathbf{Z}}_t + \mathbf{D})^{-1}. \tag{6}$$

Similarly, by closing the input ports with a diagonal impedance $\widetilde{\mathbf{Z}}_t$, the impedance as seen from the output ports can be estimated as:

$$\mathbf{Z}_s = (\widetilde{\mathbf{Z}}_t \times \mathbf{C} - \mathbf{A})^{-1} \times (\widetilde{\mathbf{Z}}_t \times \mathbf{D} - \mathbf{B}). \tag{7}$$

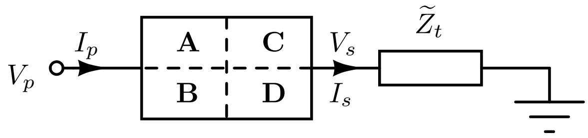


Figure 5: Closing impedance on the output side.

2.3 Determination of the combined system ABCD parameters

The ABCD parameters can be used to determine the impedance "visible" from the desired node or a component port. The power system can take forms with different number of component pins, e.g. (but not limited to) a three-phase AC system in abc parameters with components having 3 input and output pins, a positive sequence equivalent (components with 2 input and output pins); or a monopolar (represented 1 input and output pin) or bipolar DC system (2 input and 2 output pins). To determine the impedance "visible" from the desired node/nodes in the system, the partition of the system is formed recursively, containing only the nodes and the components included in the path between the desired nodes. The example of the obtained subsystem is depicted in Fig. 6, where the nodes denoted as $V_{p1,s}$, $V_{p2,s}$ and $V_{s1,s}$, $V_{s2,s}$ represent 2×2 port system, respectively.

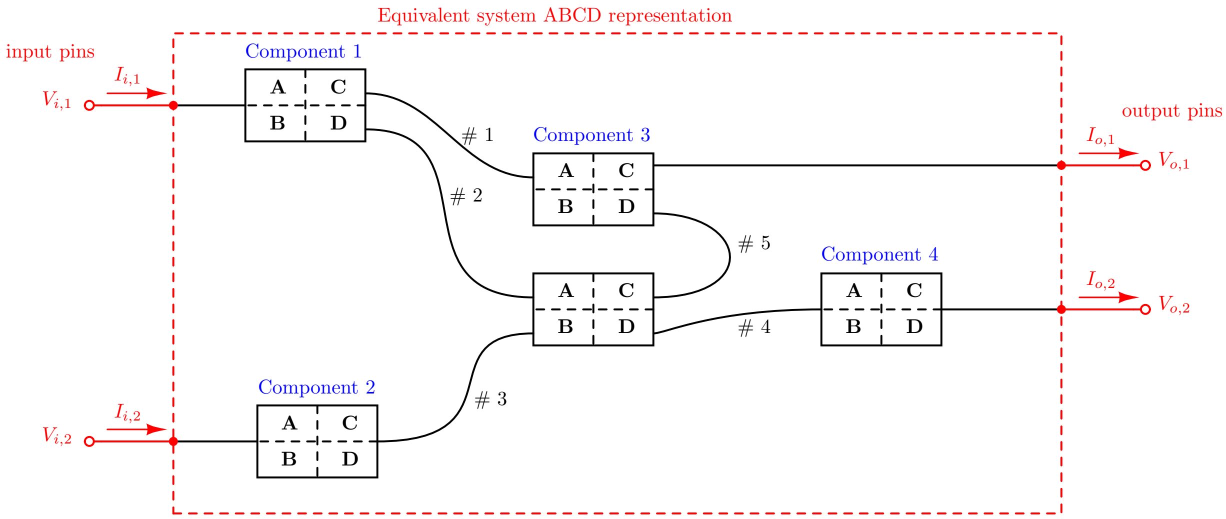


Figure 6: Model of the polyphase subsystem.

As can be seen from Fig. 6, the subsystem between input and output nodes contains m components, where each component $j \in \{1, \ldots, m\}$ is represented with p_i^j inputs and p_o^j outputs. The subsystem also contains total number of n_n nodes, of which n_o nodes are output nodes (denoted as $V_{s1,s}$ and $V_{s2,s}$ in Fig. 6) or ground nodes. Subsystem has n_c input currents/voltages (denoted as $I_{p1,s}$ and $I_{p2,s}$ in Fig. 6).

Let us assume the following naming convention. The ith component has input voltages and currents denoted as $\mathbf{V}_{pi}$ and $\mathbf{I}_{pi}$ (positive currents enter the component and exit the node), and output voltages and currents $\mathbf{V}_{si}$ and $\mathbf{I}_{si}$ (positive currents exit the component and enter the node). $\mathbf{I}_i$ are the input subsystem currents entering the subsystem and exiting the input nodes, and $\mathbf{I}_0$ are the currents through ground(s) and the output pins, and enter the output nodes.

Now, the set of $n_n + \sum_{i=1}^{m} 2p_p^i$ equations can be written:

- n_n equations for each of the nodes inside the network denoting the sum of the currents exiting the node. By convention, the currents are considered positive when their exit the node.

- $\sum_{i=1}^{m} p_p^i$ equations giving ABCD component relationships between input voltages $\mathbf{V}_{pi}$ and output voltages $\mathbf{V}_{si}$, and currents $\mathbf{I}_{si}$.
- $\sum_{i=1}^{m} p_p^i$ equations giving ABCD component relationships between input currents $\mathbf{I}_{pi}$ and output voltages $\mathbf{V}_{si}$, and currents $\mathbf{I}_{si}$.

The unknown variables are $n_v = n_n - n_o$ node voltages, n_c input currents denoted as $\mathbf{I}_i$ and $\sum_{i=1}^{m} p_p^i + p_s^i$ component currents $\mathbf{I}_{pi}$ and $\mathbf{I}_{si}$.

The complete set of equations is written in matrix form and consists of $n_n + \sum_{i=1}^{m} 2p_p^i$ equations with $n_v + n_c + \sum_{i=1}^{m}(p_p^i + p_s^i)$ variables and matrix of outputs with the size $\left(n_n + \sum_{i=1}^{m} 2p_p^i\right) \times 2n_o$. It is:

$$\mathbf{M} \times \mathbf{X} = \mathbf{N} \times \mathbf{Y} \tag{8}$$

where the matrices $\mathbf{M}$ and $\mathbf{N}$ consist of numerical and symbolic coefficients, vector $\mathbf{X} = \begin{bmatrix} V_1 & \cdots & V_{n_v} & I_{i1} & \cdots & I_{in_c} & \mathbf{I}_{p1} & \mathbf{I}_{s1} & \cdots & \mathbf{I}_{pm} & \mathbf{I}_{sm} \end{bmatrix}$ consists of the unknown variables and vector $\mathbf{Y} = \left[\mathbf{V}_{0j}, \mathbf{I}_{0j}\right]^T \Big|_{j=1}^{n_o}$ of the output and ground voltages and currents. The solution of the previous system (8) is given as reduced row echelon (or gaussian elimination) form of concatenated matrices $[\mathbf{M}, \mathbf{N}]$.

2.4 Transformation abc to dqz

In order to transform three-phase voltages and currents from the stationary abc frame to the rotating dqz frame, Park's transformation defined in Appendix A.2 is employed. For the transformation of the admittance from abc to dqz frame the following theorem can be formulated.

There has been reported work [18] that applies the transformation formula only for the symmetric systems. This formula is successfully applied for modeling of the overhead lines in the dq frame [19, 20].

Theorem 1. *Every 3×3 admittance in abc domain $\mathbf{Y}(j\omega)$ can be transformed to dq domain without loss of generality as $\mathbf{Y}_{dq}(j\omega) = \frac{1}{3}\left(\left(\mathbf{a}\,\mathbf{Y}(j(\omega+\omega_0)) + \bar{\mathbf{a}}\,\mathbf{Y}(j(\omega-\omega_0))\right)\Re\{\mathbf{a}\}^T\right)_{dq}$, where $\mathbf{a}$ is a transformation matrix defined in Appendix A.2.*

Proof. Currents and voltages in dqz frame are related to the currents and voltages in abc frame as:

$$\mathbf{i}_{dqz}(t) = \mathbf{P}_{\omega_0}(t)\,\mathbf{i}_{abc}(t),$$
$$\mathbf{v}_{dqz}(t) = \mathbf{P}_{\omega_0}(t)\,\mathbf{v}_{abc}(t),$$

and vice versa, abc can be transformed to dqz as:

$$\mathbf{i}_{abc}(t) = \mathbf{P}_{\omega_0}^{-1}(t)\,\mathbf{i}_{dqz}(t),$$
$$\mathbf{v}_{abc}(t) = \mathbf{P}_{\omega_0}^{-1}(t)\,\mathbf{v}_{dqz}(t),$$

with ω_0 being the angular frequency of the rotation frame. In the previous equations, multiplications become convolutions in the spectral domain after applying the Fourier transform, see Appendix A.3.

In the spectral domain, the relation between the currents and voltages in abc frame can be written as:

$$\mathbf{I}_{abc}(j\omega) = \mathbf{Y}(j\omega)\,\mathbf{V}_{abc}(j\omega).$$

This equation can be further used as:

$$\mathbf{I}_{dqz}(j\omega) = \frac{1}{2\pi}\,\mathbf{P}_{\omega_0}(j\omega) * (\mathbf{Y}(j\omega)\,\mathbf{V}_{abc}(j\omega)) = \frac{1}{2\pi}\,\mathbf{P}_{\omega_0}(j\omega) * \left(\mathbf{Y}(j\omega)\,\frac{1}{2\pi}\,\mathbf{P}^{-1}_{\omega_0}(j\omega) * \mathbf{V}_{dqz}(j\omega)\right). \tag{9}$$

Denoting $\mathbf{G}(j\omega) = \frac{1}{2\pi}\,\mathbf{P}^{-1}_{\omega_0}(j\omega) * \mathbf{V}_{dqz}(j\omega)$, one can obtain:

$$\begin{aligned}\mathbf{G}(j\omega) &= \frac{1}{2}\,(\mathbf{a}^T\,\delta(\omega+\omega_0) + \overline{\mathbf{a}}^T\,\delta(\omega-\omega_0) + 2\mathbf{c}^T\,\delta(\omega)) * \mathbf{V}_{dqz}(j\omega) = \\ &= \frac{1}{2}\,(\mathbf{a}^T\,\mathbf{V}_{dqz}(j(\omega+\omega_0)) + \overline{\mathbf{a}}^T\,\mathbf{V}_{dqz}(j(\omega-\omega_0)) + 2\mathbf{c}^T\,\mathbf{V}_{dqz}(\omega)).\end{aligned} \tag{10}$$

Similarly, considering that $\mathbf{H}(j\omega) = \mathbf{Y}(j\omega)\,\mathbf{G}(j\omega)$, we can write:

$$\begin{aligned}\mathbf{I}_{dqz}(j\omega) &= \frac{1}{3}\,(\mathbf{a}\,\delta(\omega+\omega_0) + \overline{\mathbf{a}}\,\delta(\omega-\omega_0) + \mathbf{c}\,\delta(\omega)) * \mathbf{H}(j\omega) = \\ &= \frac{1}{3}\,(\mathbf{a}\,\mathbf{H}(j(\omega+\omega_0)) + \overline{\mathbf{a}}\,\mathbf{H}(j(\omega-\omega_0)) + \mathbf{c}\,\mathbf{H}(j\omega)) = \\ &= \frac{1}{3}\,(\mathbf{a}\,\mathbf{Y}(j(\omega+\omega_0))\,\mathbf{G}(j(\omega+\omega_0)) + \overline{\mathbf{a}}\,\mathbf{Y}(j(\omega-\omega_0))\,\mathbf{G}(j(\omega-\omega_0)) + \mathbf{c}\,\mathbf{Y}(j\omega)\,\mathbf{G}(j\omega)) = \\ &= \frac{1}{6}\,\mathbf{a}\,\mathbf{Y}(j(\omega+\omega_0)\,(\mathbf{a}^T\,\mathbf{V}_{dqz}(j(\omega+2\omega_0)) + \overline{\mathbf{a}}^T\,\mathbf{V}_{dqz}(j\omega) + 2\mathbf{c}^T\,\mathbf{V}_{dqz}(j(\omega+\omega_0))) + \\ &\quad + \overline{\mathbf{a}}\,\mathbf{Y}(j(\omega-\omega_0)\,(\mathbf{a}^T\,\mathbf{V}_{dqz}(j\omega) + \overline{\mathbf{a}}^T\,\mathbf{V}_{dqz}(j(\omega-2\omega_0)) + 2\mathbf{c}^T\,\mathbf{V}_{dqz}(j(\omega-\omega_0))) + \\ &\quad + \mathbf{c}\,\mathbf{Y}(j\omega)\,(\mathbf{a}^T\,\mathbf{V}_{dqz}(j(\omega+\omega_0)) + \overline{\mathbf{a}}^T\,\mathbf{V}_{dqz}(j(\omega-\omega_0)) + 2\mathbf{c}^T\,\mathbf{V}_{dqz}(j\omega))).\end{aligned}$$

Since, we need to represent everything using d and q components, the corresponding expressions are zero in dq except for the zero value. These expressions are:

$$\begin{aligned}(\mathbf{a}\,\mathbf{Y}(j(\omega+\omega_0)\,2\mathbf{c}^T\,\mathbf{V}_{dqz}(j(\omega+\omega_0)))_{dq} &= \mathbf{0}_{2\times 2}, \\ (\overline{\mathbf{a}}\,\mathbf{Y}(j(\omega-\omega_0)\,2\mathbf{c}^T\,\mathbf{V}_{dqz}(j(\omega-\omega_0)))_{dq} &= \mathbf{0}_{2\times 2}, \\ (\mathbf{c}\,\mathbf{Y}(j\omega)\,(\mathbf{a}^T\,\mathbf{V}_{dqz}(j(\omega+\omega_0)) + \overline{\mathbf{a}}^T\,\mathbf{V}_{dqz}(j(\omega-\omega_0)) + 2\mathbf{c}^T\,\mathbf{V}_{dqz}(j\omega)))_{dq} &= \mathbf{0}_{2\times 2}.\end{aligned}$$

Then,

$$\begin{aligned}\mathbf{I}_{dq}(j\omega) &= \frac{1}{6}\,(\mathbf{a}\,\mathbf{Y}(j(\omega+\omega_0))\,\overline{\mathbf{a}}^T + \overline{\mathbf{a}}\,\mathbf{Y}(j(\omega-\omega_0))\,\mathbf{a}^T)_{dq}\,\mathbf{V}_{dq}(j\omega) + \\ &\quad + \frac{1}{6}\,(\mathbf{a}\,\mathbf{Y}(j(\omega+\omega_0))\,\mathbf{a}^T)_{dq}\,\mathbf{V}_{dq}(j(\omega+2\omega_0)) + \frac{1}{6}\,(\overline{\mathbf{a}}\,\mathbf{Y}(j(\omega-\omega_0))\,\overline{\mathbf{a}}^T)_{dq}\,\mathbf{V}_{dq}(j(\omega-2\omega_0)).\end{aligned}$$

Since the rotation to dqz frame has a form of the ideal low pass filter with the bandwidth $2\omega_0$ symmetrically around $\omega = 0$, $\mathbf{V}_{dq}(j(\omega-2\omega_0)) = \mathbf{V}_{dq}(j(\omega+2\omega_0)) = \mathbf{V}_{dq}(j\omega)$. Finally,

$$\mathbf{I}_{dq}(j\omega) = \frac{1}{3}\,((\mathbf{a}\,\mathbf{Y}(j(\omega+\omega_0)) + \overline{\mathbf{a}}\,\mathbf{Y}(j(\omega-\omega_0)))\,\Re\{\mathbf{a}\}^T)_{dq}\,\mathbf{V}_{dq}(j\omega) \tag{11}$$

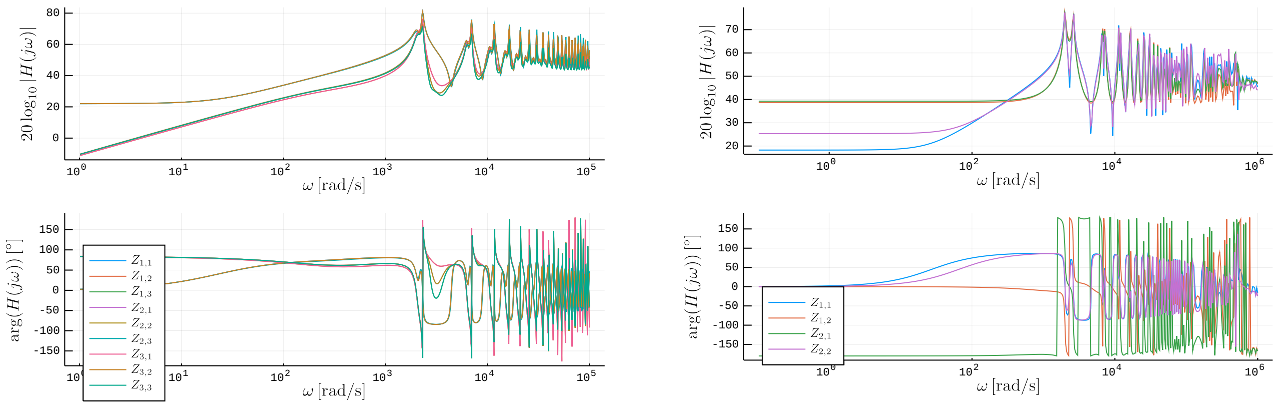
and

$$\mathbf{Y}_{dq}(j\omega) = \frac{1}{3}\,((\mathbf{a}\,\mathbf{Y}(j(\omega+\omega_0)) + \overline{\mathbf{a}}\,\mathbf{Y}(j(\omega-\omega_0)))\,\Re\{\mathbf{a}\}^T)_{dq}. \tag{12}$$

□

One example of the short circuit impedance of the three-phase overhead line is depicted in Fig. 7, where the impedance can be seen before and after the application of the transformation. The obtained diagrams correspond to [20].

The obtained formula for the admittance can be checked on a few well known examples:



(a) (b)

Figure 7: Short circuit impedance of the three phase overhead line: (a) without applied transformation; (b) with applied transformation.

- Three-phase inductor set given with the formula: $L\,\dot{\mathbf{i}}_{abc} = \mathbf{v}_{abc}$, transforms to using formula (12) to

$$\mathbf{Y}_{dq}(j\omega) = \begin{bmatrix} \frac{-j\omega}{L(\omega^2-\omega_0^2)} & \frac{\omega_0}{L(\omega^2-\omega_0^2)} \\ -\frac{\omega_0}{L(\omega^2-\omega_0^2)} & \frac{-j\omega}{L(\omega^2-\omega_0^2)} \end{bmatrix},$$

which corresponds to [21].

- Three-phase capacitor set described as $C\dot{\mathbf{v}}_{abc} = \mathbf{i}_{abc}$ gives

$$\mathbf{Y}_{dq}(j\omega) = \begin{bmatrix} j\omega C & \omega_0 C \\ -\omega_0 C & j\omega C \end{bmatrix},$$

which corresponds to [21].

In the case of ABCD parameters, the same transformation should be applied to every matrix $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ and $\mathbf{D}$. The new matrices are:

$$\begin{aligned}
\mathbf{A}_{dq}(j\omega) &= \frac{1}{3}\left((\mathbf{a}\,\mathbf{A}(j(\omega+\omega_0)) + \bar{\mathbf{a}}\,\mathbf{A}(j(\omega-\omega_0)))\,\Re\{\mathbf{a}\}^T\right)_{dq}, \\
\mathbf{B}_{dq}(j\omega) &= \frac{1}{3}\left((\mathbf{a}\,\mathbf{B}(j(\omega+\omega_0)) + \bar{\mathbf{a}}\,\mathbf{B}(j(\omega-\omega_0)))\,\Re\{\mathbf{a}\}^T\right)_{dq}, \\
\mathbf{C}_{dq}(j\omega) &= \frac{1}{3}\left((\mathbf{a}\,\mathbf{C}(j(\omega+\omega_0)) + \bar{\mathbf{a}}\,\mathbf{C}(j(\omega-\omega_0)))\,\Re\{\mathbf{a}\}^T\right)_{dq}, \\
\mathbf{D}_{dq}(j\omega) &= \frac{1}{3}\left((\mathbf{a}\,\mathbf{D}(j(\omega+\omega_0)) + \bar{\mathbf{a}}\,\mathbf{D}(j(\omega-\omega_0)))\,\Re\{\mathbf{a}\}^T\right)_{dq}.
\end{aligned}$$

2.5 Bipolar to monopolar transformation

Since power converters are modeled in this package as a three pins components, where one pin corresponds to the DC-side connection and two pins for the AC-side connection represented in the dq frame, it is necessary to represent the DC network with single-pin components. Bipolar DC components are represented by means of ABCD parameters as:

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & b_{11} & b_{12} \\ a_{21} & a_{22} & b_{21} & b_{22} \\ c_{11} & c_{12} & d_{11} & d_{12} \\ c_{21} & c_{22} & d_{21} & d_{22} \end{bmatrix}, \tag{13}$$

and

$$\begin{bmatrix} v_{p1} \\ v_{p2} \\ i_{p1} \\ i_{p2} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} \times \begin{bmatrix} v_{s1} \\ v_{s2} \\ i_{s1} \\ i_{s2} \end{bmatrix}. \tag{14}$$

For balanced bipolar DC networks connected to power converters or DC sources, the relation is valid: $v_{s1} = -v_{s2} = \frac{v_s}{2}$ and $i_{s1} = -i_{s2} = i_s$. At the input of the DC network component, it is known that $v_p = v_{p1} - v_{p2}$ and $i_{p1} = -i_{p2} = i_p$. Then, it can be estimated that:

$$\begin{bmatrix} v_p \\ i_p \end{bmatrix} = \begin{bmatrix} \frac{a_{11}+a_{22}-a_{12}-a_{21}}{2} & \frac{b_{11}+b_{22}-b_{12}-b_{21}}{2} \\ \frac{c_{11}+c_{22}-c_{12}-c_{21}}{2} & \frac{d_{11}+d_{22}-d_{12}-d_{21}}{2} \end{bmatrix} \times \begin{bmatrix} v_s \\ i_s \end{bmatrix}, \tag{15}$$

which represents an equivalent single-pin model.

2.6 Reduction of ABCD, Y and Z parameters matrix

- ABCD parameters reduction

  ABCD matrix is divided into parts: matrix part with the superscript 11 should be kept, 22 should be removed and 12 and 21 are their interconnections.

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} & \mathbf{B}_{11} & \mathbf{B}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} & \mathbf{B}_{21} & \mathbf{B}_{22} \\ \mathbf{C}_{11} & \mathbf{C}_{12} & \mathbf{D}_{11} & \mathbf{D}_{12} \\ \mathbf{C}_{21} & \mathbf{C}_{22} & \mathbf{D}_{21} & \mathbf{D}_{22} \end{bmatrix}$$

  The new, reduced matrix is obtained by applying the formula:

$$\begin{aligned} \widetilde{\mathbf{A}} &= \mathbf{A}_{11} - (\mathbf{A}_{12}\mathbf{Z}_s + \mathbf{B}_{12})\,\mathbf{E}\,(\mathbf{Z}_p\,\mathbf{C}_{21} + \mathbf{A}_{21}), \\ \widetilde{\mathbf{B}} &= \mathbf{B}_{11} - (\mathbf{A}_{12}\mathbf{Z}_s + \mathbf{B}_{12})\,\mathbf{E}\,(\mathbf{Z}_p\,\mathbf{D}_{21} + \mathbf{B}_{21}), \\ \widetilde{\mathbf{C}} &= \mathbf{C}_{11} - (\mathbf{C}_{12}\mathbf{Z}_s + \mathbf{D}_{12})\,\mathbf{E}\,(\mathbf{Z}_p\,\mathbf{C}_{21} + \mathbf{A}_{21}), \\ \widetilde{\mathbf{D}} &= \mathbf{D}_{11} - (\mathbf{C}_{12}\mathbf{Z}_s + \mathbf{D}_{12})\,\mathbf{E}\,(\mathbf{Z}_p\,\mathbf{D}_{21} + \mathbf{B}_{21}), \\ \mathbf{E} &= ((\mathbf{A}_{22}\,\mathbf{Z}_s + \mathbf{B}_{22}) + \mathbf{Z}_p\,(\mathbf{C}_{22}\,\mathbf{Z}_s + \mathbf{D}_{22}))^{-1}, \end{aligned}$$

  for $\mathbf{Z}_p$ and $\mathbf{Z}_s$ being diagonal quadratic matrices representing closing loads (impedances) of the pins that should be reduced from the input and output side, respectively.

  Newly defined ABCD parameters are:

$$\begin{bmatrix} \widetilde{\mathbf{A}} & \widetilde{\mathbf{B}} \\ \widetilde{\mathbf{C}} & \widetilde{\mathbf{D}} \end{bmatrix}.$$

- Y and Z matrix

  Here we apply the so-called Kron reduction. It will be formulated only for the Y parameters, since it can be applied exactly the same for Z parameters.

  Y matrix is divided into four parts: matrix part with the superscript 11 should be kept, 22 should be removed and 12 and 21 are their interconnections.

$$\mathbf{Y} = \begin{bmatrix} \mathbf{Y}_{11} & \mathbf{Y}_{12} \\ \mathbf{Y}_{21} & \mathbf{Y}_{22} \end{bmatrix}$$

  The new Y parameters are then:

$$\widetilde{\mathbf{Y}} = \mathbf{Y}_{11} - \mathbf{Y}_{12}\,\mathbf{Y}_{22}^{-1}\,\mathbf{Y}_{21}.$$

2.7 Conclusion

Although ABCD parameters can only be properly defined when the number of input and output nodes (voltages and currents) is the same, this multiport representation has multiple advantages:

- The input and output multiport impedance can be found directly.
- There is the unique representation of the each multiport network using ABCD parameters. For instance, ABCD parameters are defined even in cases where the admittance matrix does not exist, e.g., in case of a infinite shunt admittance.
- ABCD parameters operate with voltages and currents and thus, the values inside ABCD matrix have clear physical dimension and "meaning". This cannot be said for H (hybrid) parameters, which is usually used for RF and microelectronics simulations.
- There is a unique relationship between multiport Z, Y, H, S and ABCD multiport parameters [22, 23].

3 Multiport Y representation

Similarly, like with ABCD parameters, the system can be described using Y parameters. Y parameters of the components can be estimated using a conversion formula (20). However, the system of equations is solved in a different manner. For an N-node network, the relationship between currents injected at the nodes $\{I_1, ...I_N\}$ and voltages on the nodes $\{V_1, ...V_N\}$ is given by

$$\begin{bmatrix} Y_{1,1} & Y_{1,2} & \cdots & Y_{1,k} & \cdots & Y_{1,N} \\ Y_{2,1} & Y_{2,2} & \cdots & Y_{2,k} & \cdots & Y_{2,N} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ Y_{k,1} & Y_{k,2} & \cdots & Y_{k,k} & \vdots & Y_{k,N} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ Y_{N,1} & Y_{N,2} & \cdots & Y_{N,k} & \cdots & Y_{N,N} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ \vdots \\ V_k \\ \vdots \\ V_N \end{bmatrix} = \begin{bmatrix} I_1 \\ I_2 \\ \vdots \\ I_k \\ \vdots \\ I_N \end{bmatrix} \tag{16}$$

It should be noted that for the construction of the voltage source, the matrix $\mathbf{B}$ is set to contain very small impedances if it was a zero matrix before. The input impedance can be calculated using the following steps. First, the currents at the nodes k_i, $i \in \{1, \ldots, N\}$ are equaled with 1 and the other currents set to zero.

$$\begin{bmatrix} Y_{1,1} & Y_{1,2} & \cdots & Y_{1,k_i} & \cdots & Y_{1,N} \\ Y_{2,1} & Y_{2,2} & \cdots & Y_{2,k_i} & \cdots & Y_{2,N} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ Y_{k_i,1} & Y_{k_i,2} & \cdots & Y_{k_i,k_i} & \vdots & Y_{k_i,N} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ Y_{N,1} & Y_{N,2} & \cdots & Y_{N,k_i} & \cdots & Y_{N,N} \end{bmatrix} \begin{bmatrix} V_k \\ V_2 \\ \vdots \\ V_1 \\ \vdots \\ V_N \end{bmatrix} = \mathbf{M} \begin{bmatrix} I_1 \\ I_2 \\ \vdots \\ I_k \\ \vdots \\ I_N \end{bmatrix}, \tag{17}$$

where matrix $\mathbf{M}$ has all zero values except to diagonal entries at the positions corresponding to the nodes k_i, which are 1. Then the solution of the previous system is estimated by solving $\mathbf{Y}^{-1}\,\mathbf{M}$ and the values of the corresponding node voltages k_i are chosen. These values represent impedances in the form of the Z parameters matrix. The matrix between input and the output can then be determined using Kron elimination [24].

3.0.1 Transformation between Y and ABCD parameters

$$\mathbf{Y} = \begin{bmatrix} \mathbf{Y}_{ee} & \mathbf{Y}_{ei} \\ \mathbf{Y}_{ie} & \mathbf{Y}_{ii} \end{bmatrix} \tag{18}$$

$$\mathbf{Y} = \begin{bmatrix} \mathbf{D}\mathbf{B}^{-1} & \mathbf{C} - \mathbf{D}\mathbf{B}^{-1}\mathbf{A} \\ -\mathbf{B}^{-1} & \mathbf{B}^{-1}\mathbf{A} \end{bmatrix} \tag{19}$$

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} = \begin{bmatrix} -\mathbf{Y}_{ie}^{-1}\,\mathbf{Y}_{ii} & -\mathbf{Y}_{ie}^{-1} \\ \mathbf{Y}_{ei} - \mathbf{Y}_{ee}\,\mathbf{Y}_{ie}^{-1}\,\mathbf{Y}_{ii} & -\mathbf{Y}_{ie}^{-1}\,Y_{ee} \end{bmatrix} \tag{20}$$

4 Implementation of the components

The main components of the hybrid AC and DC power system are DC and AC grids equivalents (represented as sources), impedances, transformers, transmission lines and cables, breakers, FACTS, shunt components and converters (MMC converters, two level converters, etc.). Each of the components will be presented as multiport network using ABCD parameters as follows.

4.1 Impedance

An impedance can be defined between n input ports (nodes) and n output ports (nodes). An impedance is represented as an $n \times n$ matrix $\mathbf{Z}$:

$$\mathbf{Z} = \begin{bmatrix} Z_{11} & Z_{12} & \cdots & Z_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ Z_{n1} & Z_{n2} & \cdots & Z_{nn} \end{bmatrix}$$

An example of an impedance with two input ports and two output ports is given in Fig. 8.

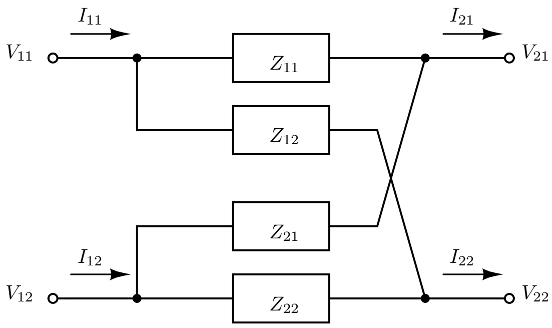


Figure 8: Model of the 2 input ports/2 output ports impedance.

To represent impedances as multiport components with ABCD parameters, it is required to solve the following equation constructed using Modified Nodal Analysis (MNA) approach [25].

$$\underbrace{\left[\begin{array}{c|c} \operatorname*{diag}_{i}\left\{ \sum_{j, Z_{ij} \neq 0} \frac{1}{Z_{ij}} \right\}_{n \times n} & \operatorname{diag}\{-1\}_{n \times n} \\ \hline \mathbf{N}_{1, n \times n} & \mathbf{0}_{n \times n} \end{array}\right]}_{\mathbf{M}_1} \times \begin{bmatrix} \mathbf{V}_p \\ \mathbf{I}_p \end{bmatrix} = \underbrace{\left[\begin{array}{c|c} \mathbf{N}_{2, n \times n} & \mathbf{0}_{n \times n} \\ \hline -\operatorname*{diag}_{i}\left\{ \sum_{j, Z_{ji} \neq 0} \frac{1}{Z_{ji}} \right\}_{n \times n} & \operatorname{diag}\{-1\}_{n \times n} \end{array}\right]}_{\mathbf{M}_2} \times \begin{bmatrix} \mathbf{V}_s \\ \mathbf{I}_s \end{bmatrix}, \tag{21}$$

where matrices $\mathbf{N}_1$ and $\mathbf{N}_2$ consist of n rows with n columns with entries at the position (i, j) equal to $-\frac{1}{Z_{ji}}$ and $\frac{1}{Z_{ij}}$, for $Z_{ij} \neq 0$ and $Z_{ji} \neq 0$ (where i represents row and j column in impedance matrix), respectively. The solution of the previous system is given as $\mathbf{M}_1^{-1}\mathbf{M}_2$ if $\mathbf{M}_1$ is invertible matrix, or by determining LU decomposition and reduced row echelon form if it is not invertible.

For example, for the circuit depicted in Fig. 8, the equations would be:

$$\left[\begin{array}{cc|cc} \frac{1}{Z_{11}} + \frac{1}{Z_{12}} & 0 & -1 & 0 \\ 0 & \frac{1}{Z_{21}} + \frac{1}{Z_{22}} & 0 & -1 \\ \hline -\frac{1}{Z_{11}} & -\frac{1}{Z_{21}} & 0 & 0 \\ -\frac{1}{Z_{12}} & -\frac{1}{Z_{22}} & 0 & 0 \end{array}\right] \times \begin{bmatrix} V_{11} \\ V_{12} \\ I_{11} \\ I_{12} \end{bmatrix} = \left[\begin{array}{cc|cc} \frac{1}{Z_{11}} & \frac{1}{Z_{12}} & 0 & 0 \\ \frac{1}{Z_{21}} & \frac{1}{Z_{22}} & 0 & 0 \\ \hline -\left(\frac{1}{Z_{11}} + \frac{1}{Z_{21}}\right) & 0 & -1 & 0 \\ 0 & -\left(\frac{1}{Z_{12}} + \frac{1}{Z_{22}}\right) & 0 & -1 \end{array}\right] \times \begin{bmatrix} V_{21} \\ V_{22} \\ I_{21} \\ I_{22} \end{bmatrix}.$$

4.2 Transformer

A transformer is modeled as in Fig. 9. The parameters of the transformer can be defined explicitly or determined from on-site test data as described in [26]. On-site test data are given in the form of open and short-circuit values of the primary side voltage V_1 and current I_1 and secondary side voltage V_2 and current I_2, along with the core power losses $P_{1,core}$ and winding power losses $P_{1,winding}$. The open and short circuit test should be performed on the secondary side.

Then the parameters from Fig. 9 can be estimated as:

$$\begin{aligned}
R_{ps} &= \frac{P_1^{short}}{(I_1^{short})^2}, & L_{ps} &= \frac{Q_1^{short}}{\omega (I_1^{short})^2},\\
R_m &= \frac{(V_1^{open})^2}{P_1^{open}}, & L_m &= \frac{(V_1^{open})^2}{\omega Q_1^{open}},\\
n &= \frac{V_1^{open}}{V_2^{open}}, & &\\
R_p &= \frac{R_{ps}}{2}, & L_p &= \frac{L_{ps}}{2},\\
R_s &= \frac{R_{ps}}{2n^2}, & L_s &= \frac{L_{ps}}{2n^2},
\end{aligned} \tag{22}$$

knowing that $Q_1^{o,s} = \sqrt{(V_1^{o,s} I_1^{o,s})^2 - P_1^{o,s2}}$.

ABCD multiport parameters are then estimated as [5, 17]:

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} = \mathbf{Y}_{turn} \times \left(\mathbf{Z}^p_{winding} \times \mathbf{Y}_{iron} \times \mathbf{N}_{tr} \times \mathbf{Z}^s_{winding} \parallel \mathbf{Z}_{stray}\right) \times \mathbf{Y}_{turn}, \tag{23}$$

where $\mathbf{Y}_{turn} = \begin{bmatrix} 1 & 0 \\ sC_t & 1 \end{bmatrix}$, $\mathbf{Z}^p_{winding} = \begin{bmatrix} 1 & sL_p + R_p \\ 0 & 1 \end{bmatrix}$, $\mathbf{Y}_{iron} = \begin{bmatrix} 1 & 0 \\ \frac{1}{sL_m} + \frac{1}{R_m} & 1 \end{bmatrix}$, $\mathbf{Z}^s_{winding} = \begin{bmatrix} 1 & sL_s + R_s \\ 0 & 1 \end{bmatrix}$, $Z_{stray} = \begin{bmatrix} 1 & \frac{1}{sC_{stray}} \\ 0 & 1 \end{bmatrix}$ and $\mathbf{N}_{tr} = \begin{bmatrix} n & 0 \\ 0 & \frac{1}{n} \end{bmatrix}$, with n the turn ratio.

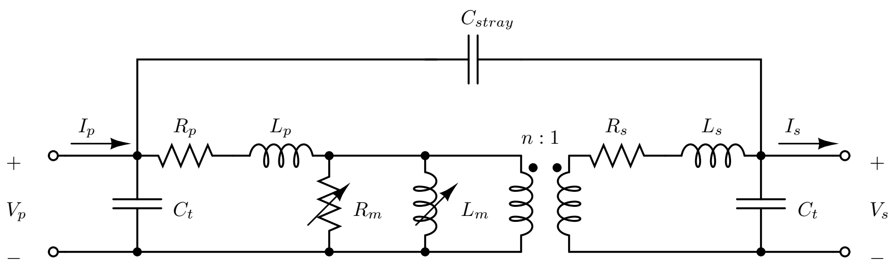


Figure 9: Model of the transformer.

Three-phase transformers can be either in the YY and ΔY configuration, where each of the three single-phase transformers is represented by its equivalent from Fig. 9.

- The YY configuration is derived from the equation (23), such that

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} = \begin{bmatrix} \text{diag}\{A\}_{3\times 3} & \text{diag}\{B\}_{3\times 3} \\ \text{diag}\{C\}_{3\times 3} & \text{diag}\{D\}_{3\times 3} \end{bmatrix}.$$

- The ΔY configuration is more complex and it is modeled using following equations. The inner primary and secondary stages of the transformer (i.e. all the components except the parasitic capacitances and the load impedance) are given by:

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}_{inner} = \mathbf{Z}^p_{winding} \times \mathbf{Y}_{iron} \times \mathbf{N}_{tr} = \begin{bmatrix} n + n\,(sL_p + R_p)\left(\frac{1}{sL_m} + \frac{1}{R_m}\right) & \frac{sL_p+R_p}{n} \\ n\left(\frac{1}{L_m} + \frac{1}{R_m}\right) & \frac{1}{n} \end{bmatrix}. \quad (24)$$

The ΔY configuration transforms voltages from the delta side $v_p^{a,b,c}$ to the wye side voltages $v_s^{a,b,c}$ as $v_p^{a,b,c} = \sqrt{3}\, v_s^{a,b,c}$, while the currents are related as:

$$\mathbf{i}_p^{a,b,c} = \begin{bmatrix} \frac{1}{\sqrt{3}} & 0 & -\frac{1}{\sqrt{3}} \\ -\frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & 0 \\ 0 & -\frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \end{bmatrix} \times \mathbf{i}_s^{a,b,c}.$$

Using the previous voltage/current relations and ABCD representation of the inner transfer function in Eq. (24), the inner impedance can be obtained as:

$$\mathbf{Z}_{inner} = \left[\begin{array}{c|c} \mathrm{diag}\{A_{inner}\sqrt{3}\}_{3\times 3} & \mathrm{diag}\{\frac{B_{inner}}{\sqrt{3}}\}_{3\times 3} \\ \hline \mathrm{diag}\{C_{inner}\sqrt{3}\}_{3\times 3} & \begin{array}{ccc} \frac{D_{inner}}{\sqrt{3}} & 0 & -\frac{D_{inner}}{\sqrt{3}} \\ -\frac{D_{inner}}{\sqrt{3}} & \frac{D_{inner}}{\sqrt{3}} & 0 \\ 0 & -\frac{D_{inner}}{\sqrt{3}} & \frac{D_{inner}}{\sqrt{3}} \end{array} \end{array}\right]. \quad (25)$$

The transformer is eventually represented using ABCD parameters as:

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} = (\mathbf{Y}_{turn} \times (\mathbf{Z}_{inner} \parallel \mathbf{Z}_{stray}) \times \mathbf{Y}_{turn}). \quad (26)$$

4.3 Autotransformer

This type of model may be expanded to a multi-winding transformer, e.g., a three-winding transformer. As an example, the positive and negative, and zero sequence impedance of an autotransformer with YNa0(d) configuration is shown in Fig 10. In Fig. 10, H, X and Y refer to the high-voltage, low-voltage and tertiary voltage side respectively. The per unit leakage impedances may be obtained from the per unit leakage impedances Z_{HX}, Z_{HY} and Z_{XY}, as obtained using the short-circuit test, and impedance to ground Z_g as [27]:

$$\begin{bmatrix} Z_{\mathrm{X}} \\ Z_{\mathrm{Y}} \\ Z_{\mathrm{Z}} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 & 1 & -1 \\ 1 & -1 & 1 \\ -1 & 1 & 1 \end{bmatrix} \begin{bmatrix} Z_{\mathrm{HX}} \\ Z_{\mathrm{HY}} \\ Z_{\mathrm{XY}} \end{bmatrix}, \text{ and} \quad (27)$$

$$\begin{bmatrix} Z_{\mathrm{X0}} \\ Z_{\mathrm{Y0}} \\ Z_{\mathrm{n0}} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 & 1 & -1 & \frac{n-1}{n} \\ 1 & -1 & 1 & -\frac{n-1}{n^2} \\ -1 & 1 & 1 & \frac{1}{n} \end{bmatrix} \begin{bmatrix} Z_{\mathrm{HX}} \\ Z_{\mathrm{HY}} \\ Z_{\mathrm{XY}} \\ 6Z_{\mathrm{g}} \end{bmatrix}, \quad (28)$$

where n is the winding transformation ratio. The phase (or physical) domain model may be derived from these sequence impedances using the Fortescue transform.

4.4 Transmission line

The ABCD model parameters of a transmission line can be than estimated as in [28, 29]

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} = \begin{bmatrix} \cosh(\Gamma l) & \mathbf{Y}_c^{-1}\sinh(\Gamma l) \\ \mathbf{Y}_c \sinh(\Gamma l) & \cosh(\Gamma l) \end{bmatrix} \quad (29)$$

where $\Gamma = \sqrt{\mathbf{ZY}}$ and $\mathbf{Y}_c = \mathbf{Z}^{-1}\,\Gamma$, and l standing for the line or cable length. The used formula is frequency dependent phase domain model.

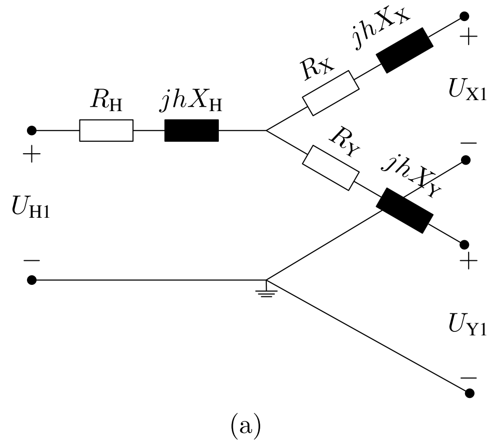


Figure 10: Three-winding transformer model for autotransformer with YNa0(d) configuration in positive and negative sequence (a) and zero sequence (b).

4.4.1 Overhead line

Focusing on the PSCAD realization of the transmission line, see Fig. 11, five possible realizations are defined as:

1. flat (horizontal, which presents flat configuration without ground wires),
2. vertical,
3. delta (for at least three phases),
4. offset (for at least three phases),
5. concentric (for at least three phases).

Besides, the conductor positions can be added manually as absolute (x, y) positions.

Thus, the simulator enables the creation of overhead lines with the **conductors** having the following properties [26]:

n_b number of bundles (or a number of phases)

n_{sb} number of subconductors per bundle

y_{bc} height of the lowest bundle above ground

Δy_{bc} vertical offset between bundles

Δx_{bc} horizontal offset between the lowest bundles

$\Delta \tilde{x}_{bc}$ horizontal offset in the case of concentric and offset organization

d_{sb} distance between closest subconductors with equidistant concentric organization (symmetric)

d_{sag} maximal sag offset

r_c radius of the conductor

R_{dc} DC resistance of the conductor

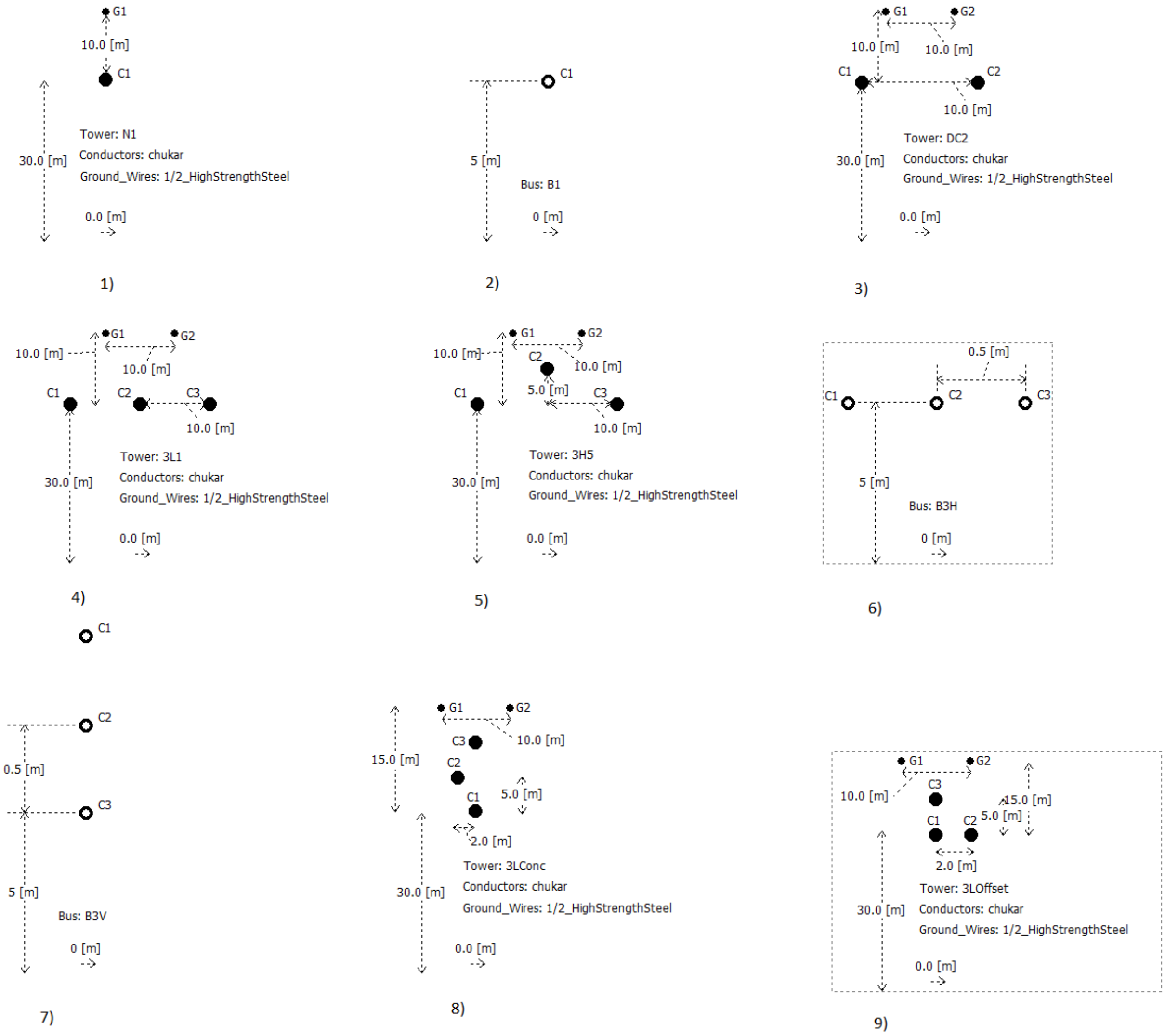


Figure 11: PSCAD overhead line orzanization: 1) single conductor with groundwire; 2) single conductor; 3) 2 conductors flat; 4) 3 conductors flat; 5) 3 conductors delta; 6) 3 conductors horizontal; 7) 3 conductors vertical; 8) 3 conductors concentric; 9) 3 conductors offset.

g_c shunt conductance of the conductor

$\mu_{r,c}$ relative permeability of the conductor

positions added manually

organization can be flat, vertical, concentric, delta and offset

Ground wires have the following properties:

n_g number of ground wires

Δx_g relative horizontal distance between ground wires

Δy_g relative vertical between ground wires and the lowest conductors

r_g radius of the ground wire

$d_{g,sag}$ ground wire sag

$R_{g,dc}$ DC resistance of the ground wires

$\mu_{r,g}$ relative permeability of the ground wire

The transmission line model is constructed using the procedure from [26,30]. The overhead transmission line consists of n_b including sub-conductors, stranding, etc. and n_g ground wires.

Each line/conductor positioned as x_c relatively starting from the central tower position and y_c vertically, measured from ground, with the sag at the midpoint between towers d_{sag}, see Fig. 12a. Thus, the modified vertical position is used in calculations as $\hat{y}_c = y_c - \frac{2}{3}\, d_{sag}$. Conductor is formed using n_{sb} sub-conductors grouped in the bundle, where all sub-conductors are grouped using symmetrical equidistant pattern with the distance between the two nearest sub-conductors being d_{bc}, or a bundle spacing. Using conductor position, the position of the each sub-conductor can be estimated. Knowing the angle between two sub-conductors on the circle and its radius

$$\varphi = \frac{360^\circ}{n_{sb}},$$
$$r = \frac{d_{sb}}{2\sin(\varphi/2)}, \tag{30}$$

the position can be estimated starting from the angle $\varphi_s = \frac{\pi}{2}$ if number of sub-conductors is odd, or from $\varphi_s = \frac{\pi+\varphi}{2}$ for the even number of sub-conductors, as follows:

$$x_{bc} = x_c + r\cos(\varphi_s + k\,\varphi),$$
$$y_{bc} = y_c + r\sin(\varphi_s + k\,\varphi) - \frac{2}{3}\, d_{sag}, \tag{31}$$

for $k \in \{1, 2, ..., n_{bc}\}$. If number of sub-conductors is one than its position is $(x_c, \hat{y}_c)$. Each conductor is characterized with the relative permeability of the material μ_r, conductor dc resistance R_{dc} and the radius r_i.

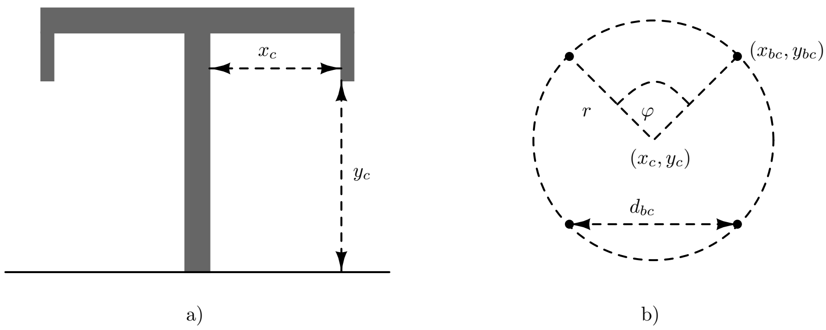


Figure 12: Overhead line modelling: a) tower and relative conductor positions; b) sub-conductor bundle.

Ground wires are modeled similarly, represented with their relative position (x_g, y_g), radius r_g, dc resistance R_{gdc} and the relative permeability of the material μ_r.

Earth parameters are given with permeability μ_e, permittivity ϵ_e and conductivity.

In order to represent transmission line using ABCD parameters, it is necessary to calculate series impedance and shunt admittance matrices [26]. Both matrices are of the size $n \times n$, where $n = \sum_{i=1}^{n_c} n_{bc}^i + n_g$. Impedance matrix has the following form:

$$\mathbf{Z} = \mathrm{diag}(Z_i) + \begin{bmatrix} Z_{0,11} & \cdots & Z_{0,1n} \\ \vdots & \ddots & \vdots \\ Z_{0,n1} & \cdots & Z_{0,nn} \end{bmatrix} \tag{32}$$

where $Z_i = \frac{m\rho_i}{2\pi r_i}\coth(0.733 m r_i) + \frac{0.3179\rho_i}{\pi r_i^2}$ for ith conductor/sub-conductor/ground wire and r_i is its radius, resistivity $\rho_i = R^i_{dc}\pi r_i^2$ and $m = \sqrt{j\omega\frac{\mu_0\mu_{r,i}}{\rho_i}}$; components $Z_{0,ij} = \frac{j\omega\mu_0}{2\pi}\log\left(\frac{\hat{D}_{ij}}{d_{ij}}\right)$ for

$$
\begin{aligned}
d_{ij} &= \begin{cases} \sqrt{(x_i-x_j)^2+(y_i-y_j)^2}, & i \neq j, \\ r_i, & i = j, \end{cases} \\
D_{ij} &= \begin{cases} \sqrt{(x_i-x_j)^2+(y_i+y_j)^2}, & i \neq j, \\ 2y_i, & i = j, \end{cases} \\
\hat{D}_{ij} &= \sqrt{(y_i+y_j+2d_e)^2+(x_i-x_j)^2}, \\
d_e &= \sqrt{\frac{1}{j\omega\,\mu_e(\sigma_e+j\omega\,\epsilon_e)}}.
\end{aligned}
\tag{33}
$$

Shunt admittance is a matrix formed as

$$\mathbf{Y} = s\,\mathbf{P}^{-1} + \mathbf{G} \tag{34}$$

from matrix $\mathbf{P}$ with the components $\mathbf{P}_{ij} = \frac{1}{2\pi\epsilon_0}\log\left(\frac{D_{ij}}{d_{ij}}\right)$ and $\mathbf{G} = \mathrm{diag}\{g_c\}$.

4.4.2 Cable

Cable groups are implemented focusing on the available configurations of the cables in PSCAD, which enables cable organization inside the pipe, so called pipe-type cables, and cables placed underground. Cables are usually coaxial with up to 4 layers of both conductors and insulators.

Cables can be insulated or pipe-type coaxial cables. At the moment it is only implemented a group of coaxial cables. Group consists of n cables, each one have maximum three conducting layers and three insulation layers, as can be seen from Fig. 13. Conducting layers of the cable denoted as core, sheath and an armor. Between conducting layers are insulators, except for the last conductor where the insulator is not a necessity, but it is common. For each conductor are given parameters: r_i^c and r_o^c as conductor inner and outer radius in meters, conductor relative permeability μ_r^c and conductor resistivity ρ_c ([Ωm] is its unit). Insulator is described using parameters: r_i^i and r_o^i as insulator inner and outer radius in meters, ϵ^i insulator relative permittivity and μ_r^i insulator relative permeability.

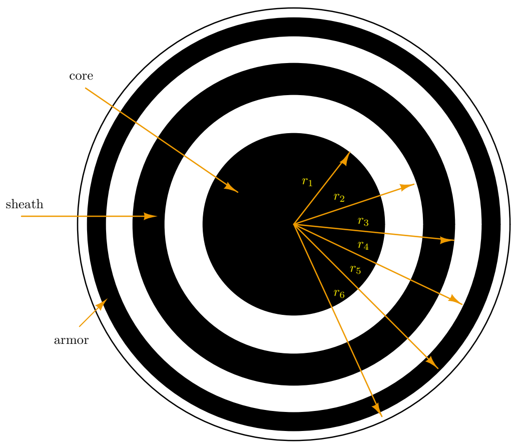


Figure 13: Coaxial cable.

Additionally, the configuration parameters can be modified by adding two semiconducting layers in the insulator 1, and implementing the sheath consisting of the wire screen and outer sheath layer. Then the procedure described in [17] is applied.

- Conductor surface impedance

  Hollow conductor surface impedance is given by:

$$\begin{aligned}
Z_{aa} &= \frac{\rho_c m}{2\pi r_i^c} \coth(m(r_o^c - r_i^c)) + \frac{\rho^c}{2\pi r_i^c\,(r_i^c + r_o^c)} \quad \left[\frac{\Omega}{\mathrm{m}}\right] \quad \text{for inner surface,}\\
Z_{bb} &= \frac{\rho^c m}{2\pi r_o^c} \coth(m(r_o^c - r_i^c)) + \frac{\rho^c}{2\pi r_o^c\,(r_i^c + r_o^c)} \quad \left[\frac{\Omega}{\mathrm{m}}\right] \quad \text{for outer surface,}\\
Z_{ab} &= \frac{\rho^c m}{\pi(r_i^c + r_o^c)} \operatorname{csch}(m(r_o^c - r_i^c)) \quad \left[\frac{\Omega}{\mathrm{m}}\right],
\end{aligned} \tag{35}$$

  where $m = \sqrt{j\omega\mu_r^c}$ For non-hollow conductor, the outer surface impedance is

$$Z_{bb} = \frac{\rho^c m}{2\pi r_o^c} \coth(0.733 m r_o^c) + \frac{0.3179\rho^c}{\pi r_o^{c2}} \quad \left[\frac{\Omega}{\mathrm{m}}\right]. \tag{36}$$

- Insulator layer between two conductors has impedance

$$Z_i = \frac{j\omega\mu_0\mu_r^i}{2\pi} \log\left(\frac{r_o^i}{r_i^i}\right). \tag{37}$$

- Earth return impedance of the cable and mutual between cables is

$$Z_g = \frac{j\omega\,\mu_g}{2\pi}\left(-\log\left(\frac{\gamma m D}{2}\right) + \frac{1}{2} - \frac{2}{3}mH\right), \tag{38}$$

  for

$$\begin{aligned}
D &= \begin{cases} \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} & \text{for cables } i \neq j,\\ r_i & \text{radius of the cable } i,\end{cases}\\
H &= \begin{cases} y_i + y_j & \text{for cables } i \neq j,\\ 2y_i & \text{for the cable } i,\end{cases}
\end{aligned} \tag{39}$$

  and $\gamma \approx 0.5772156649$ being Euler's constant.

According to [31,32], one cable is represented with series impedance $\mathbf{Z}_{ii}$ matrix. Each matrix $\mathbf{Z}_{ii}$ has the size $n_c \times n_c$ and its components are

$$\begin{aligned}
\mathbf{Z}_{ii}(j,j) &= Z_{bb}^j + Z_i^j + Z_{aa}^{j+1},\\
\mathbf{Z}_{ii}(j,j+1) = Z_{ii}(j+1,j) &= -Z_{ab}^{j+1},\\
\mathbf{Z}_{ii}(n_c,n_c) &= Z_{bb}^{n_c} + Z_i^{n_c} + Z_g^{ii},
\end{aligned} \tag{40}$$

and otherwise 0 for jth conductor and insulator and $j \in \{1,\ldots,n_c\}$.

Interconnections with the cables are given by matrix $\mathbf{Z}_{ij}$ having all components equal Z_g^{ij}.

Shunt admittance matrix can be estimated as $\mathbf{Y} = s\mathbf{P}^{-1}$ and matrix Pm which has the form

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_{11} & \mathbf{P}_{12} & \cdots & \mathbf{P}_{1n}\\ \vdots & \ddots & & \vdots\\ \mathbf{P}_{n1} & \mathbf{P}_{n2} & \cdots & \mathbf{P}_{nn}\end{bmatrix}.$$

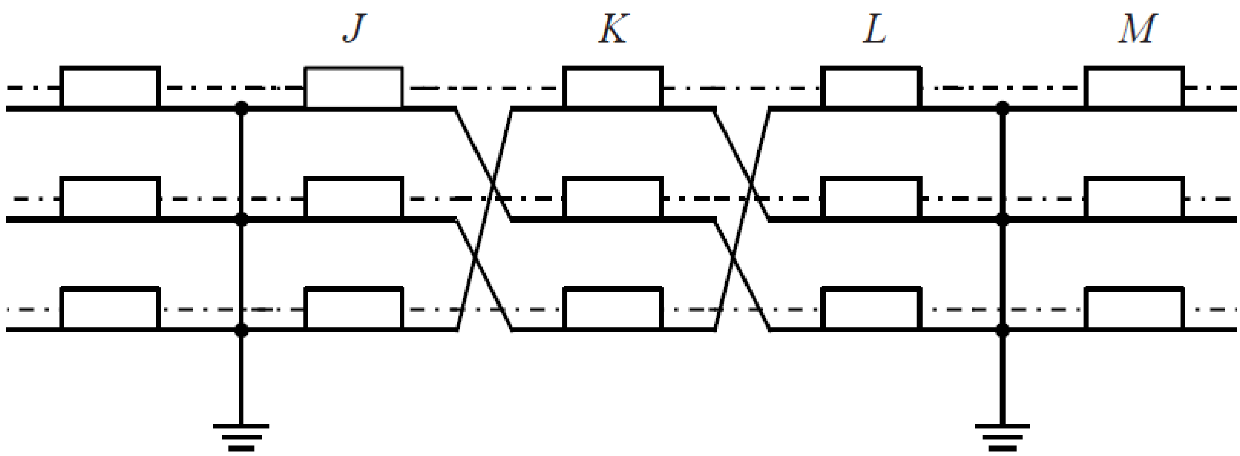


Figure 14: Cross-bonded cable.

Matrices $\mathbf{P}_{ii}$ have components

$$\mathbf{P}_{ii} = \begin{bmatrix} P_c + P_s + P_a & P_s + P_a & P_a \\ P_s + P_a & P_s + P_a & P_a \\ P_a & P_a & P_a \end{bmatrix} + \begin{bmatrix} P_{ii} & P_{ii} & P_{ii} \\ P_{ii} & P_{ii} & P_{ii} \\ P_{ii} & P_{ii} & P_{ii} \end{bmatrix}, \tag{41}$$

where $P_{c,s,a}$ belong to core, shield and armor insulators and have values: $P = \frac{\log(r_o/r_i)}{2\pi\epsilon}$ and $P_{ii} = \frac{\log(2h_i/r)}{2\pi\epsilon_0}$ is a earth return. Matrices $\mathbf{P}_{ij}$, for $i \neq j$,have all components equal to $P_{ij} = \frac{\log(D_2/D_1)}{2\pi\epsilon_0}$, where $D_1 = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$ and $D_2 = \sqrt{(x_i - x_j)^2 + (y_i + y_j)^2}$ [31].

As it is valid to assume that sheath and armor are grounded, it is allowed to use Kron reduction. Using Kron reduction, as proposed in [33, 34], applied to the matrices Y and Z is obtained compact shunt admittance and series impedance model. For determining ABCD parameters is used the same procedure as for the transmission line from equation (29).

4.4.3 Cross-bonded cables

Cables are cross-bonded in order to reduce sheath circulating currents. The cross-bonding is made by transposing sheaths of the cable sections. As in [17], this transposition can be made in ABCD domain.

Cross-bonded cable consists of minor sections as in Fig. 14, where the minor sections are marked with J, K and L. Minor sections are then grouped into major sections, for which all the cable layers except core are short connected to ground. Thus the ABCD parameters of the major section can be estimated using Kron elimitation.

Procedure for determining ABCD parameters of the whole cross-bonded cable is as follows. Let us assume that ABCD parameters of each major section are marked as $ABCD^r_\eta$. Thus, the equivalent cable ABCD parameters are $ABCD = \prod_{\eta=1}^{n} ABCD^r_\eta$, where n is a number of major sections.

Equivalent ABCD parameters of one major section can be estimated as follows:

- Determine ABCD parameters of the each minor section inside the major: $ABCD_{\eta,i}$, for $i \in \{1, m\}$ and m is the number of the minor sections inside η major section
- Reorganize ABCD matrix for each minor section as: $M_{\eta,i} = \mathbf{R}\, ABCD_{\eta,i}\, \mathbf{R}^{-1}$. Matrix $\mathbf{R}$ is the transposition matrix that sorts the voltages and currents from the form: $[V_{1,c},\ V_{1,s},\ V_{2,c},\ \ldots,\ I_{1,c},\ I_{1,s},\ I_{2,c},\ \ldots]^T$ to $[V_{1,c},\ V_{2,c},\ V_{3,c},\ V_{1,s} \ldots,\ I_{1,c},\ I_{2,c},\ I_{3,c},\ \ldots]^T$. Basically, it groups first all core cable voltages, then all sheath voltages, ...

- Apply transposition from A-B-C to C-A-B for all minor sections except for the first: $\mathbf{M}_{CB}\,\mathbf{T}\,\mathbf{M}_{\eta,i}\,\mathbf{T}^{-1}$, where $\mathbf{M}_{CB}$ introduces sheath cross-bonding losses. Matrix $\mathbf{M}_{CB}$ is identity matrix except to the indices that belong to interconnection of sheath voltages and currents. Assuming that $n_c = 3$ is number of the cables, sheath is the second layer of the total n_l layers and thus $\mathbf{M}_{CB}\,\langle n_c+1 : 2n_c, n_c * n_l + n_c + 1 : n_c * n_l + 2n_c\rangle = \mathrm{diag}\{2Z_{CB}\}_{n_c \times n_c}$. Impedance Z_{CB} presents introduced losses.
- Apply ABCD reduction introduced in [17] and described in section 2.

According to the previous description, ABCD parameters of the major section are:

$$ABCD_\eta = M_{\eta,1} \times \prod_{i=2}^{m} \mathbf{M}_{CB}\,(\mathbf{T}\,\mathbf{M}_{\eta,i}\,\mathbf{T}^{-1}) \tag{42}$$

4.4.4 Mixed OHL-cables

Mixed OHL-cable components contain OHLs and cable sections in a desired order. Each OHL and cable sections are characterized individually and a complete component is presented with the equivalent ABCD representation.

The ABCD representation has the form:

$$ABCD = \prod_{\eta=1}^{n} ABCD_\eta, \tag{43}$$

where $ABCD_\eta$ are ABCD parameters of an OHL or cable section, while n is the total number of sections.

4.5 AC and DC grid equivalents

AC and DC grids are represented by AC and DC sources. They are described using the following relations:

$$\begin{aligned} \mathbf{V}_p &= \mathbf{V}_s + \mathbf{Z}\,\mathbf{I}_s + \mathbf{V}, \\ \mathbf{I}_p &= \mathbf{I}_s. \end{aligned} \tag{44}$$

For the estimation of the equivalent impedance of the network, independent voltage sources are short circuited, which means that in this case, the grid ABCD parameters can be represented as an identity matrix. Additionally, the internal grid impedance can be added as an serial connection of the impedance and voltage source. The matrix $\mathbf{Z}$ is thus diagonal with the values of the series impedance: $\mathbf{Z} = \mathrm{diag}\{Z_s\}$.

The ABCD parameters of the equivalent network are then:

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{Z} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}. \tag{45}$$

4.6 MMC

Voltage source converters (VSC) are often implemented as modular multiterminal converters (MMC). They are used for the fast and efficient conversion of energy. In this work they are represented as inverters or rectifiers, which have two DC side pins and three AC side pins. The converter is represented with its admittance matrix which is incorporated with the ABCD system of equations.

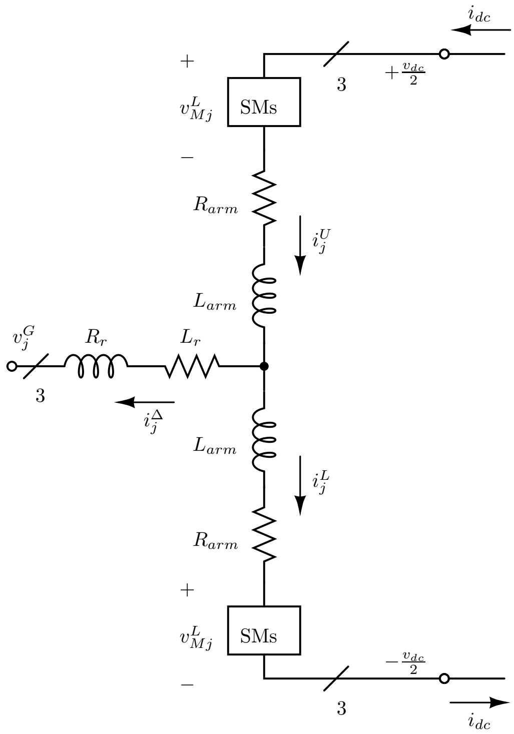


Figure 15: MMC.

4.6.1 MMC model

An MMC is depicted in Fig. 15. The variables from Fig. 15 are defined for all three phases, $j \in \{a, b, c\}$. The sets of submodules are represented by their averaged equivalent, and thus, the following equations for voltages and currents can be written:

$$\begin{aligned} v_{Mj}^{U,L} &= m_j^{U,L} v_{Cj}^{U,L}, \\ i_{Mj}^{U,L} &= m_j^{U,L} i_j^{U,L}, \end{aligned} \tag{46}$$

where $m_j^{U,L}$ are the corresponding insertion indices.

Using $\Sigma - \Delta$ nomenclature, the variables can be represented as:

$$\begin{aligned} i_j^\Delta &= i_j^U - i_j^L, \quad i_j^\Sigma = \frac{i_j^U + i_j^L}{2}, \\ v_{Cj}^\Delta &= \frac{v_{Cj}^U - v_{Cj}^L}{2}, \quad v_{Cj}^\Sigma = \frac{v_{Cj}^U + v_{Cj}^L}{2}, \\ m_j^\Delta &= m_j^U - m_j^L, \quad m_j^\Sigma = m_j^U + m_j^L, \\ v_{Mj}^\Delta &= \frac{-v_{Mj}^U + v_{Mj}^L}{2} = -\frac{m_j^\Delta v_{Cj}^\Sigma + m_j^\Sigma v_{Cj}^\Delta}{2}, \\ v_{Mj}^\Sigma &= \frac{v_{Mj}^U + v_{Mj}^L}{2} = \frac{m_j^\Sigma v_{Cj}^\Sigma + m_j^\Delta v_{Cj}^\Delta}{2}, \end{aligned}$$

In order to obtain the differential equations in the dqz frame, Park's transformation is used represent the system in a set of several rotating frames, each of which being related to a different angular frequency. Park's transformation is defined in A.2.

For the purpose of the modeling, the MMC converter is represented using 12 differential equations for the state variables [35, 36]:

$$
\begin{aligned}
\frac{di_d^\Delta}{dt} &= -\frac{v_d^G - v_{Md}^\Delta + R_{eq}^{ac} i_d^\Delta + \omega L_{eq}^{ac} i_q^\Delta}{L_{eq}^{ac}}, \\
\frac{di_q^\Delta}{dt} &= -\frac{v_q^G - v_{Mq}^\Delta + R_{eq}^{ac} i_q^\Delta - \omega L_{eq}^{ac} i_d^\Delta}{L_{eq}^{ac}}, \\
\frac{di_d^\Sigma}{dt} &= -\frac{v_{Md}^\Sigma + R_{arm} i_d^\Sigma - 2\omega L_{arm} i_q^\Sigma}{L_{arm}}, \\
\frac{di_q^\Sigma}{dt} &= -\frac{v_{Mq}^\Sigma + R_{arm} i_q^\Sigma + 2\omega L_{arm} i_d^\Sigma}{L_{arm}}, \\
\frac{di_z^\Sigma}{dt} &= -\frac{v_{Mz}^\Sigma - \frac{v_{dc}}{2} + R_{arm} i_z^\Sigma}{L_{arm}}, \\
\frac{dv_{Cd}^\Delta}{dt} &= \frac{N}{2C_{arm}} \left(i_z^\Sigma m_d^\Delta - \frac{i_q^\Delta m_q^\Sigma}{4} + i_d^\Sigma \left(\frac{m_d^\Delta}{2} + \frac{m_{Zd}^\Delta}{2} \right) - i_q^\Sigma \left(\frac{m_q^\Delta}{2} + \frac{m_{Zq}^\Delta}{2} \right) \right. \\
&\quad \left. i_d^\Delta \left(\frac{m_d^\Sigma}{4} + \frac{m_z^\Sigma}{2} \right) - 2\omega C_{arm}\, v_{Cq}^\Delta \right), \\
\frac{dv_{Cq}^\Delta}{dt} &= -\frac{N}{2C_{arm}} \left(\frac{i_d^\Delta m_q^\Sigma}{4} - i_z^\Sigma m_q^\Delta + i_q^\Sigma \left(\frac{m_d^\Delta}{2} - \frac{m_{Zd}^\Delta}{2} \right) + i_d^\Sigma \left(\frac{m_q^\Delta}{2} - \frac{m_{Zq}^\Delta}{2} \right) \right. \\
&\quad \left. i_q^\Delta \left(\frac{m_d^\Sigma}{4} - \frac{m_z^\Sigma}{2} \right) - 2\omega C_{arm}\, v_{Cd}^\Delta \right), \\
\frac{dv_{CZd}^\Delta}{dt} &= -\frac{N}{8C_{arm}} \left(i_d^\Delta m_d^\Sigma + 2 i_d^\Sigma m_d^\Delta + i_q^\Delta m_q^\Sigma + 2 i_q^\Sigma m_q^\Delta + 4 i_z^\Sigma m_{Zd}^\Delta \right) - 3\omega v_{CZq}^\Delta, \\
\frac{dv_{CZq}^\Delta}{dt} &= -\frac{N}{8C_{arm}} \left(i_q^\Delta m_d^\Sigma + 2 i_d^\Sigma m_q^\Delta - i_d^\Delta m_q^\Sigma - 2 i_q^\Sigma m_d^\Delta + 4 i_z^\Sigma m_{Zq}^\Delta \right) + 3\omega v_{CZd}^\Delta, \\
\frac{dv_{Cd}^\Sigma}{dt} &= \frac{N}{2C_{arm}} \left(i_d^\Sigma m_z^\Sigma + i_z^\Sigma m_d^\Sigma + i_d^\Delta \left(\frac{m_d^\Delta}{4} + \frac{m_{Zd}^\Delta}{4} \right) - i_q^\Delta \left(\frac{m_q^\Delta}{4} - \frac{m_{Zq}^\Delta}{4} \right) \right) + 2\omega C_{arm} v_{Cq}^\Sigma, \\
\frac{dv_{Cq}^\Sigma}{dt} &= -\frac{N}{2C_{arm}} \left(i_q^\Delta \left(\frac{m_d^\Delta}{4} - \frac{m_{Zd}^\Delta}{4} \right) - i_z^\Sigma m_q^\Sigma + i_d^\Delta \left(\frac{m_q^\Delta}{4} + \frac{m_{Zq}^\Delta}{4} \right) - i_q^\Sigma m_z^\Sigma \right) + 2\omega C_{arm} v_{Cd}^\Sigma, \\
\frac{dv_{Cz}^\Sigma}{dt} &= -\frac{N}{8C_{arm}} \left(i_d^\Delta m_d^\Delta + i_q^\Delta m_q^\Delta + 2 i_d^\Sigma m_d^\Sigma + 2 i_q^\Sigma m_q^\Sigma + 4 i_z^\Sigma m_z^\Sigma \right),
\end{aligned} \tag{47}
$$

where $L_{eq}^{ac} = L_f + \frac{L_{arm}}{2}$ and $R_{eq}^{ac} = R_f + \frac{R_{arm}}{2}$. The state variables are $\mathbf{x} = [\mathbf{i}_{dq}^\Delta,\ \mathbf{i}_{dqz}^\Sigma,\ \mathbf{v}_{CdqZ}^\Delta,\ \mathbf{v}_{Cdqz}^\Sigma]^T$. The 12 algebraic relations used for determining 7 voltages $[v_{Md}^\Delta, v_{Mq}^\Delta, v_{MZd}^\Delta, v_{MZq}^\Delta, v_{Md}^\Sigma, v_{Mq}^\Sigma, v_{Mz}^\Sigma]$

and insertation indeices $\left[m_d^\Delta,\quad m_q^\Delta,\quad m_{Zd}^\Delta,\quad m_{Zq}^\Delta,\quad m_d^\Sigma,\quad m_q^\Sigma,\quad m_z^\Sigma\right]^T$ are given as:

$$
\begin{aligned}
v_{Md}^\Delta &= \frac{m_q^\Delta v_{Cq}^\Sigma}{4} - \frac{m_d^\Delta v_{Cz}^\Sigma}{2} - \frac{m_d^\Delta v_{Cd}^\Sigma}{4} - \frac{m_{Zd}^\Delta v_{Cd}^\Sigma}{4} + \frac{m_{Zq}^\Delta v_{Cq}^\Sigma}{4} - \frac{m_d^\Sigma v_{Cd}^\Delta}{4} - \frac{m_z^\Sigma v_{Cd}^\Delta}{2} + \frac{m_q^\Sigma v_{Cq}^\Delta}{4} \\
&\quad - \frac{m_d^\Sigma v_{CZd}^\Delta}{4} + \frac{m_q^\Sigma v_{CZq}^\Delta}{4}, \\
v_{Mq}^\Delta &= \frac{m_d^\Delta v_{Cq}^\Sigma}{4} + \frac{m_q^\Delta v_{Cd}^\Sigma}{4} - \frac{m_q^\Delta v_{Cz}^\Sigma}{2} - \frac{m_{Zd}^\Delta v_{Cq}^\Sigma}{4} - \frac{m_{Zq}^\Delta v_{Cd}^\Sigma}{4} + \frac{m_d^\Sigma v_{Cq}^\Delta}{4} + \frac{m_q^\Sigma v_{Cd}^\Delta}{4} - \frac{m_z^\Sigma v_{Cq}^\Delta}{2} \\
&\quad - \frac{m_d^\Sigma v_{CZq}^\Delta}{4} - \frac{m_q^\Sigma v_{CZd}^\Delta}{4}, \\
v_{MZd}^\Delta &= -\frac{m_d^\Delta v_{Cd}^\Sigma}{4} - \frac{m_q^\Delta v_{Cq}^\Sigma}{4} - \frac{m_{Zd}^\Delta v_{Cz}^\Sigma}{2} - \frac{m_d^\Sigma v_{Cd}^\Delta}{4} - \frac{m_q^\Sigma v_{Cq}^\Delta}{4} - \frac{m_z^\Sigma v_{Zd}^\Delta}{2}, \\
v_{MZq}^\Delta &= -\frac{m_d^\Delta v_{Cq}^\Sigma}{4} - \frac{m_q^\Delta v_{Cd}^\Sigma}{4} - \frac{m_{Zq}^\Delta v_{Cz}^\Sigma}{2} - \frac{m_d^\Sigma v_{Cq}^\Delta}{4} + \frac{m_q^\Sigma v_{Cd}^\Delta}{4} - \frac{m_z^\Sigma v_{Zq}^\Delta}{2}, \\
v_{Md}^\Sigma &= \frac{m_d^\Delta v_{Cd}^\Delta}{4} - \frac{m_q^\Delta v_{Cq}^\Delta}{4} + \frac{m_d^\Delta v_{CZd}^\Delta}{4} + \frac{m_{Zd}^\Delta v_{Cd}^\Delta}{4} + \frac{m_q^\Delta v_{Zq}^\Delta}{4} + \frac{m_{Zq}^\Delta v_{Cq}^\Delta}{4} + \frac{m_d^\Sigma v_{Cz}^\Sigma}{2} + \frac{m_z^\Sigma v_{Cd}^\Sigma}{2}, \\
v_{Mq}^\Sigma &= \frac{m_q^\Delta v_{Zd}^\Delta}{4} - \frac{m_q^\Delta v_{Cd}^\Delta}{4} - \frac{m_d^\Delta v_{Zq}^\Delta}{4} - \frac{m_d^\Delta v_{Cq}^\Delta}{4} + \frac{m_{CZd}^\Delta v_{Cq}^\Delta}{4} - \frac{m_{Zq}^\Delta v_{Cd}^\Delta}{4} + \frac{m_q^\Sigma v_{Cz}^\Sigma}{2} + \frac{m_z^\Sigma v_{Cq}^\Sigma}{2}, \\
v_{Mz}^\Sigma &= \frac{m_d^\Delta v_{Cd}^\Delta}{4} + \frac{m_q^\Delta v_{Cq}^\Delta}{4} + \frac{m_{Zd}^\Delta v_{CZd}^\Delta}{4} + \frac{m_{Zq}^\Delta v_{CZq}^\Delta}{4} + \frac{m_d^\Sigma v_{Cd}^\Sigma}{4} + \frac{m_q^\Sigma v_{Cq}^\Sigma}{4} + \frac{m_z^\Sigma v_{Cz}^\Sigma}{2},
\end{aligned}
\tag{48}
$$

$$
\begin{bmatrix} m_d^\Delta \\ m_q^\Delta \\ m_{Zd}^\Delta \\ m_{Zq}^\Delta \\ m_d^\Sigma \\ m_q^\Sigma \\ m_z^\Sigma \end{bmatrix} = \frac{2}{v_{dc}} \begin{bmatrix} -v_{Md,ref}^\Delta \\ -v_{Mq,ref}^\Delta \\ -v_{MZd,ref}^\Delta \\ -v_{MZq,ref}^\Delta \\ v_{Md,ref}^\Sigma \\ v_{Mq,ref}^\Sigma \\ v_{Mz,ref}^\Sigma \end{bmatrix}.
\tag{49}
$$

The set of the previous 12 differential equations and the set of algebraic equations are accompanied with the 7 equations for the reference values of the voltages $[\mathbf{v}_{MdqZ,ref}^\Delta, \mathbf{v}_{Mdqz,ref}^\Sigma]$. The reference voltages are given as zero by default, except for the value of $v_{Cz,ref}^\Sigma = \frac{v_{dc}}{2}$.

4.6.2 Operating point

Converter's operating point can be added manually and by solving the power flow of the power system. In both situations, the following fields should be present:

- P_{min}, P_{max} minimum and maximum active AC power of the converter;
- P estimated active AC power;
- Q_{min}, Q_{max} minimum and maximum reactive power;
- Q estimated reactive power;
- P_{dc} estimated DC power;
- V_{DC} DC voltage;
- V_m, θ amplitude and phase of the AC voltage.

Using previous fields, the converter's operating point is estimated by solving a set of linear differential equations to obtain converter's steady-state. As a reference for the currents are chosen values:

$$\begin{aligned}
i_{d,ref}^{\Delta C} &= \frac{2}{3}\frac{(v_d^{GC}P + v_q^{GC}Q)}{v_d^{GC2} + v_q^{GC2}}, \\
i_{q,ref}^{\Delta C} &= \frac{2}{3}\frac{(v_q^{GC}P - v_d^{GC}Q)}{v_d^{GC2} + v_q^{GC2}}, \\
i_{z,ref}^{\Sigma} &= \frac{P_{dc}}{3V_{DC}}, \\
P_{ac,ref} &= P, \\
Q_{ar,ref} &= Q, \\
v_{dc,ref} &= V_{DC}, \\
W_{z,ref}^{\Sigma} &= \frac{3C_{arm}V_{DC}^2}{N}.
\end{aligned} \tag{50}$$

4.6.3 Control implementations

For the PI controls in the dqz frame are developed additional equations [35–37]. Several controlling methods are considered tuned using pole placement method. PLL is also implemented using PI controller [38].

Phase locked loop (PLL) PLL is used to synchronize converter's frequency to the grid frequency. As all converter's variables are mapped to dqz frame using the same Park's transformation without phase shift, in order to map all controls employing the frequency dependency, the following control law is formulated. According to Fig. 16 the following equations for PLL can be written.

$$\begin{aligned}
\frac{d\xi_{pll}}{dt} &= -v_q^{G,C}, \\
\frac{d\theta}{dt} &= \Delta\omega, \\
\Delta\omega &= -K_{p,pll}\, v_q^{G,C} + K_{i,pll}\, \xi_{pll}, \\
\omega_C &= \Delta\omega + \omega_0.
\end{aligned} \tag{51}$$

It should be noted that output current control and circulating current control are implemented in the converter's reference frame. Thus, the rotation should be applied to the corresponding variables before the control law is applied. However, since the converter's internal dynamics is analyzed in grid's reference frame, output of the mentioned controls should be restored to the grid's reference frame using inverse of the rotation matrix. Rotation matrix is given with:

$$T(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}, \tag{52}$$

while its inverse is:

$$T^{-1}(\theta) = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix}. \tag{53}$$

Mapping of the $i_{dq,ref}^{\Delta}$ from grid's to converter's reference frame is done by:

$$\begin{bmatrix} i_{d,ref}^{\Delta C} \\ i_{q,ref}^{\Delta C} \end{bmatrix} = T(\theta)\ \begin{bmatrix} i_{d,ref}^{\Delta} \\ i_{q,ref}^{\Delta} \end{bmatrix},$$

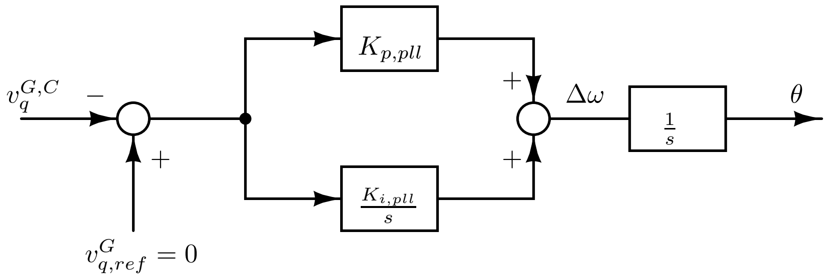


Figure 16: PLL implementation.

and currents i_d^Δ and i_q^Δ are also mapped to:

$$\begin{bmatrix} i_d^{\Delta C} \\ i_q^{\Delta C} \end{bmatrix} = T(\theta) \begin{bmatrix} i_d^{\Delta} \\ i_q^{\Delta} \end{bmatrix}.$$

Similarly:

$$\begin{bmatrix} i_{d,ref}^{\Sigma C} \\ i_{q,ref}^{\Sigma C} \end{bmatrix} = T(-2\theta) \begin{bmatrix} i_{d,ref}^{\Sigma} \\ i_{q,ref}^{\Sigma} \end{bmatrix}, \qquad \begin{bmatrix} i_d^{\Sigma C} \\ i_q^{\Sigma C} \end{bmatrix} = T(-2\theta) \begin{bmatrix} i_d^{\Sigma} \\ i_q^{\Sigma} \end{bmatrix}.$$

DC voltage control DC voltage control (DCC) provides the reference value for the $i_{d,ref}^{\Delta C}$ depending of the variation of v_{dc}. The control provides following equations

$$\begin{aligned} \frac{dv_{dc}}{dt} &= \frac{6N}{C_{arm}} \left(\frac{P_{dc,ref}}{V_{DC}} - 3i_z^\Sigma \right), && (54) \\ \frac{d\xi_{v_{dc}}}{dt} &= v_{dc,ref} - v_{dc}, && (55) \\ i_{d,ref}^{\Delta C} &= -K_{p,dc} \left(v_{dc,ref} - v_{dc} \right) - K_{i,dc}\, \xi_{v_{dc}} 3i_z^\Sigma. \end{aligned}$$

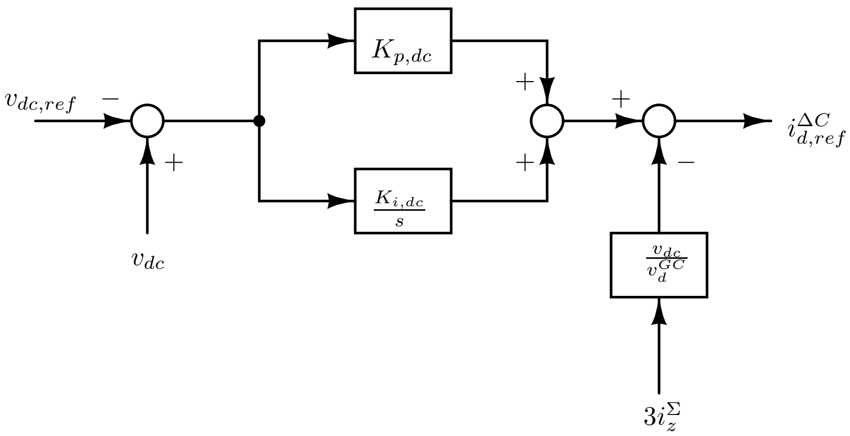


Figure 17: DC voltage control.

Output current control (OCC) defines the reference values for the output currents $i_{d,ref}^\Delta$ and $i_{q,ref}^\Delta$ given in the grid reference frame. This control method adds several equations:

$$\begin{aligned} \frac{d\xi_d^\Delta}{dt} &= i_{d,ref}^{\Delta C} - i_d^{\Delta C}, \\ \frac{d\xi_q^\Delta}{dt} &= i_{q,ref}^{\Delta C} - i_q^{\Delta C}, \\ v_{Md,ref}^{\Delta C} &= K_{i,occ}\xi_d^\Delta + K_{p,occ} \left(i_{d,ref}^{\Delta C} - i_d^{\Delta C} \right) + \omega_C L_{eq}^{ac} i_q^\Delta + v_d^{G,C}, \\ v_{Mq,ref}^{\Delta C} &= K_{i,occ}\xi_q^\Delta + K_{p,occ} \left(i_{q,ref}^{\Delta C} - i_q^{\Delta C} \right) - \omega_C L_{eq}^{ac} i_d^{\Delta C} + v_q^{G,C}. && (56) \end{aligned}$$

Voltages $v_{Md,ref}^{\Delta C}$ and $v_{Mq,ref}^{\Delta C}$ are used in grid's reference frame for the further calculations:

$$\begin{bmatrix} v_{Md,ref}^{\Delta} \\ v_{Mq,ref}^{\Delta} \end{bmatrix} = T^{-1}(\theta) \; \begin{bmatrix} v_{Md,ref}^{\Delta C} \\ v_{Mq,ref}^{\Delta C} \end{bmatrix}. \tag{57}$$

If the controller is defined only using bandwidth ω_n and ζ, the proportional and integral gains are then tuned as:

$$\begin{aligned} K_{i,occ} &= L_{eq}^{ac}\,\omega_n^2, \\ K_{p,occ} &= 2\zeta\omega_n L_{eq}^{ac} - R_{eq}^{ac}. \end{aligned}$$

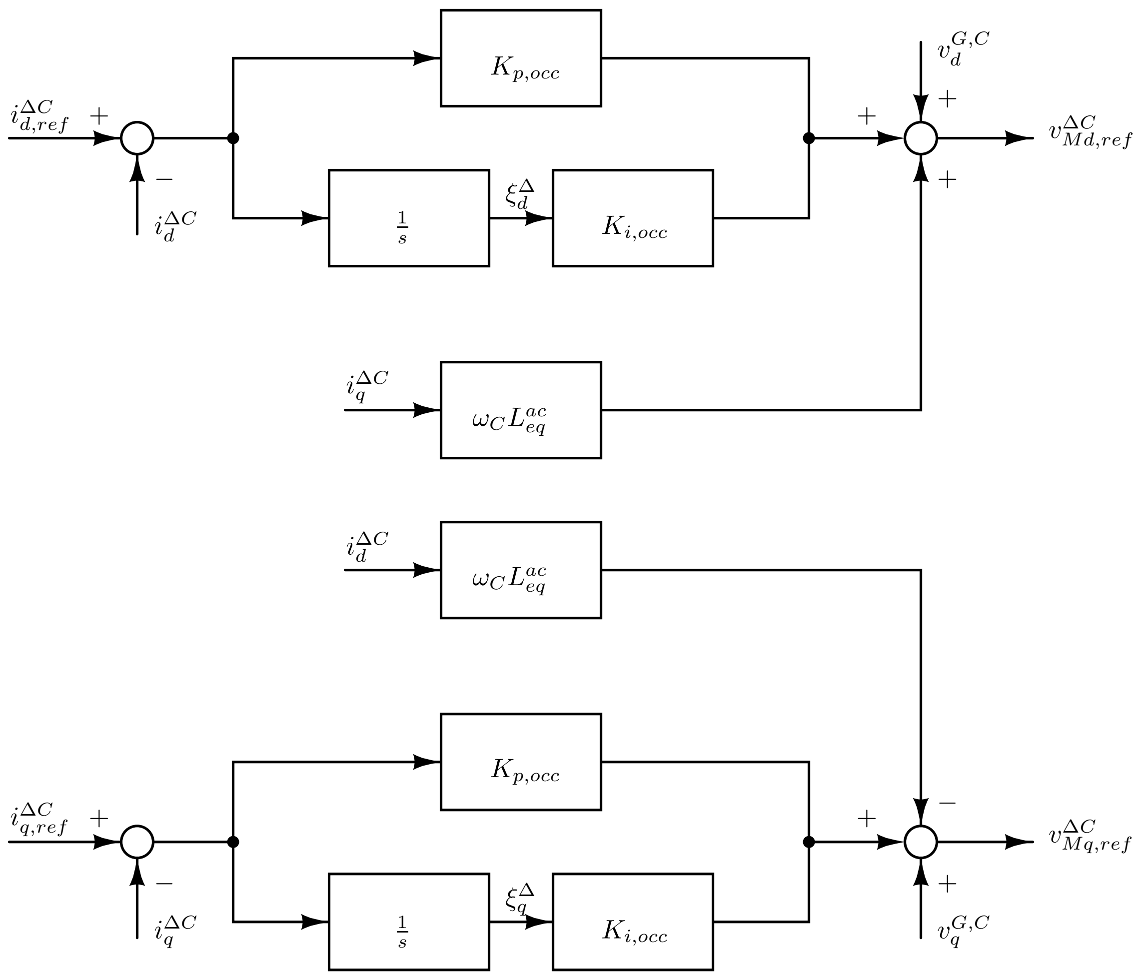


Figure 18: OCC implementation.

Circulating current control (CCC) is constructed to set the circulating current to it reference, which is considered to be $i_{d,ref}^{\Sigma} = 0$, $i_{q,ref}^{\Sigma} = 0$.

The equations added by CCC are:

$$\begin{aligned} \frac{d\xi_d^{\Sigma}}{dt} &= i_{d,ref}^{\Sigma C} - i_d^{\Sigma C}, \\ \frac{di_q^{\Sigma}}{dt} &= i_{q,ref}^{\Sigma C} - i_q^{\Sigma C}, \\ v_{Md,ref}^{\Sigma C} &= -K_{i,ccc}\,\xi_d^{\Sigma} - K_{p,ccc}\,(i_{d,ref}^{\Sigma C} - i_d^{\Sigma C}) + 2\omega_C L_{arm} i_q^{\Sigma C}, \\ v_{Mq,ref}^{\Sigma C} &= -K_{i,ccc}\xi_q^{\Sigma} - K_{p,ccc}\,(i_{q,ref}^{\Sigma C} - i_q^{\Sigma C}) - 2\omega_C L_{arm} i_d^{\Sigma C}. \end{aligned} \tag{58}$$

To return to grid's reference frame, the following is applied:

$$\begin{bmatrix} v^{\Sigma}_{Md,ref} \\ v^{\Sigma}_{Mq,ref} \end{bmatrix} = T^{-1}(-2\theta) \begin{bmatrix} v^{\Sigma C}_{Md,ref} \\ v^{\Sigma C}_{Mq,ref} \end{bmatrix}. \tag{59}$$

The proportional and integral gains are tuned as:

$$\begin{aligned} K_i &= L_{arm}\,\omega_n^2, \\ K_p &= 2\zeta\omega_n L_{arm} - R_{arm}. \end{aligned}$$

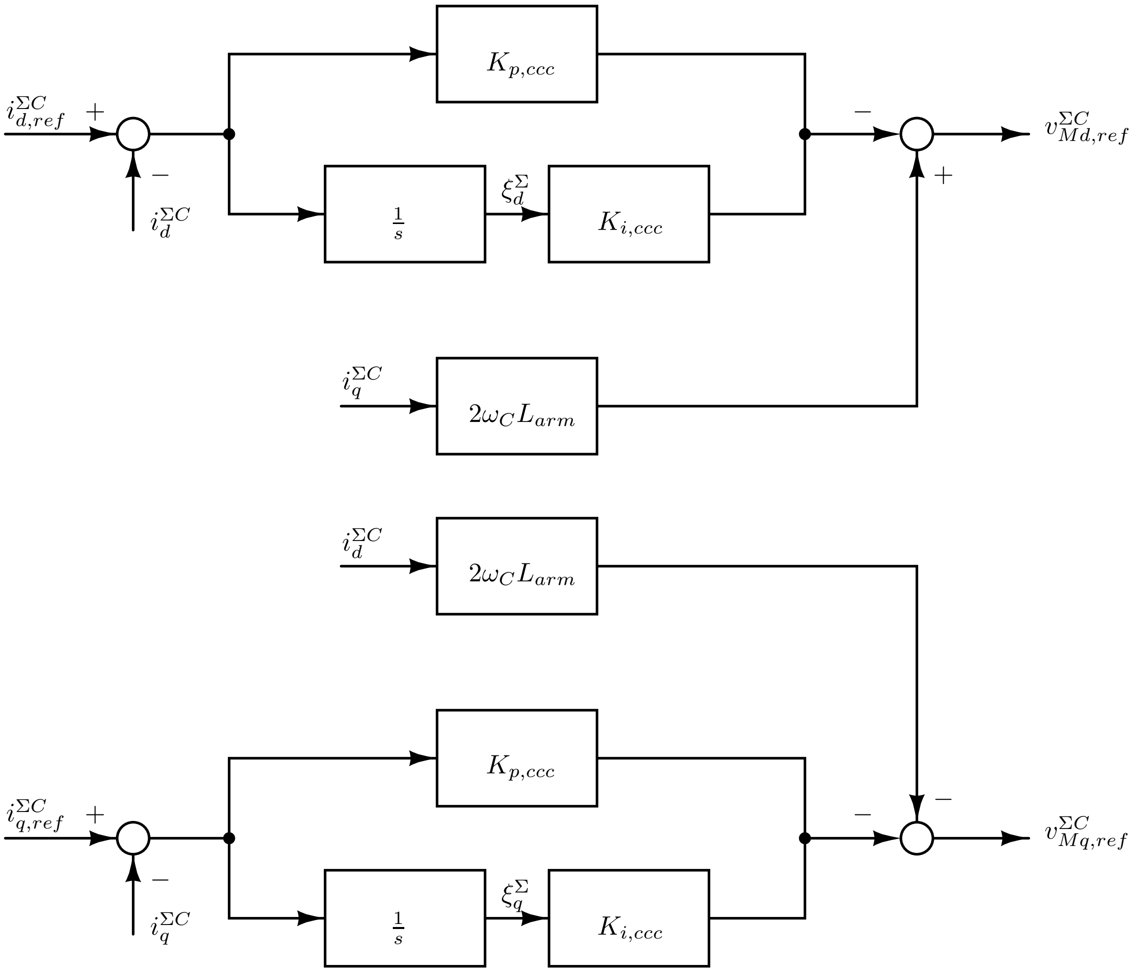


Figure 19: CCC implementation.

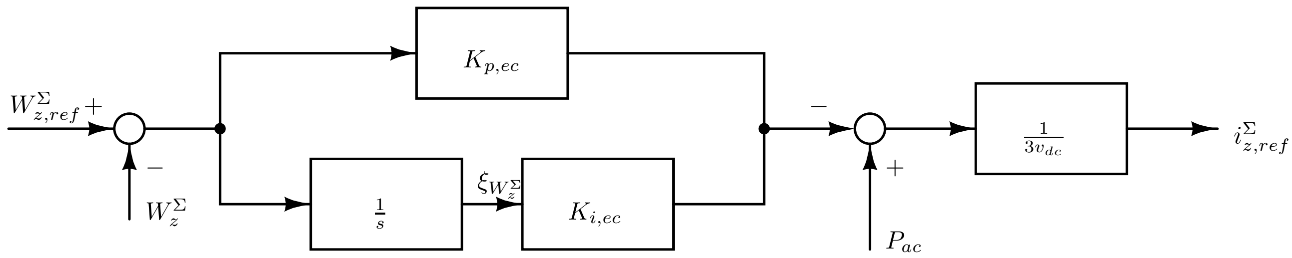
Energy control and zero current control Energy control is built around the "zero" energy and as a result it provides a reference value for the "zero" current $i^{\Sigma}_{z,ref}$. Energy controller involves the following equations as visible from Fig. 20a.

$$\begin{aligned} W^{\Sigma}_{z} &= \frac{3C_{arm}}{2N}\,({v^{\Delta}_{Cd}}^2 + {v^{\Delta}_{Cq}}^2 + {v^{\Delta}_{CZd}}^2 + {v^{\Delta}_{CZq}}^2 + {v^{\Sigma}_{Cd}}^2 + {v^{\Sigma}_{Cq}}^2 + 2{v^{\Sigma}_{Cz}}^2), \\ \frac{d\xi_{W^{\Sigma}_{z}}}{dt} &= W^{\Sigma}_{z,ref} - W^{\Sigma}_{z}, \\ P_{ac} &= \frac{3}{2}\,(v^{G,C}_{d} i^{\Delta C}_{d} + v^{G,C}_{q} i^{\Delta C}_{q}), \\ i^{\Sigma}_{z,ref} &= \frac{K_{p,ec}\,(W^{\Sigma}_{z,ref} - W^{\Sigma}_{z}) + K_{i,ec}\xi_{W^{\Sigma}_{z}} + P_{ac}}{3v_{dc}}. \end{aligned} \tag{60}$$

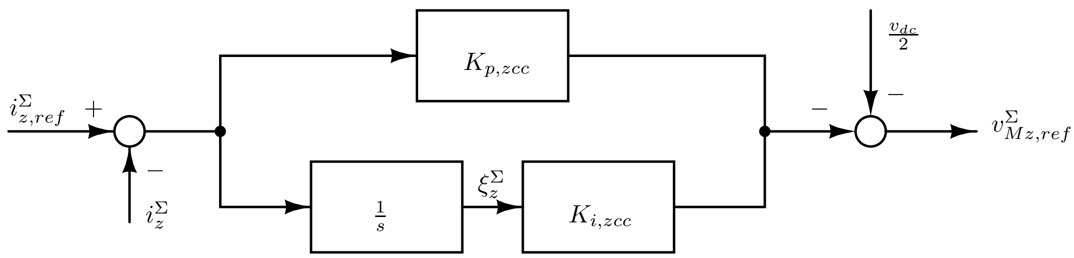
Additionally, the zero current control (ZCC) sets the zero current to the desired value. The implementation of this control is depicted in Fig. 20b. It can work without the energy controller.

$$\begin{aligned}\frac{d\xi_z^\Sigma}{dt} &= i_{z,ref}^\Sigma - i_z^\Sigma,\\ v_{Mz,ref}^\Sigma &= \frac{v_{dc}}{2} - K_{p,zcc}\,(i_{z,ref}^\Sigma - i_z^\Sigma) - K_{i,zcc}\,\xi_z^\Sigma.\end{aligned} \tag{61}$$

Tuning of the ZCC is the same as for CCC.



(a)



(b)

Figure 20: Energy control and ZCC implementation.

Active and reactive power control An outer control loop for the control of the active and reactive power can be added, see Fig. 21. These control loops are used to successfully estimate AC currents $i_{dq,ref}^\Delta$. The control loops operate according to equations:

$$\begin{aligned}P_{ac} &= \frac{3}{2}\,(v_d^{G,C} i_d^{\Delta C} + v_q^{G,C} i_q^{\Delta C}),\\ Q_{ac} &= \frac{3}{2}\,(-v_d^{G,C} i_q^{\Delta C} + v_q^{G,C} i_d^{\Delta C}),\\ \frac{d\xi_{P_{ac}}}{dt} &= P_{ac,ref} - P_{ac},\\ \frac{d\xi_{Q_{ac}}}{dt} &= Q_{ac,ref} - Q_{ac},\\ i_{d,ref}^{\Delta C} &= K_p^{P_{ac}}\,(P_{ac,ref} - P_{ac}) + K_i^{P_{ac}} \xi_{P_{ac}},\\ i_{q,ref}^{\Delta C} &= -K_p^{Q_{ac}}\,(Q_{ac,ref} - Q_{ac}) - K_i^{Q_{ac}} \xi_{Q_{ac}}.\end{aligned} \tag{62}$$

4.6.4 Steady state solution and admittance model

The previous system of differential and algebraic equations is solved for the equilibrium using Julia package NLsolve. After determining of the equilibrium, the system is represented as a

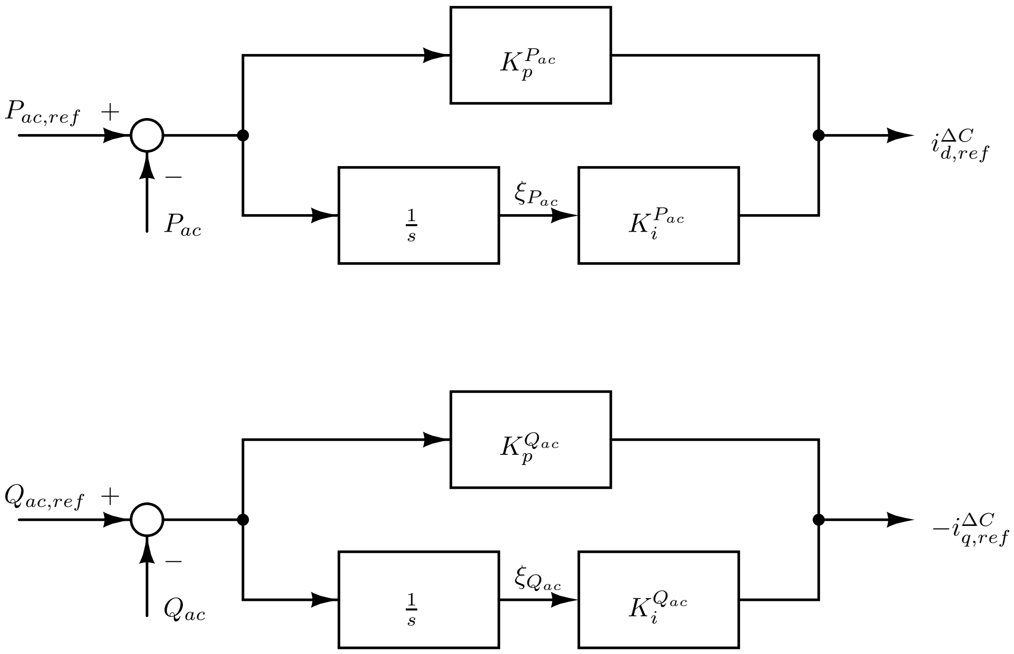


Figure 21: Active and reactive power control implementation.

multi-input multi-output system (MIMO), where the variables $\mathbf{x} = \begin{bmatrix}\mathbf{i}_{dq}^{\Delta} & \mathbf{i}_{dqz}^{\Sigma} & \mathbf{v}_{CdqZ}^{\Delta} & \mathbf{v}_{Cdqz}^{\Sigma}\end{bmatrix}$ represent the state-variables and input vector is given as $\mathbf{u} = \begin{bmatrix}v_{dc} & v_d^G & v_q^G\end{bmatrix}$. In order to get transfer functions from the input to output given with the values $\mathbf{y} = \begin{bmatrix}3i_z^{\Sigma} & i_d^{\Delta} & i_q^{\Delta}\end{bmatrix}$, the previous equations are rewritten to satisfy the following form:

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{A}_{MIMO}\mathbf{x}(t) + \mathbf{B}_{MIMO}\mathbf{u}(t),\\ \mathbf{y}(t) &= \mathbf{C}_{MIMO}\mathbf{x}(t) + \mathbf{D}_{MIMO}\mathbf{u}(t).\end{aligned} \tag{63}$$

Corresponding matrices $\mathbf{A}_{MIMO}$, $\mathbf{B}_{MIMO}$, $\mathbf{C}_{MIMO}$ and $\mathbf{D}_{MIMO}$ are determined as Jacobians around the equilibrium for the state variables and inputs. Jacobian is successfully determined using Julia package ForwardDiff [39]. Applying the Laplace transform, the previous system of equations (63) transforms to:

$$\begin{aligned}s\mathbf{X}(s) &= \mathbf{A}_{MIMO}\mathbf{X}(s) + \mathbf{B}_{MIMO}\mathbf{U}(s),\\ \mathbf{Y}(s) &= \mathbf{C}_{MIMO}\mathbf{X}(s) + \mathbf{D}_{MIMO}\mathbf{U}(s).\end{aligned} \tag{64}$$

The MIMO transfer function is thus given by:

$$\mathbf{Y}_{MMC}(s) = \mathbf{Y}(s)\mathbf{U}(s)^{-1} = \mathbf{C}_{MIMO}\ (s\mathbf{I} - \mathbf{A}_{MIMO})^{-1}\ \mathbf{B}_{MIMO} + \mathbf{D}_{MIMO}. \tag{65}$$

Obtained matrix transfer function presents admittances

$$\mathbf{Y}_{MMC}(s) = \begin{bmatrix} Y_{zz} & Y_{zd} & Y_{zq}\\ Y_{dz} & Y_{dd} & Y_{dq}\\ Y_{qz} & Y_{qd} & Y_{qq}\end{bmatrix} \tag{66}$$

that connect vector of currents $[i_{dc}(s)\, i_d^{\Delta}(s),\ i_q^{\Delta}(s),\ ]^T$ with the voltages vector $[v_{dc}(s),\ v_d^G(s),\ v_q^G(s)]^T$.

If we model the converter as in Fig. 22, then MMC can be represented as one input, two output component.

As ABCD parameters work correctly for the components with the same number of the input and the output pins, the equations which are solved for the MMC are given with the Y_{MMC} matrix.

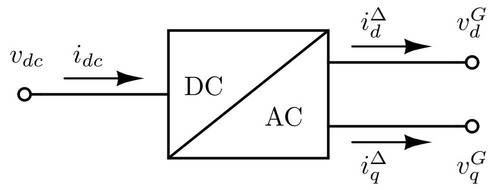


Figure 22: MMC block model.

For the case when the converter controls the DC voltage, then the vector of inputs and outputs are: $\mathbf{u} = \begin{bmatrix} i_{dc} & v_d^G & v_q^G \end{bmatrix}$ and $\mathbf{y} = \begin{bmatrix} v_{dc} & i_d^\Delta & i_q^\Delta \end{bmatrix}$. Then, the system is also represented as MIMO (65), but in order to determine $\mathbf{Y}_{MMC}(s)$, the transformation is applied to change te places of $v_{dc}(s)$ and $i_{dc}(s)$

4.7 Shunt reactor

The simulator allows for the implementation of single-phase and three-phase shunt reactors. It is possible to account for the layering of the component, and the winding of each phase is constructed by connecting all layers in series, as represented in Figure 23(a). The equivalent circuit of the single-phase model is given in Figure 23(b), where inductances, resistances, and parasitic capacitances are modeled as lumped components.

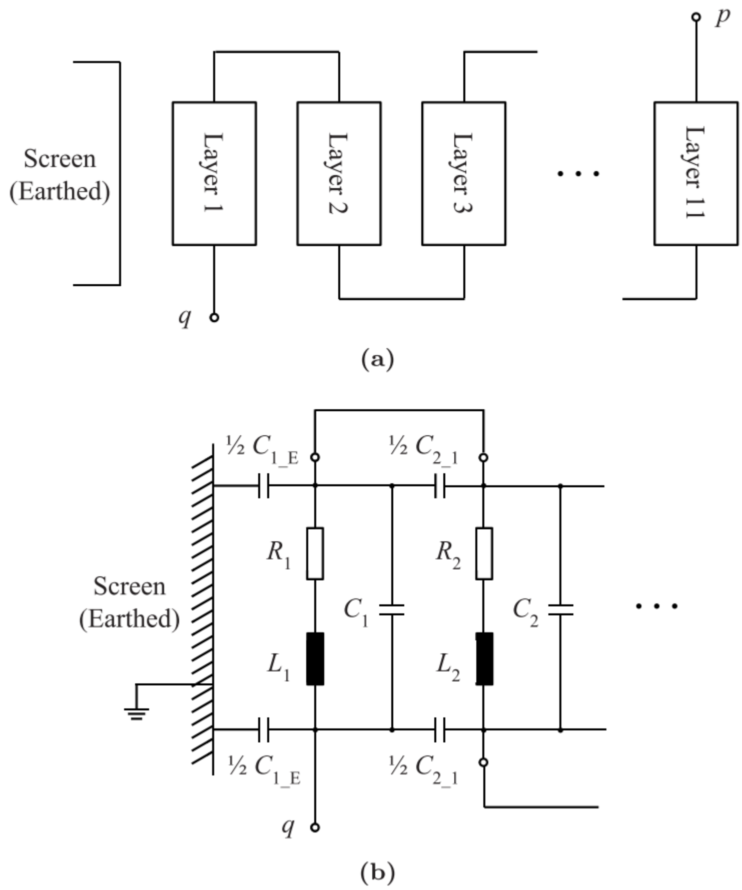


Figure 23: (a) Layers configuration and (b) equivalent circuit diagram for the shunt reactor [17]

A shunt reactor is characterised by the following set of parameters [17, Section 2.5]. For a three-phase shunt reactor, it is assumed that each phase is characterised by the same parameters. The three-phase shunt reactor can be connected in Wye or in Delta.

pins the number of phases;

N the number of layers;

L_k the series inductance of layer k, with $k = 1, ..., N$;

R_k the series resistance of layer k, with $k = 1, ..., N$;

C_k the cross-over resistance of layer k, with $k = 1, ..., N$;

C_{k_k-1} the inter-layer capacitance between layers k and $k-1$, with $k = 2, ..., N$;

C_{1_E} the capacitance between layer 1 and the earthed screen;

If inductances, resistances, cross-over capacitances and inter-layer capacitances are known for each layer, they can be provided as vectors of values. Otherwise, total or average values can be specified, in which case the layer values are obtained $\forall k$ as:

$$L_k = \frac{L_{\text{tot}}}{N} \tag{67}$$

$$R_k = \frac{R_{\text{tot}}}{N} \tag{68}$$

$$C_k = C_{\text{CO, avg}} \tag{69}$$

$$C_{k\_k-1} = C_{\text{IL, avg}} \tag{70}$$

where CO stands for *cross-over* and IL stands for *inter-layer*.

4.7.1 Calculation of the ABCD matrix

The procedure for the determination of the ABCD matrix is as follows. First, the matrix is obtained for a N-port component, assuming that the layers are disconnected (N pins on the p side and N pins on the q side). The connection of the layers in series is done at a later stage by means of boundary conditions. The corresponding $2N$-by-$2N$ ABCD matrix is obtained as:

$$\mathbf{ABCD} = \mathbf{K}_{CIL,q-side} \cdot \mathbf{K}_{RLC} \cdot \mathbf{K}_{CIL,p-side} \tag{71}$$

where the matrices are defined as (see Fig. 24):

ABCD for the overall N-port component with all layers disconnected;

$\mathbf{K}_{CIL,p-side}$ for the N-port component comprising all p-side inter-layer capacitors;

$\mathbf{K}_{RLC}$ the N-port component comprising the RL series components in parallel with the cross-over capacitors;

$\mathbf{K}_{CIL,q-side}$ for the N-port component comprising all q-side inter-layer capacitors.

Matrix $\mathbf{K}_{RLC}$

Matrix $\mathbf{K}_{RLC}$ can be obtained by writing the following set of equations:

$$\begin{cases} I_{b1} = I_{a1} \\ I_{b2} = I_{a2} \\ \vdots \\ I_{bN} = I_{aN} \\ U_{b1} = U_{a1} - ((sL_1 + R_1)^{-1} + sC_1)^{-1} I_{a1} \\ U_{b2} = U_{a2} - ((sL_2 + R_2)^{-1} + sC_2)^{-1} I_{a2} \\ \vdots \\ U_{bN} = U_{aN} - ((sL_N + R_N)^{-1} + sC_N)^{-1} I_{aN} \end{cases} \tag{72}$$

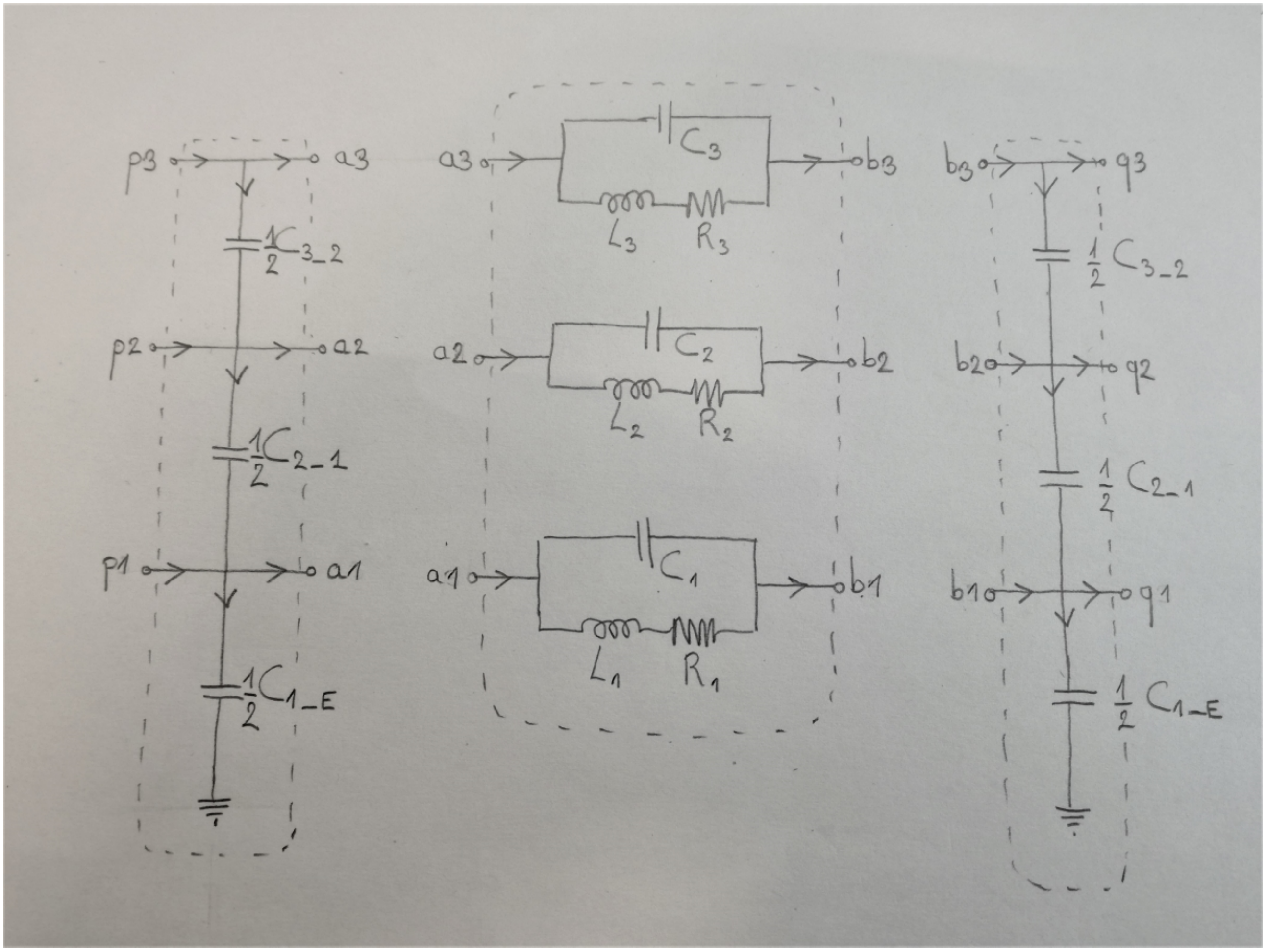


Figure 24: Calculation example for $N = 3$

which results in an ABCD matrix of the form:

$$\mathbf{K}_{RLC} = \begin{bmatrix} \mathbf{I} & \mathbf{Z} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}, \tag{73}$$

with in particular the block element $\mathbf{Z}$:

$$\mathbf{Z} = \begin{bmatrix} -((sL_1 + R_1)^{-1} + sC_1)^{-1} & 0 & \cdots & 0 \\ 0 & -((sL_2 + R_2)^{-1} + sC_2)^{-1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & -((sL_N + R_N)^{-1} + sC_N)^{-1} \end{bmatrix}. \tag{74}$$

Matrices $\mathbf{C}_{IL-LE,p}$ and $\mathbf{C}_{IL-LE,q}$

Matrices $\mathbf{C}_{IL-LE,p}$ and $\mathbf{C}_{IL-LE,q}$ can be obtained by writing the following set of equations, for example for matrix $\mathbf{C}_{IL-LE,p}$:

$$\begin{cases} U_{a1} = U_{p1} \\ U_{a2} = U_{p2} \\ \vdots \\ U_{aN} = U_{pN} \\ I_{a1} = I_{p1} - s\frac{1}{2}C_{1\_E}(U_{p1} - 0) + s\frac{1}{2}C_{2\_1}(U_{p2} - U_{p1}) \\ I_{a2} = I_{p2} - s\frac{1}{2}C_{2\_1}(U_{p2} - U_{p1}) + s\frac{1}{2}C_{3\_2}(U_{p3} - U_{p2}) \\ \vdots \\ I_{aN} = I_{pN} - s\frac{1}{2}C_{N\_N-1}(U_{pN} - U_{p(N-1)}) \end{cases} \tag{75}$$

which results in matrices of the form:

$$\mathbf{K}_{CIL,q-side} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{Y}_q & \mathbf{I} \end{bmatrix}, \qquad \mathbf{K}_{CIL,p-side} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{Y}_p & \mathbf{I} \end{bmatrix}. \tag{76}$$

with in particular the block elements $\mathbf{Y}_p$ and $\mathbf{Y}_q$:

$$\mathbf{Y}_p = \mathbf{Y}_q = s\frac{1}{2}\begin{bmatrix} -C_{1\_E} - C_{2\_1} & C_{2\_1} & 0 & 0 & \cdots & 0 & 0 & 0 \\ C_{2\_1} & -C_{2\_1} - C_{3\_2} & C_{3\_2} & 0 & \cdots & 0 & 0 & 0 \\ 0 & C_{3\_2} & -C_{3\_2} - C_{4\_3} & C_{4\_3} & \cdots & 0 & 0 & 0 \\ \vdots & & & & \ddots & & & \vdots \\ 0 & 0 & 0 & 0 & \cdots & C_{N-1\_N-2} & -C_{N-1\_N-2} - C_{N\_N-1} & C_{N\_N-1} \\ 0 & 0 & 0 & 0 & \cdots & 0 & C_{N\_N-1} & -C_{N\_N-1} \end{bmatrix}. \tag{77}$$

4.7.2 Application of the boundary conditions

The application of the boundary conditions allows to define the connections of the layers in series, such as:

$$p_1 \longleftrightarrow p_2 \tag{78}$$

$$q_2 \longleftrightarrow q_3 \tag{79}$$

$$p_3 \longleftrightarrow p_4 \tag{80}$$

$$q_4 \longleftrightarrow q_5 \tag{81}$$

$$\vdots \tag{82}$$

which results in the following set of conditions:

$$\begin{cases} U_{p1} = U_{p2} \\ U_{q2} = U_{q3} \\ U_{p3} = U_{p4} \\ U_{q4} = U_{q5} \\ \vdots \\ I_{p1} + I_{p2} = 0 \\ I_{q2} + I_{q3} = 0 \\ I_{p3} + I_{p4} = 0 \\ I_{q4} + I_{q5} = 0 \\ \vdots \end{cases} \tag{83}$$

These conditions can be imposed to the system by a series of rows and columns operations. A transformation from **ABCD** to **Y** makes the application of the boundary conditions easier, as it gathers all voltages in the input vector and all currents in the output vector.

$$\begin{bmatrix}\mathbf{U}_q\\ \mathbf{I}_q\end{bmatrix} = \begin{bmatrix}\mathbf{A} & \mathbf{B}\\ \mathbf{C} & \mathbf{D}\end{bmatrix}\begin{bmatrix}\mathbf{U}_p\\ \mathbf{I}_p\end{bmatrix} \longrightarrow \begin{bmatrix}\mathbf{I}_p\\ \mathbf{I}_q\end{bmatrix} = \begin{bmatrix}-\mathbf{B}^{-1}\mathbf{A} & \mathbf{B}^{-1}\\ \mathbf{C}-\mathbf{D}\mathbf{B}^{-1}\mathbf{A} & \mathbf{D}\mathbf{B}^{-1}\end{bmatrix}\begin{bmatrix}\mathbf{U}_p\\ \mathbf{U}_q\end{bmatrix} \tag{84}$$

The equality of voltages U_x and U_y is expressed by adding column y to column x, removing column y and replacing U_x by U_z. This voltage is an intermediate voltage of the series connection which is not relevant and will be eliminated later.
The fact that the sum of two currents I_x and I_y is equal to zero is expressed by adding row y to row x, removing row y and replacing I_x by 0.

4.7.3 Reduction of the system

The system is reduced by eliminating all intermediate voltages from the voltage vector and all zero entries from the current vector. This is done by first exchanging the intermediate voltages with the zero entries. The procedure to do so is explained in [17]. After the exchange, the columns corresponding to zero currents and the rows corresponding to intermediate voltages can simple be removed.

Eventually, we obtain an admittance description of the form:

$$\begin{bmatrix}\mathbf{I}_i\\ \mathbf{I}_o\end{bmatrix} = \begin{bmatrix}Y_{11} & Y_{12}\\ Y_{21} & Y_{22}\end{bmatrix}\begin{bmatrix}\mathbf{U}_i\\ \mathbf{U}_o\end{bmatrix} \tag{85}$$

4.7.4 Single-phase and three-phase connections

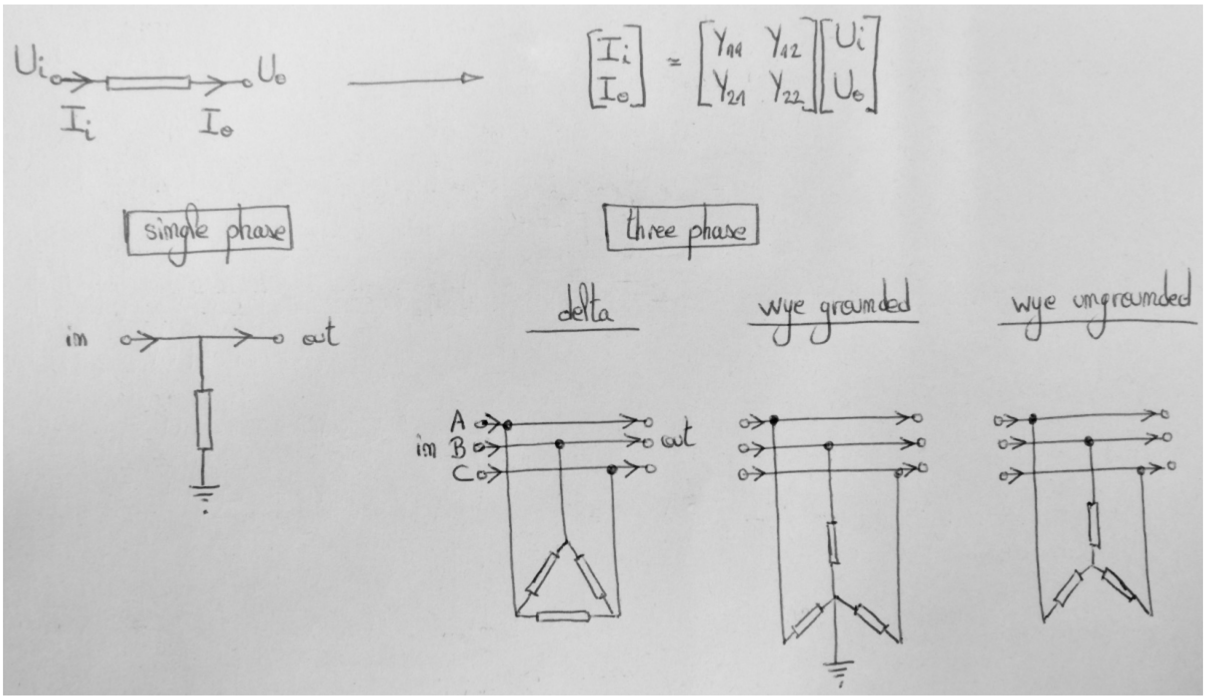


Figure 25: Single and three-phase connections

For all connections presented in Fig. 25, we obtain an ABCD representation in the form of:

$$\begin{bmatrix}\mathbf{U}_o\\ \mathbf{I}_o\end{bmatrix} = \begin{bmatrix}\mathbf{I} & \mathbf{0}\\ \mathbf{Y} & \mathbf{I}\end{bmatrix}\begin{bmatrix}\mathbf{U}_i\\ \mathbf{I}_i\end{bmatrix} \tag{86}$$

where $\mathbf{I}$ and $\mathbf{0}$ are the identity and zero matrices of adequate size; $\mathbf{U}$ and $\mathbf{I}$ are scalar for the single phase connection and vectors for the three-phase configurations, e.g.:

$$\mathbf{U}_o = \begin{bmatrix} U_{Ao} \\ U_{Bo} \\ U_{Co} \end{bmatrix} \tag{87}$$

Finally, we have the following, where Y_{11} and Y_{12} are as defined in Eq. 85:

- Single phase:

$$\mathbf{Y} = -Y_{11} \tag{88}$$

- Three-phase delta:

$$\mathbf{Y} = \begin{bmatrix} Y_{12} - Y_{11} & -Y_{12} & Y_{11} \\ Y_{11} & Y_{12} - Y_{11} & -Y_{12} \\ -Y_{12} & Y_{11} & Y_{12} - Y_{11} \end{bmatrix} \tag{89}$$

- Three-phase wye grounded:

$$\mathbf{Y} = \begin{bmatrix} -Y_{11} & 0 & 0 \\ 0 & -Y_{11} & 0 \\ 0 & 0 & -Y_{11} \end{bmatrix} \tag{90}$$

- Three-phase wye ungrounded:

$$\mathbf{Y} = \frac{1}{3} Y_{11} \begin{bmatrix} -2 & 1 & 1 \\ 1 & -2 & 1 \\ 1 & 1 & -2 \end{bmatrix} \tag{91}$$

5 Power flow

As the ABCD formulation is a linear representation of the power system, nonlinear descriptions of the components such as power converters must be linearized around an operating point. This operating point is determined by solving the power flow of the system.

In practice, the network is initialized using the optimal power flow tool [40] implemented as a Julia package, which can be found in the repository [41]. The package relies on the power flow models developed for the MatACDC simulator [42], which extends Matpower [43] AC power system models with the DC representations and with power converters.

As a result, the constructed power system is divided into AC and DC system and the converters. It contains AC and DC branches and buses, converters, generators, loads, shunts and storage elements. Components implemented in this simulator are represented using their equivalent models for the purpose of the power flow analysis.

5.1 AC and DC branches

AC and DC branches represent three-phase and DC connections between buses, respectively. Branches are grouped inside AC or DC grids (zones). AC branches are defined with parameters described in [43], while DC branches parameters are given in [42].

For the purpose of modeling the the system components, the model of the AC branch as provided in [43] is depicted in Fig. 26. Beside the shunt admittance $j\frac{b_c}{2}$, Julia package [41] supports the admittance as $\frac{g_c}{2} + j\frac{b_c}{2}$. The full expression for the AC admittance parameters is given by the equation:

$$\mathbf{Y}_{ac} = \begin{bmatrix} \left(y_s + \frac{g_c}{2} + j\frac{b_c}{2}\right)\frac{1}{\tau^2} & -\frac{y_s}{\tau\ \exp(-j\theta_{\text{shift}})} \\ -\frac{y_s}{\tau\ \exp(-j\theta_{\text{shift}})} & \left(y_s + \frac{g_c}{2} + j\frac{b_c}{2}\right) \end{bmatrix}. \tag{92}$$

It should be noted that the AC network is considered as balanced, and thus, all the components have diagonal matrix models, with equal elements on the matrix diagonal.

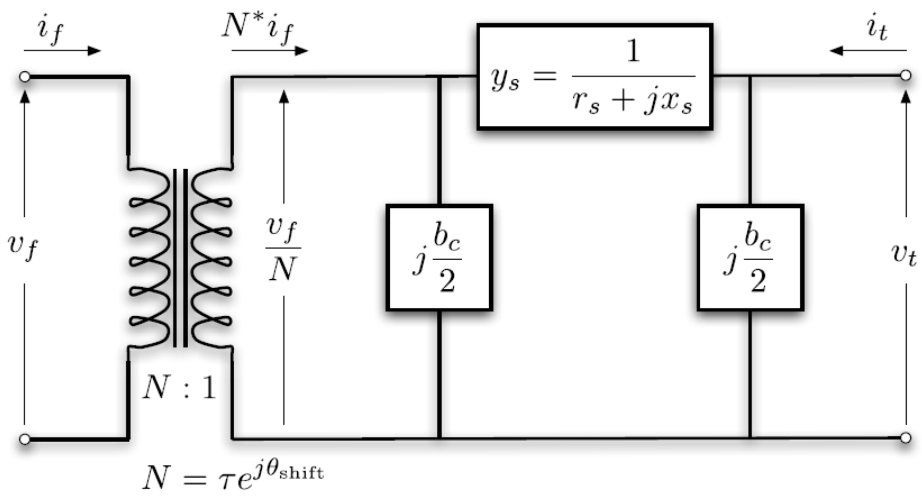


Figure 26: Matpower AC branch model [43].

A DC branch is modeled with its equivalent series resistance [42]. For the power flow calculation, some components are modeled as the AC and DC branches and their models are described in detail in this subsection.

5.1.1 Impedance

As described in Subsection 4.1, an impedance is modeled using ABCD parameters:

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{Z} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}. \tag{93}$$

In the case of the DC impedance, all matrices are of size 1×1, while three-phase impedances are of size 3×3.

The DC branch model is then given as $r = \Re\{Z\}$. The AC branch is modeled as an ideal transformer with $\tau = 1$ and $\theta_{\text{shift}} = 0$, and with $r_s = \Re\{\mathbf{Z}(j\omega)\langle 1,1\rangle\}$, $x_s = \Im\{\mathbf{Z}(j\omega)\langle 1,1\rangle\}$, $g_c = 0$ and $b_c = 0$.

5.1.2 Transformer

Since the transformer model considered in 4.2 cannot be easily represented as the model in Fig. 26,Y parameters are extracted from the ABCD parameters, as described in the Appendix in Eq. (20).

In the case of DC branches, since ABCD parameters are each of size 1×1 (i.e. scalars), the tap value can be determined as $\tau = \sqrt{\frac{A}{D}}$, while the series impedance is obtained as $r = \Re\{\frac{B}{\tau}\}$.

In the case of AC networks and three-phase transformers, using the assumption of the balanced system, the submatrices $\mathbf{Y}\langle 1:3, 1:3\rangle$, $\mathbf{Y}\langle 1:3, 4:6\rangle$, $\mathbf{Y}\langle 4:6, 1:3\rangle$ and $\mathbf{Y}\langle 4:6, 4:6\rangle$ are diagonal. Thus, it is sufficient to use a single diagonal value from each submatrix. Then, $\mathbf{Y}\langle 1,1\rangle = \left(y_s + \frac{g_c}{2} + j\frac{b_c}{2}\right)\frac{1}{\tau^2}$, $\mathbf{Y}\langle 1,4\rangle = \mathbf{Y}\langle 4,1\rangle = -\frac{y_s}{\tau\,\exp(-j\theta_{\text{shift}})}$ and $\mathbf{Y}\langle 4,4\rangle = \left(y_s + \frac{g_c}{2} + j\frac{b_c}{2}\right)$. The following expressions are derived:

$$\begin{aligned} \tau &= \sqrt{\tfrac{\mathbf{Y}\langle 4,4\rangle}{\mathbf{Y}\langle 1,1\rangle}}, \quad \theta_{\text{shift}} = 0, \\ y_s &= -\mathbf{Y}\langle 1,4\rangle\,\tau\,\exp(-j\theta_{\text{shift}}), \\ y_c &= \mathbf{Y}\langle 4,4\rangle - y_s, \\ r_s &= \Re\left\{\tfrac{1}{y_s}\right\}, \quad x_s = \Im\left\{\tfrac{1}{y_s}\right\}, \\ g_c &= \Re\{y_c\}, \quad b_c = \Im\{y_c\}. \end{aligned}$$

5.1.3 Transmission line

A transmission line (OHL, cable, cross-bonded cable or mixed OHL-cable) is represented using its nominal π-model depicted in Fig. 27, where

$$\begin{aligned} \mathbf{Z}(j\omega) &= \mathbf{Y}_c^{-1}\sinh(\mathbf{\Gamma}l), \\ \mathbf{Y}(j\omega) &= \mathbf{Y}_c\tanh(\mathbf{\Gamma}l). \end{aligned} \tag{94}$$

For the DC case, the shunt admittance is not considered, while the branch resistance is equal to $r = \Re\{Z(0)\}$.

For the balanced AC transmission line, the impedance and admittance matrices are diagonal. It can be chosen $Z(j\omega) = \mathbf{Z}(j\omega)\langle 1,1\rangle$ and $Y(j\omega) = \mathbf{Y}\langle 1,1\rangle$. Then, the AC branch model is given by:

$$\begin{aligned} \tau &= 0, \quad \theta_{\text{shift}} = 0, \\ r_s &= \Re\{Z(j\omega)\}, \quad x_s = \Im\{Z(j\omega)\}, \\ g_c &= \Re\{Y(j\omega)\}, \quad b_c = \Im\{Y(j\omega)\}. \end{aligned} \tag{95}$$

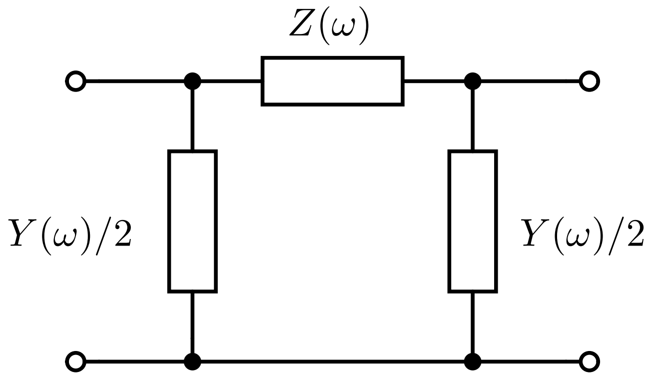


Figure 27: Nominal π-model of the transmission line.

5.2 Shunt components

Shunt reactors and capacitors are defined with their admittance value as $y = g_s + jb_s$ [43].

5.3 Generators

In this simulator, generators are represented as three-phase AC sources, and are defined as reference buses.

5.4 Power converter

A power converter is, in accordance with [42], modeled together with its phase reactor, filter and transformer. In order to match the constructed MMC model in Section 4.6, only the reactor is considered.

Losses of the converter are calculated in the form of $P_{loss} = a + bI_c + cI_c^2$. Since the switches are modeled as ideal, the model implementation supports only a constant value $c = \frac{R_{arm}}{2}$.

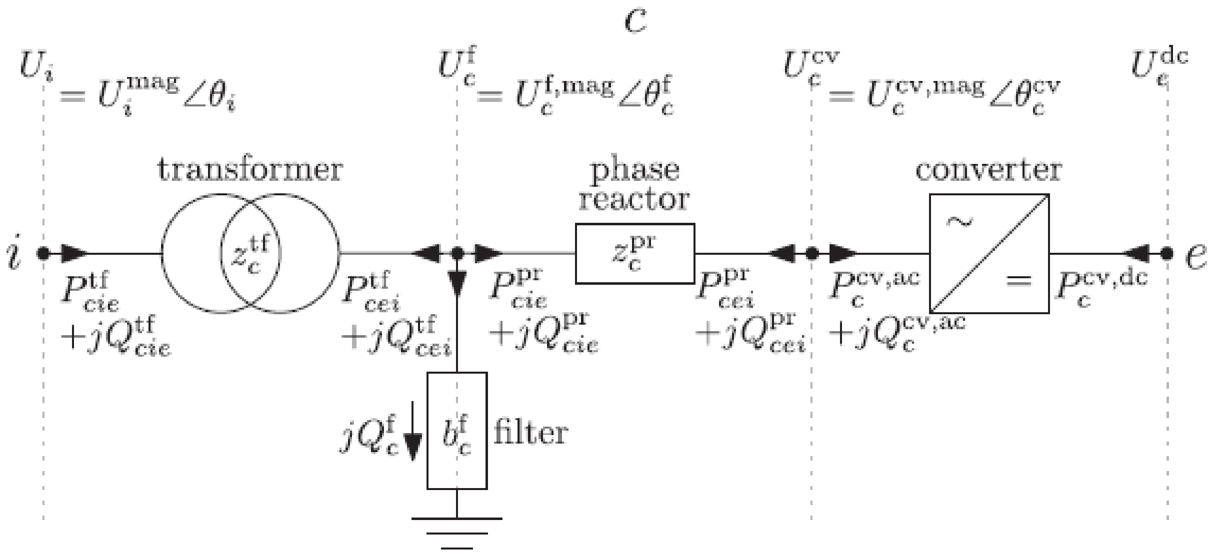


Figure 28: Power flow model of the power converter.

According to the realisation of the converter's controls, the parameters of the converter can be set as DC voltage controlling or the active power controlling converters.

6 Simulator impementation

The simulator is implemented in Julia programming language and it consists of multiple structures organised within a specific hierarchy. The top class consists of a structure named `Network`, which is "responsible" for the creation of the complete system. The structure `Network` consists of the sub-classes `Nets` and `Elements`. Each `Net` is a set of connections between elements, and the set `Elements` presents a collection of all the elements in the `Network` (system).

An `Element` can be made as a special power system component, or as a composite element, which can contain a number of components and their connections. The composite element is by default defined as a `Network`.

The simulator is implemented using the following Julia packages: SymEngine, LinearAlgebra, NLsolve and ForwardDiff [39] for symbolic and numerical calculations; DelimittedFiles and FileIO for the reading and writing in file; DataStructures and Parameters for data types support; Plots and LaTeXStrings for the visualisation. For the power flow are used packages PowerModels, PowerModelsACDC, Ipopt and JuMP.

6.1 Files organization

All package related files are organized in the folder `/src/Network`.

- `HVDCstability.jl` - generates the complete simulator package. It calls all the used packages and links all generated Julia files.
- `compat.jl` - Julia file with the definitions that implement the compatibility between Julia versions.
- `globals.jl` - defines all global constants and functions used inside the package (empty at the moment).
- `Network.jl` - generates the top structure `Network` and all its functions.
- `Components` - folder containing files with the component definitions:
  - `AbstractElement.jl` - generates the structure `Element`, which presents the type of the each component in the power system.
  - `element_types.jl` - contains paths to all defined components and it links them to the structure `Element.`
  - `converter` - folder containing converter definitions:
    - `controller.jl` - abstract structure `Controller`, which can be used for the desired controller definition. So far, only the derived `PI_controller` is implemented.
    - `converter.jl` - defines abstract type `Converter` and the set of functions that can be generally applied to the wide range of power converters' definitions.
    - `MMC.jl` - creates a specific type of the power converter named MMC and the functions that can be only applied to MMCs.
  - `impedance` - folder containing an impedance definition and the functions related to impedances:
    - `impedance.jl` - implements `Impedance` structure and its functions.
  - `shunt_reactor` - folder with the shunt reactor definitions:

    * `shunt_reactor.jl` - implementation of the component `Shunt_reactor` and its functions.

  - `source` - folder with source definitions:
    * `source.jl` - defines abstract type `Source` and the set of functions that can be generally applied to the wide range of voltage source definitions.
    * `dc_source.jl` - defines the functions for the DC voltage source.
    * `ac_source.jl` - defines the functions for the AC voltage source.
    * TO DO - Can be added current source definition.
  - `transformer` - folder containing transformer and autotransformer definitions:
    * `transformer` - defines structure `Transformer`, which can be single-phase and three-phase in YY and ΔY configuration. Can be extended with the different transformer implementations.
    * `autotransformer.jl` - implements `Autotransformer` structure that supports only one autotransformer implementation. For the more general model should be extended.
  - `transmission_line` - folder with transmission line descriptions:
    * `transmission_line.jl` - general `Transmission_line` structure and functions.
    * `overhead_line.jl` - structure `Overhead_line` derived from the `Transmission_line` structure, and the function specific to OHL.
    * `cable.jl` - structure `Cable` derived from the `Transmission_line` structure, and the function specific to cable.
    * `crossbonded_cable.jl` - structure `Crossbonded_cable` derived from the `Transmission_l` structure, and the function specific to cross-bonded cable.
    * `mixed_OHL_cable.jl` - structure `Mixed_OHL_cable` derived from the `Transmission_line` structure, and the function specific for this structure.
  - `tools` - folder with the tools for the `Element` structure:
    * `plot.jl` - implements bode plotting.
    * `abcd_parameters.jl` - various functions used for ABCD parameter calculations and manipulations.
    * `kron.jl` - Kron elimination for ABCD and Y parameters.
    * `tools.jl` - merges all files inside `tools` folder in order to create good file linking. Every time the new tool is generated, its path has to be added to this file.
  - `Solvers` - folder containing power system solving functionalities:
    * `solvers.jl` - file containing path of the each solving possibility. It does linking to the complete Julia package.
    * `determine_impedance.jl` - implementation of the impedance determination function.
    * `stability.jl` - implementation of the function used to determine feedback transfer function of the interconnection between power converter and the rest of the network.
    * `make_abcd.jl` - makes ABCD representation of the subnetwork.
    * `make_y.jl` - makes Y representation of the subnetwork.
- `GUI` - Place for the GUI, to be implemented.

Package test files in folder `/test/`:

- `runtests.jl` - initialization of the packages used for tests and setting for the testing procedure.
- `tests.jl` - collection of the tests to be run together with the comparing values.
- `tests` - folder containing various tests.

Documentation which is generated automatically in folder `/docs`:

- `make.jl` - file that defines how the documentation should be translated from `.md` files to HTML. Also defines how HTML files should be connected and in which `.git` account deployed.
- `README.md` - read me file for the github.
- `src` - folder containing `.md` files with the descriptions of the functions which are to be added in the HTML documentation.

Additional files:

- `gen_pr.jl` - generates `Project.toml` file from the written `REQUIRE` file, which contains the names of packages used inside this package.
- `Project.toml` - automatically generated by calling `gen_pr.jl`. `REQUIRE` - all packages called inside this package have to be named inside this file, each in the new line (for example package: `PowerModels`, etc.). See the file.
- `.travis.yml` - Travis file used for checking implemented package functionalities by `travis-ci.com`.
- `LICENSE`
- `HVDCstability.pdf` - user and development manual.
- `README.md` - github read me file.
- `.gitignore` - files and folders that should not be uploaded to github.

6.2 Network structure

As introduced, a `Network` consists of a structure defining the collection of all components and their interconnections. The components are defined as type `Element` and the nodes of the system are of type dictionary. A Symbol is used to refer to each node, which consists of the set of element pins to which it is connected.

It should be noted that the designator `gnd` is chosen as an universal symbol for the ground. It is not necessary to initialise it inside the network, since it is generated directly from the provided designator. In order to provide decoupling between different circuit parts, symbols `gnd` with a desirable suffix can be chosen (e.g. `gnd1`, `gnd2`, etc.). Using the same designator `gnd` for multiple components means that those components are short connected through the ground.

A `Network` is initialized with the macro `@network` as follows:

```
@network begin #= ... =# end
```

where `#= ... =#` denotes the set of expressions describing the network. It provides a simple domain-specific language to describe networks. The begin/end block can hold element definitions of the form `refdes = elementfunc(params)`, where `refdes` is the user-defined symbol that refers to the constructed element, `elementfunc` is the function called with the list of parameters `params` for the creation of the element. Connection specifications are in the following form. Note that == can also be used in place of ⟷.

```
refdes[pin1] ⟷ refdes2[pin2] ⟷ MyNode
```

The part ⟷ MyNode is optional, but it gives an opportunity to give a symbolic name to the specified connection.
Examples: The following code can be used to construct a network with an impedance (a pure resistance in this case) and a voltage source.

```
net = @network begin
    src = dc_source(V = 5)
    r = impedance(z = 1000, pins = 1)
    src[1.1] ⟷ r[1.1]
    src[2.1] ⟷ r[2.1]
end
```

Alternatively, connection specifications can be given after an element specification, separated by commas. In that case, the mention of `refdes` may be omitted, defaulting to the current element, as follows:

```
@network begin
    src = dc_source(V = 5)
    r = impedance(z = 1000, pins = 1), src[1.1] ⟷ [1.1], src[2.1] ⟷ [2.1]
end
```

Finally, a connection endpoint may simply be in the form of the symbolic `netname`, to connect to a named net. (Such named nets are created as needed.)

```
@network begin
    src = dc_source(V = 5), [2.1] ⟷ gnd
    r = impedance(z = 1000, pins = 1), [1.1] ⟷ src[1.1], [2.1] ⟷ gnd
end
```

6.2.1 Add and delete components

Different power system components are added as type `Element` in the `Network`. Functions that add and delete components are as follows.

- `add!(n::Network, elem::Element)`

  Adds the element `elem` to the network `n`, creating and returning a new, unique reference designator for the element (not for the network, whose designator remains the same). The pins of the element are left disconnected.

- `add!(n::Network, designator::Symbol, elem::Element)`

  Adds the element `elem` to the network `n` with the reference designator `designator`, leaving its pins unconnected. If the network already contains an element with the same designator, it is removed first.

- `delete!(n::Network, designator::Symbol)`

  Deletes the element that corresponds to this designator from the network `n` (disconnecting all its pins). The element is completely removed from the memory.

- `connect!(n::Network, pins::UnionSymbol,TupleSymbol,Any...)`

  Connects the given pins (or named nets - set of nodes) to each other in the network `n`. Named nets are given as `Symbols`, pins are given as `Tuple{Symbols,Any}`, where the first entry is the reference `designator` of an element in `n`, and the second entry is the pin name. For convenience, the latter is automatically converted to a `Symbol` as needed.

- `disconnect!(n::Network, p::TupleSymbol,Symbol)`

  Disconnects the given pin `p` from anything else in the network `n`. The pin is given as a `Tuple{Symbols,Any}`, where the first entry is the reference `designator` of an element in `n`, and the second entry is the pin name. For convenience, the latter is automatically converted to a `Symbol` as needed. Note that if e.g. three pins `p1`, `p2`, and `p3` are connected then `disconnect!(n, p1)` will disconnect `p1` from `p2` and `p3`, but leave `p2` and `p3` connected to each other.

- `composite_element(subnet::Network, input_pins::Array{Any}, output_pins::Array{Any})`

  Creates a network element from the (sub-)network `net`. The `input_pins` and `output_pins` define input and output nodes of the element.

Example: Using functions `add!` and `connect!`.

```
network = Network()
add!(network, :r, impedance(z = 1e3, pins = 1))
add!(network, :src, dc_source(V = 5))
connect!(network, (:src, 2.1), (:r, 2.1), :gnd) # connect to gnd node
```

Example: Creating element from the network.

```
my_network = @network begin
   r1 = impedance(z = 10e3, pins = 1)
   r2 = impedance(z = 10e3, pins = 1), [1.1] == r1[2.1]
   c = impedance(z = 10e3, pins = 1), [1.1] == r2[1.1], [2.1] == r2[2.1]
   src = dc_source(V = 5), [1.1] == r1[1.1], [2.1] == r2[2.1]
end
composite_element(my_network, Any[(:r2, Symbol(1.1))], Any[(:r2, Symbol(2.1))])
```

6.2.2 Additional checks

After the construction of a network `n`, at the end of the macro `n = @network begin #= ... =# end`, the program calls the following two functions to check if the network is well connected and to construct the ABCD parameter equivalents for all the components. The following functions are only called internally.

- `check_lumped_elements(network :: Network)`

  This function checks if the network is well connected, e.g. that there is no lumped (i.e. disconnected) pins in the network. If the network is not well connected, the user receives an error message.

Note: If in the future use and development this function is not needed any more, it can be removed from the calling at the end of `@network` function. The directive that should be deleted then is `push!(ccode.args, :(check_lumped_elements(network)))`

- `power_flow(network :: Network)`

  This function is called in order to set the operating points of the converters. It generates a dictionary with the syntax used in [41] and it solves the power flow problem using the Julia package PowerModelsACDC [41]. The results are used to update the operating point of the converters.

  If there is not a converter in the network, the function is not called.

6.2.3 Impedance determination and stability assessment

- `function determine_impedance(network::Network; input_pins :: Array{Any}, output_pins :: Array{Any}, elim_elements :: Array{Symbol}, omega_range = (-3, 5, 100), , parameters_type = :ABCD)`

  For the selected multiport, it is possible to determine the impedance using the procedure described in Section 2. A multiport is depicted in Fig. 2.

  Input or outputs pins can be connected to some elements, which should not be considered for the impedance estimation. Those elements are listed as symbols in `elim_elements`.

  The function generates the impedance as seen from the port defined by input and output pins. The impedance is calculated numerically at each frequency point along a use-defined frequency range. The number of frequency points is user-defined or else set to 1000 by default.

  Additionally, it can be chosen how the network will be solved: using ABCD or Y parameters. Defining `parameters_type` as `:ABCD` sets the solving using ABCD parameters, while setting it to `:Y` defines usage of Y parameters. In these cases function `determine_impedance` internally calls either function `make_abcd` either `make_y`.

  Example: The specification for the determination of an impedance is given in the example, where the network consists of a DC voltage source and a cable.

```
net = @network begin
    vs = dc_source(V = 500e3)
    c = cable(length = 100e3, positions = [(0,1)], earth_parameters = (1,1,1),
    C1 = Conductor(rₒ = 24.25e-3, ρ = 1.72e-8),
    C2 = Conductor(rᵢ = 41.75e-3, rₒ = 46.25e-3, ρ = 22e-8),
    C3 = Conductor(rᵢ = 49.75e-3, rₒ = 60.55e-3, ρ = 18e-8, μᵣ = 10),
    I1 = Insulator(rᵢ = 24.25e-3, rₒ = 41.75e-3, ϵᵣ = 2.3),
    I2 = Insulator(rᵢ = 46.25e-3, rₒ = 49.75e-3, ϵᵣ = 2.3),
    I3 = Insulator(rᵢ = 60.55e-3, rₒ = 65.75e-3, ϵᵣ = 2.3))
    vs[1.1] ⟷ c[1.1] ⟷ Node1
    vs[2.1] ⟷ c[2.1] ⟷ gnd
end
```

  To determine the impedance visible from the voltage source `vs`, the following command should be called:

```
imp, omega = determine_impedance(net, elim_elements = [:vs],
        input_pins = Any[:Node1], output_pins = Any[:gnd],
        omega_range = (-1,6,10000))
```

The impedance is determined inside network `net`, from the element `vs` (without including the element `vs`) and the port defined with `input_pins` as array consisting of `Node1` and the `output_pins` containing array with `gnd`. The impedance is estimated over the frequency range 10^{-1} [rad/s] to 10^{6} [rad/s] in 10000 points.

The function returns two complex arrays: the first array being the impedance array and the second array containing the angular frequencies at which the impedance is calculated.

- `function check_stability(net :: Network, mmc :: Element, direction :: Symbol = :dc)`

  This function determines two impedances inside the network, from which it forms the feedback transfer function. It allows "cutting" the power network next to the converter on its dc or ac side (determined by `direction` parameter). Afterwards, it is chacked the impedance Z_{conv} obtained by "looking" in the converter and the other one Z_h from the converter to the remaining of the circuit.

  Using previous two impedances, the feedback transfer function is estimated as $Z_h Y_{conv}$.

  The impedances are calculated for the angular frequencies whose range is defined by `omega_range`.

6.2.4 Saving and plotting data

Any component in the network has its own data representation. Component data can be saved and plotted using the following functions.

- `save_data(element :: Element, file_name :: String; omega_range = (-3, 5, 1000), scale = :log)`

  This function saves component specific data in csv textual file. The data is saved as frequency dependent. Thus, an additional parameter `omega_range` provides the possibility to manually add the frequency range for saving data. The scale can be given as logarithmic with `:log` and linear with `:lin`.

- `plot_data(element :: Element; omega_range = (-3, 5, 1000), scale = :log)`

  This function is a component-defined function for plotting data, it differs from component to component. It plots the component defined data with the desired frequency range. An additional parameter `omega_range` provides the possibility to manually add the frequency scale for saving data. The scale can be given as logarithmic with `:log` and linear with `:lin`.

- `bode(transfer_function :: Array{Any}; omega_range = (-3, 5, 100), titles :: Array{String} = [""], omega = [], axis_type = :loglog, save_data = false)`

  This function is used for plotting the transfer function or frequency dependent data as a bode plot. The function takes frequency points in the form of `omega_range` or as mapped values `omega`. For a nice display, the labels can be given as strings by calling the parameter `titles`.

  It can be specified how to plot data:

  - `:loglog` - logarithmic frequency scale and logarithmic impedance in dB;
  - `:linlog` - linear frequency scale and logarithmic impedance in dB;
  - `:loglin` -logarithmic frequency scale and linear impedance (magnitude);

– `:logrealimag` - logarithmic frequency scale and real/imaginary part;
– `:linrealimag` - linear frequency scale and real/imaginary part.

Data can be saved by setting `save_data = true`.

6.3 Component definition

6.3.1 Impedance

An impedance is constructed as an element by calling the function: `impedance(;z :: Union{Int, Float64, Basic, Array{Basic}} = 0, pins :: Int = 0)`.

The function creates an impedance with the specified number of input/output pins. The number of input pins is equal to the number of output pins. The impedance expression `exp` has to be given in ohms and can have both numerical and symbolic value (example: `z = s-2`). Pins are named: 1.1, 1.2, ..., 1.n and 2.1, 2.2, ..., 2.n, where n is a number of input/output pins.

In the case of a 1×1 impedance, the parameter `z` has only one value. Example: `impedance(z = 1000, pins = 1)`.

If the impedance is multiport (i.e. if it has a number of input pins and output pins greater than 1), then its value is given as an array `[val]` with one, n or $n \times n$ number of values. When the arrway as length 1, then the impedance is defined with only diagonal nonzero values equal to val. When the array has a length equal to the number of pins, the impedance has only diagonal nonzero values equal to the values in the array. When the array is of size $n^2 \times 1$, the impedance matrix is of size $n \times n$ and all its values are defined accordingly to the array.

Examples:

```
# 3×3 impedance with diagonal values equal to s
impedance(z = [s], pins = 3)
# 3×3 impedance with diagonal values equal 2, s, 0.5s, respectively
impedance(z = [2,s,s/2], pins = 3)
# 2×2 impedance with all values defined
impedance(z = [1,s,3,4], pins = 2)
```

In this example, `s` is the Laplace parameter. The impedance can be added in the form of a transfer function, which can be either a polynomial or a nonlinear function of `s` and even a noninteger powers of `s`, e.g. a pure time delay may be represented by an exponential function of `s`.

The code below can be used to define the network with its impedances, as depicted in Fig. 29, and to generate its bode plot.

```
using SymEngine

s = symbols("s")
net = @network begin
    vs = dc_source(V = 5)
    z1 = impedance(z = s+2, pins = 1)
    z2 = impedance(z = s, pins = 1)
    z3 = impedance(z = s, pins = 1)

    vs[1.1] ⟷ z1[1.1] ⟷ Node1
    z1[2.1] ⟷ z2[1.1] ⟷ z3[1.1]
```

```
    vs[2.1] ⟷ z2[2.1] ⟷ z3[2.1] ⟷ gnd
end
imp, omega = determine_impedance(net, elim_elements = [:vs],
                            input_pins = Any[:Node1], output_pins= Any[:gnd])
bode(imp, omega = omega)
```

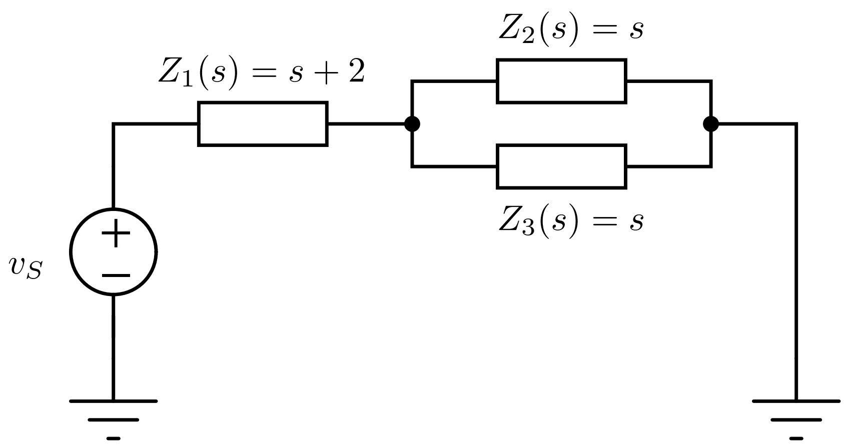


Figure 29: Example of the network with impedances.

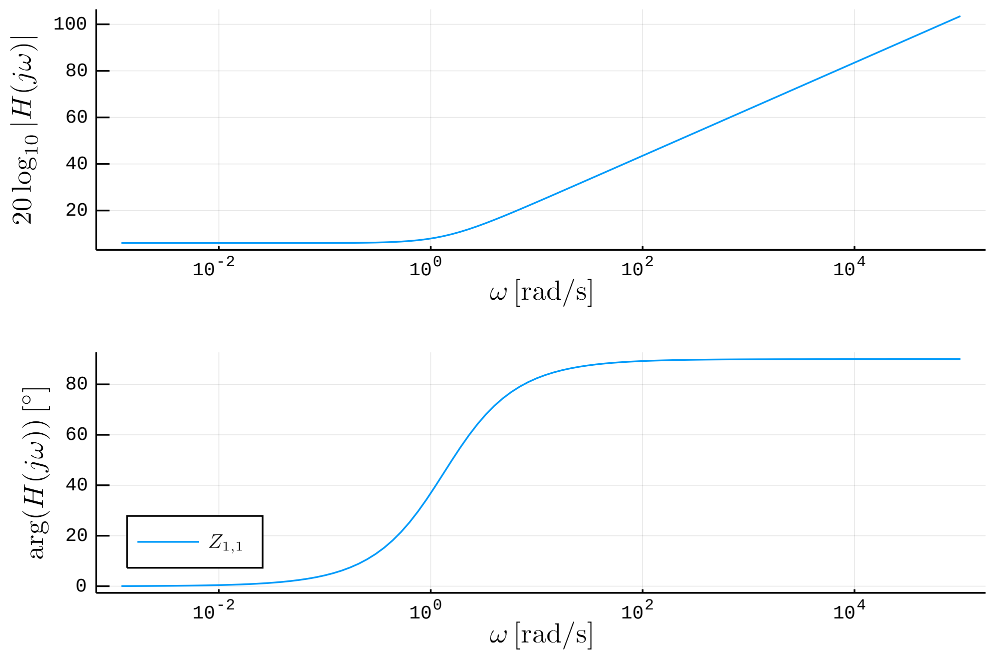


Figure 30: Magnitude and a phase of the equivalent impedance.

The equivalent impedance magnitude and phase are depicted in Fig. 30, which corresponds to the fact that the impedance seen from the DC source is equal to $2 + 1.5s$.

6.3.2 Transformer

A transformer element is created using the function `transformer(;args...)`, which creates a dc transformer, a single phase transformer or a three-phase transformer in `YY` or Δ`Y` configuration.

The component is defined using the following structure.

```
@with_kw mutable struct Transformer
    value :: Array{Basic} = []          # ABCD value
```

```
    pins :: Int = 1                      # marks single or three phase
    organization :: Symbol = :YY         # three phase organization (:YY or :ΔY)

    ω :: Union{Int, Float64} = 2*π*50 # rated frequency in [Hz]
    V₁ᵒ :: Union{Int, Float64} = 0      # open circuit primary voltage [V]
    V₁ˢ :: Union{Int, Float64} = 0      # short circuit primary voltage [V]
    I₁ᵒ :: Union{Int, Float64} = 0      # open circuit primary current [V]
    I₁ˢ :: Union{Int, Float64} = 0      # short circuit primary current [V]
    P₁ᵒ :: Union{Int, Float64} = 0      # open circuit losses on primary side [W]
    P₁ˢ :: Union{Int, Float64} = 0      # short circuit losses on primary side [W]
    V₂ᵒ :: Union{Int, Float64} = 0      # open circuit secondary voltage [V]
    V₂ˢ :: Union{Int, Float64} = 0      # short circuit secondary voltage [V]

    n  :: Union{Int, Float64} = 0        # turn ratio
    Lₚ :: Union{Int, Float64} = 0       # primary side inductance [H]
    Rₚ :: Union{Int, Float64} = 0       # primary side resistance [Ω]
    Rₛ :: Union{Int, Float64} = 0       # secondary side resistance [Ω]
    Lₛ :: Union{Int, Float64} = 0       # secondary side inductance [H]
    Lₘ :: Union{Int, Float64} = 0       # magnetising inductance [H]
    Rₘ :: Union{Int, Float64} = 0       # magnetising resistance [Ω]
    Cₜ :: Union{Int, Float64} = 0       # turn-to-turn capacitance [F]
    Cₛ :: Union{Int, Float64} = 0       # stray capacitance [F]
end
```

The pins are defined as : 1.1, 2.1 for single phase transformers and as: 1.1, 1.2, 1.3, 2.1, 2.2, 2.3 for three-phase transformers.
Example: For a circuit containing a single-phase transformer and an inductive load, the circuit definition is given in the following listing.

```
s = symbols("s")
net = @network begin
    vs = dc_source(V = 5e3)
    t = transformer(V₁ᵒ = 2.4e3, V₁ˢ = 51.87, V₂ᵒ = 240, P₁ᵒ = 171.1, P₁ˢ = 642.1,
                    I₁ᵒ = 0.48, I₁ˢ = 20.83, Cₛ = 12e-6, Cₜ = 7e-6)
    z = impedance(pins = 1, z = 25e-3*s)

    vs[1.1] ⟷ t[1.1] ⟷ Node1
    t[2.1] ⟷ z[1.1]
    vs[2.1] ⟷ z[2.1] ⟷ gnd
end
```

The impedance seen from the voltage source is depicted in Fig. 31.

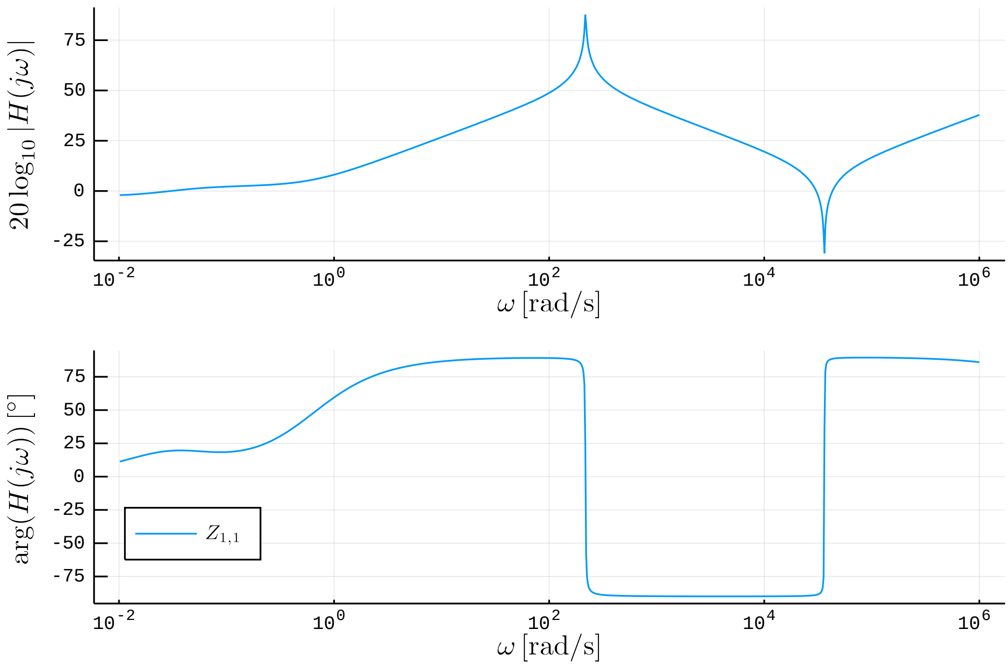


Figure 31: Magnitude and a phase of the equivalent impedance of the network containing a transformer.

6.3.3 Overhead line

An overhead line is defined with a structure and added as a field `element_value` inside an Element. It is called with the function: `overhead_line(;args...)`.

The function `overhead_line` generates the element `elem` with the `element_value` of the type `Transmission_line`. Arguments should be given according to the `Overhead_line` fields:

1. length - line length [m]

2. conductors - defined in the following structure:

```
struct Conductors
    # number of bundles (phases)
    nᵇ :: Int = 1
    # number of subconductors per bundle
    nˢᵇ :: Int = 1
    # height above the ground of the lowest bundle  [m]
    yᵇᶜ :: Union{Int, Float64}  = 0
    # vertical offset between the bundles [m]
    Δyᵇᶜ :: Union{Int, Float64} = 0
    # horizontal offset between the lowest bundles  [m]
    Δxᵇᶜ :: Union{Int, Float64} = 0
    # horizontal offset in group of bundles    [m]
    Δx̃ᵇᶜ :: Union{Int, Float64} = 0
    # sag offset    [m]
    dˢᵃᵍ :: Union{Int, Float64} = 0
    # subconductor spacing (symmetric)  [m]
    dˢᵇ :: Union{Int, Float64} = 0
    # conductor radius  [m]
    rᶜ :: Union{Int, Float64} = 0
    # DC resistance for the entire conductor [Ω/m]
    Rᵈᶜ :: Union{Int, Float64} = 0
    # shunt conductance
    gᶜ :: Union{Int, Float64} = 1e-11
    # relative conductor permeability
    μᵣᶜ :: Union{Int, Float64} = 1
    # add absolute positions manually
    positions :: Tuple{Vector{Union{Int, Float64}},
                    Vector{Union{Int, Float64}}} = ([],[])
    # organization can be :flat, :vertical, :delta, :concentric, :offset
    organization :: Symbol = Symbol()
end
```

3. groundwires - defined in the following structure:

```
struct Groundwires
    # number of groundwires (typically 0 or 2)
    nᵍ :: Int = 0
    # horizontal offset between groundwires [m]
    Δxᵍ :: Union{Int, Float64} = 0
    # vertical offset between the lowest conductor and groundwires  [m]
    Δyᵍ :: Union{Int, Float64} = 0
```

```
    # ground wire radius  [m]
    rᵍ :: Union{Int, Float64} = 0
    # sag offset [m]
    dᵍˢᵃᵍ ::  Union{Int, Float64} = 0
    # groundwire DC resistance [Ω/m]
    Rᵍᵈᶜ :: Union{Int, Float64} = 0
    # relative groundwire permeability
    μᵍᵣ :: Union{Int, Float64} = 1
    # add absolute positions manually
    positions :: Tuple{Vector{Union{Int, Float64}},
                 Vector{Union{Int, Float64}}} = ([],[])
    eliminate :: Bool = true
end
```

4. earth_parameters - with default value (1,1,1) and defining $(\mu_r, \epsilon_r, \rho)$ in units of ([], [], [Ωm])

Example:

```
overhead_line(length = 227e3, earth_parameters = (1,1,100),
    conductors = Conductors(nᵇ = 2, nˢᵇ = 2, organization = :flat,
    Rᵈᶜ = 0.06266, rᶜ = 0.01436, yᵇᶜ = 27.5, Δxᵇᶜ = 11.8, dˢᵇ = 0.4572, dˢᵃᵍ = 10),
    groundwires = Groundwires(nᵍ = 2, Δxᵍ = 6.5, Δyᵍ = 7.5, Rᵍᵈᶜ = 0.9196,
    rᵍ = 0.0062, dᵍˢᵃᵍ = 10))
```

The short-circuit impedance for the transmission line defined in the previous listing is plotted in Fig. 32.

6.3.4 Cable

A cable is defined with a structure and added as a field `element_value` inside an Element. It is called with the function: `cable(;args...)`.

The function `cable()` generates the element `elem` with the `element_value` of the type `Cable`. Arguments should be given according to the `Cable` fields:

1. length - length of the cable in [m]

2. earth_parameters - with default value (1,1,1) and defining $(\mu_r\_earth, \epsilon_r\_earth, \rho\_earth)$ in units of ([], [], [Ωm]), i.e. ground (earth) relative premeability, relative permittivity and ground resistivity

3. conductors - dictionary with the key symbol being: C1, C2, C3 or C4, and the value given with the structure `Conductor`. If the sheath consists of a metallic screen and a sheath, then screen must be added with a key symbol SC and a sheath with key symbol C2.

```
struct Conductor
    rᵢ :: Union{Int, Float64} = 0          inner radius
    rₒ :: Union{Int, Float64} = 0           # outer radius
    ρ  :: Union{Int, Float64} = 0           # conductor resistivity [Ωm]
    μᵣ  :: Union{Int, Float64} = 1          # relative permeability

    A :: Union{Int, Float64} = 0            # nominal area
end
```

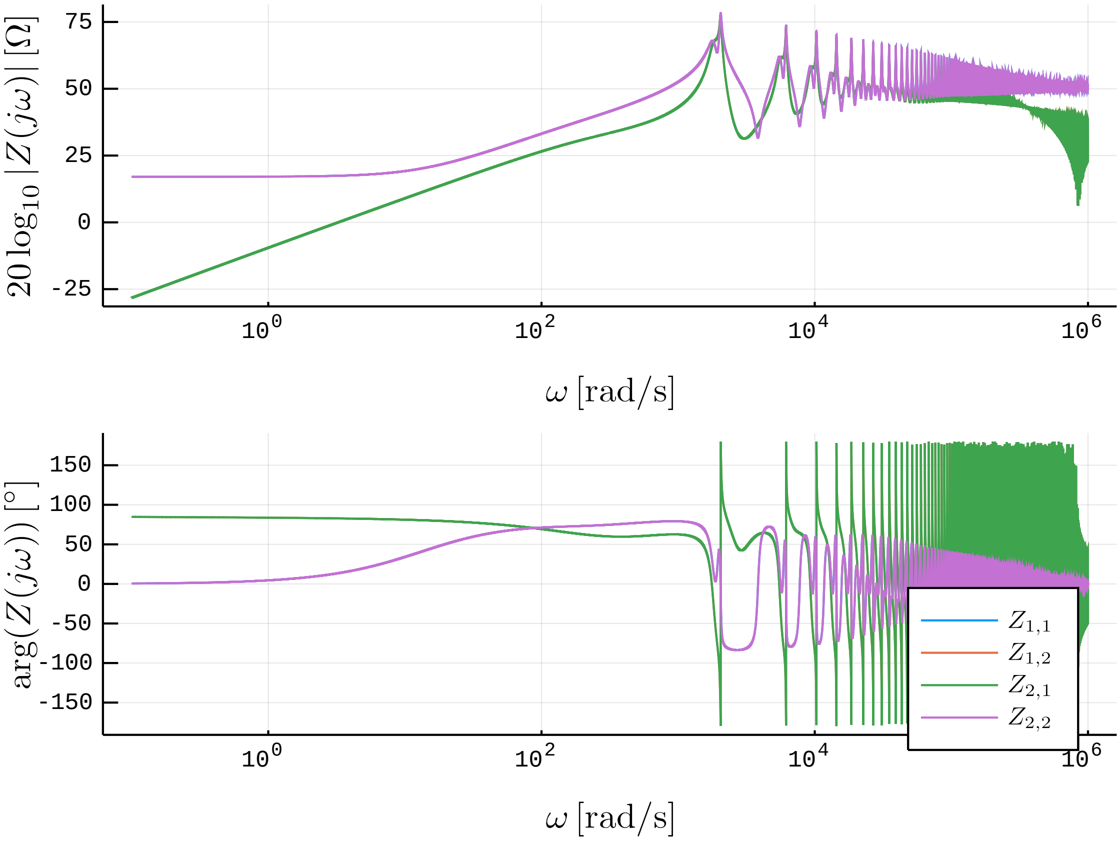


Figure 32: Transmission line example.

4. insulators - a dictionary with as key symbols: I1, I2, I3 and I4, and the value given with the structure `Insulator`. For the 2nd insulator, the semiconducting layers can be added by specifying outer radius of the inner semiconducting layer and inner radius of the outer semiconducting layer.

```
struct Insulator
    rᵢ :: Union{Int, Float64} = 0              # inner radius
    rₒ :: Union{Int, Float64} = 0              # outer radius
    ϵᵣ :: Union{Int, Float64} = 1              # relative permittivity
    μᵣ :: Union{Int, Float64} = 1              # relative permeability

    # inner semiconductor outer radius
    a :: Union{Int, Float64} = 0
    # outer semiconductor inner radius
    b :: Union{Int, Float64} = 0
end
```

5. positions - given as an array in (x,y) format

6. type - symbol representing aerial or underground cable

7. configuration - symbol with two possible values: coaxial (default) and pipe-type

8. eliminate - indicator weather the shaeth and armor layers should be grounded

Example:

```
cable(length = 100e3, positions = [(0,1)],
    C1 = Conductor(rₒ = 24.25e-3, ρ = 1.72e-8),
    C2 = Conductor(rᵢ = 41.75e-3, rₒ = 46.25e-3, ρ = 22e-8),
    C3 = Conductor(rᵢ = 49.75e-3, rₒ = 60.55e-3, ρ = 18e-8, μᵣ = 10),
    I1 = Insulator(rᵢ = 24.25e-3, rₒ = 41.75e-3, ϵᵣ = 2.3),
    I2 = Insulator(rᵢ = 46.25e-3, rₒ = 49.75e-3, ϵᵣ = 2.3),
    I3 = Insulator(rᵢ = 60.55e-3, rₒ = 65.75e-3, ϵᵣ = 2.3))
```

A bode plot is provided in Fig. 33 for the short-connected cable defined in the previous listing with a length of 100 km.

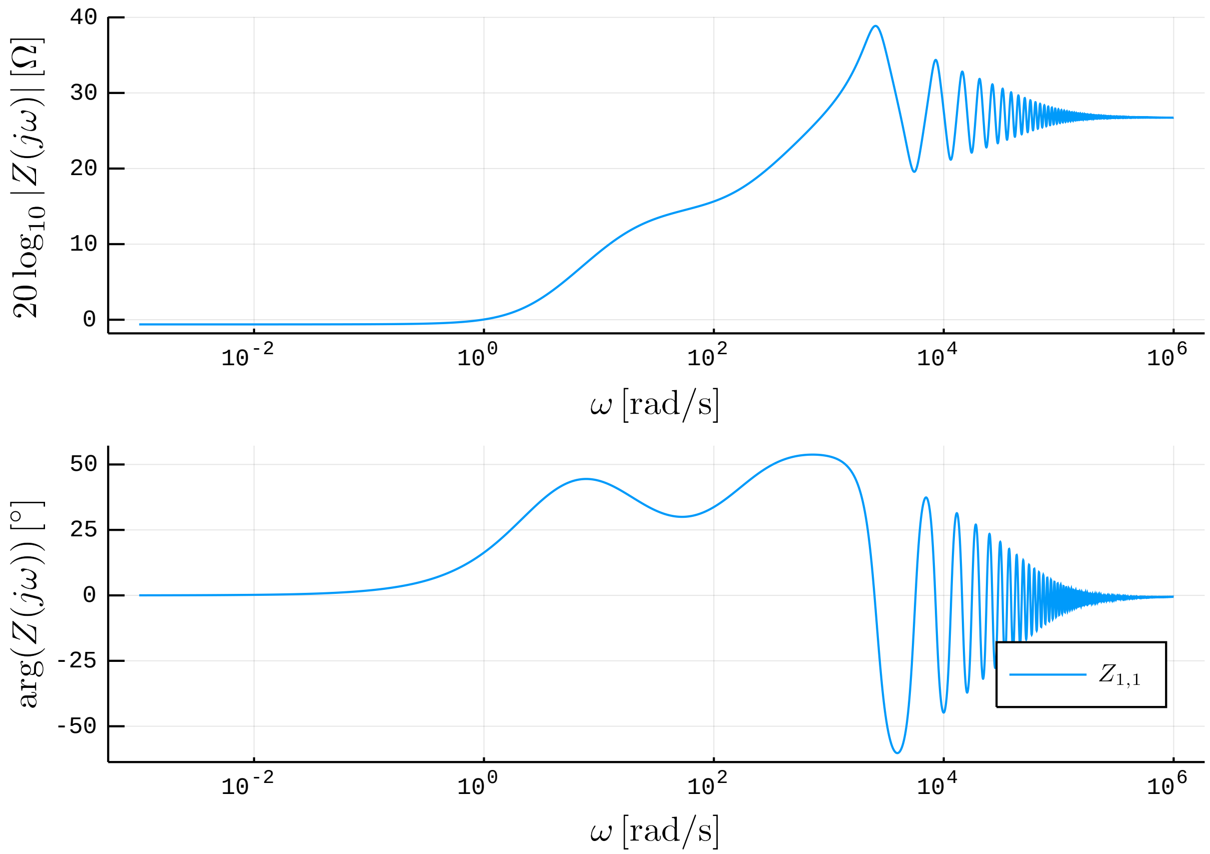


Figure 33: Cable example for line length 100 km.

6.3.5 MMC

An MMC is implemented as an `Element` using the function `mmc(;args...)`. The field `element_value` of the element is defined according to the following structure:

```
ω₀ :: Union{Int, Float64} = 100*π # nominal angular frequency

P :: Union{Int, Float64} = -10              # active power [MW]
Q :: Union{Int, Float64} = 3                # reactive power [MVA]
P_dc :: Union{Int, Float64} = 100           # DC power [kW]
P_min :: Union{Float64, Int} = -100         # min active power output [MW]
P_max :: Union{Float64, Int} = 100          # max active power output [MW]
Q_min :: Union{Float64, Int} = -50          # min reactive power output [MVA]
Q_max :: Union{Float64, Int} = 50           # max reactive power output [MVA]
```

```
θ :: Union{Int, Float64} = 0
Vm :: Union{Int, Float64} = 333             # AC voltage, amplitude [kV]
Vdc :: Union{Int, Float64} = 640            # DC-bus voltage [kV]

Larm :: Union{Int, Float64}  = 50e-3        # arm inductance [H]
Rarm :: Union{Int, Float64}  = 1.07         # equivalent arm resistance [Ω]
Carm :: Union{Int, Float64}  = 10e-3        # capacitance per submodule [F]
N :: Int = 400                              # number of submodules per arm

Lr :: Union{Int, Float64}  = 60e-3          # inductance of the phase reactor [H]
Rr :: Union{Int, Float64}  = 0.535          # resistance of the phase reactor [Ω]

# used inside the functions
controls :: OrderedDict{Symbol, Controller} = OrderedDict{Symbol, Controller}()
equilibrium :: Array{Union{Int, Float64}} = [0]
A :: Array{Union{Int, Float64}} = [0]
B :: Array{Union{Int, Float64}} = [0]
C :: Array{Union{Int, Float64}} = [0]
D :: Array{Union{Int, Float64}} = [0]
```

The constructed MMC has two pins on the AC side: 2.1 and 2.2, and one pin on its DC-side: 1.1 and 2.2. The component is described using two ABCD parameters with a matrix of size 4×4.

The controls are defined as `PI_control` and the keyword `occ` is for output current control, `ccc` for circulating current control, `zcc` for zero current control, `power` for active and reactive current control and `energy` for zero energy control.

Example:

The following example demonstrates the implementation of an MMC with output current control, circulating current control and power and energy controls. It is assumed that $P = 1000$ MW, $Q = 0$, $V_m = 320$ kV and $\theta = 0$.

```
mmc(energy = PI_control(Kp = 120, Ki = 400),
    occ = PI_control(ζ = 0.7, bandwidth = 1000),
    ccc = PI_control(ζ = 0.7, bandwidth = 300),
    zcc = PI_control(ζ = 0.7, bandwidth = 300),
    power = PI_control(Kp = 2.0020e-07, Ki = 1.0010e-04))
```

The admittances of the MMC can be displayed using the function `plot_data` and are presented in Figs. 34, 35 and 36.

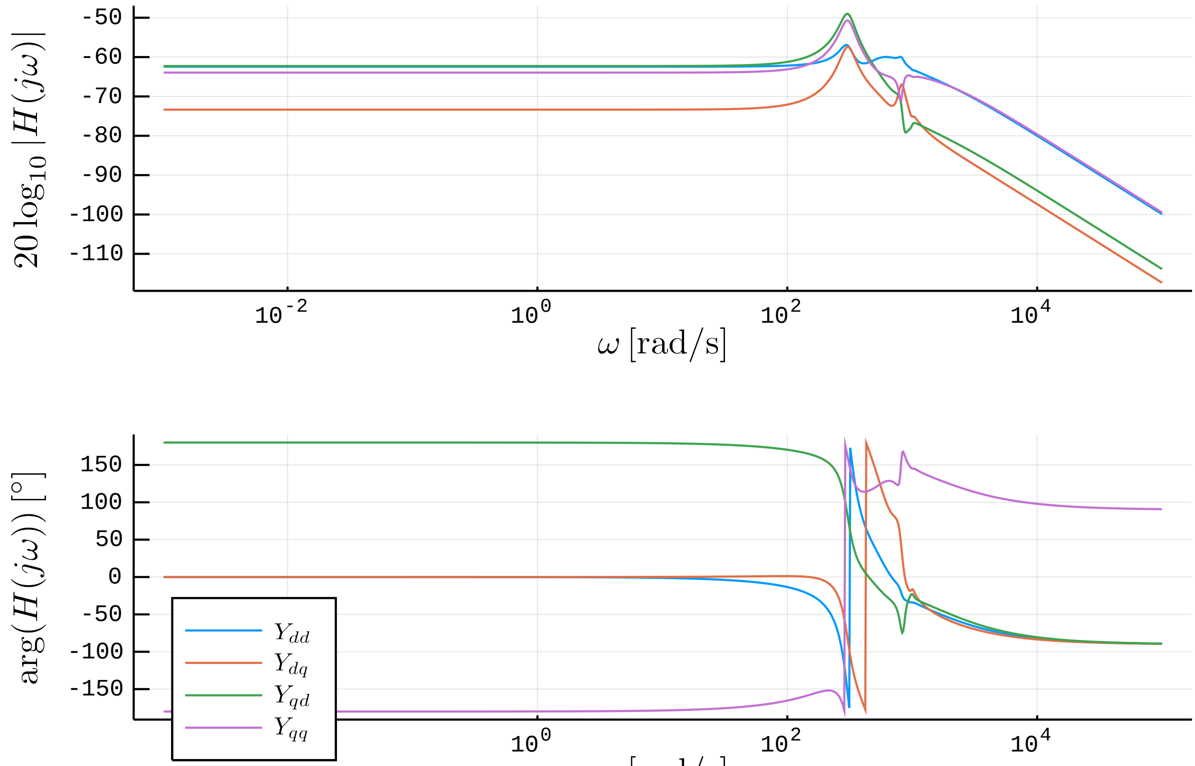


Figure 34: AC side admittance of the MMC.

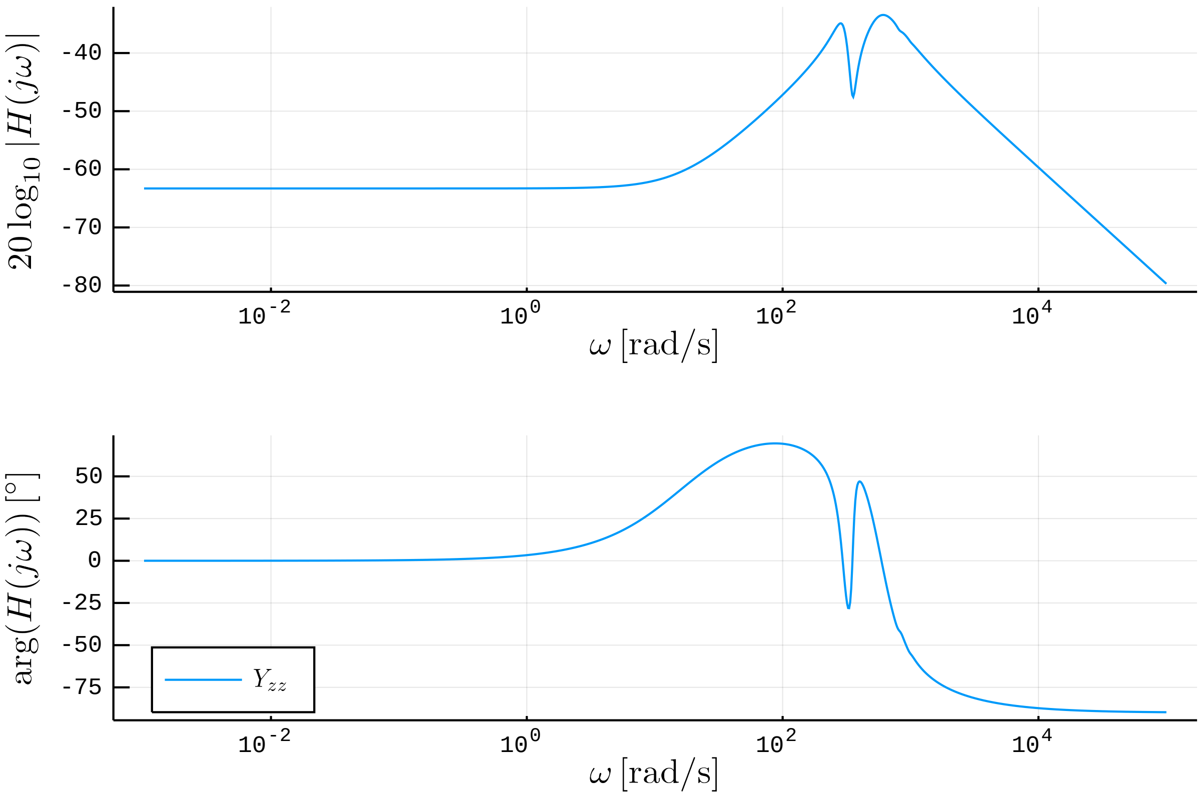


Figure 35: DC side impedance of the MMC.

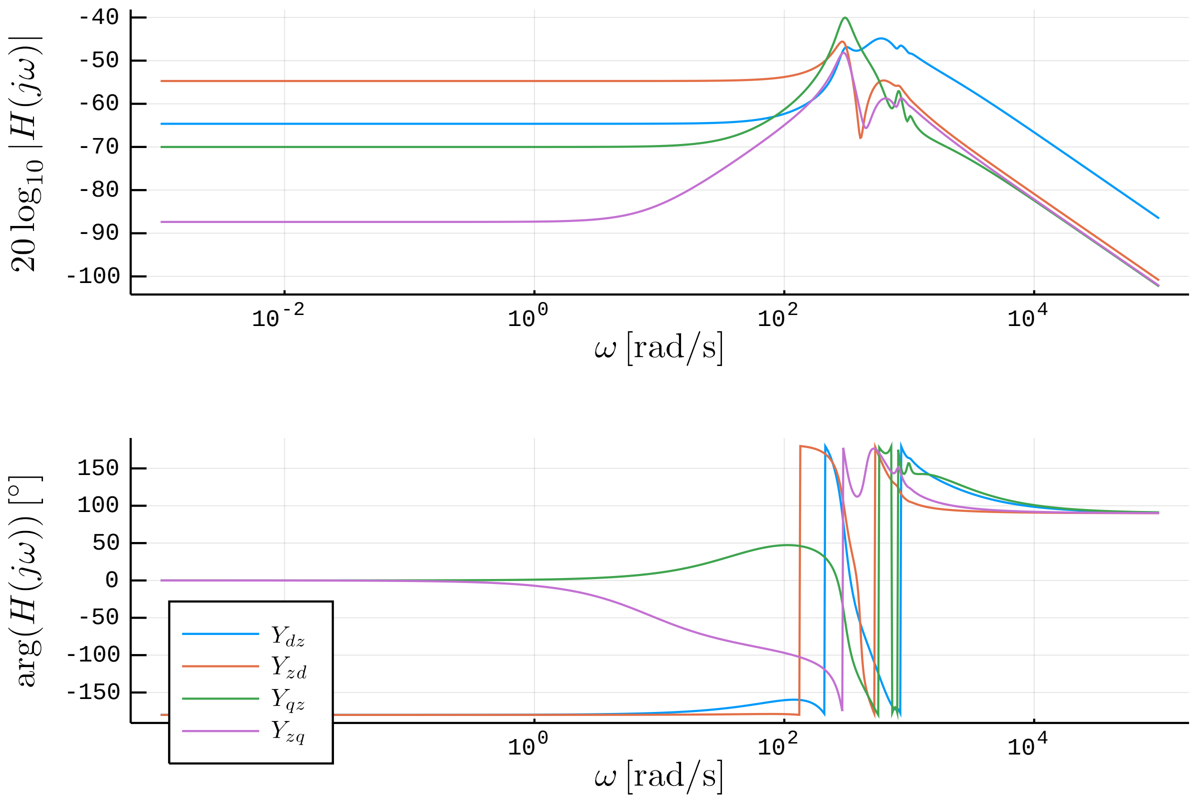


Figure 36: Admittance interconnection between AC and DC side.

In order to demonstrate the functionality of the simulator, multiple diagrams are generated for the same test example, but with base values $P = 1000$ MVA, $Q = 0$, $V_m = 333$ kV and $\theta = 0$.

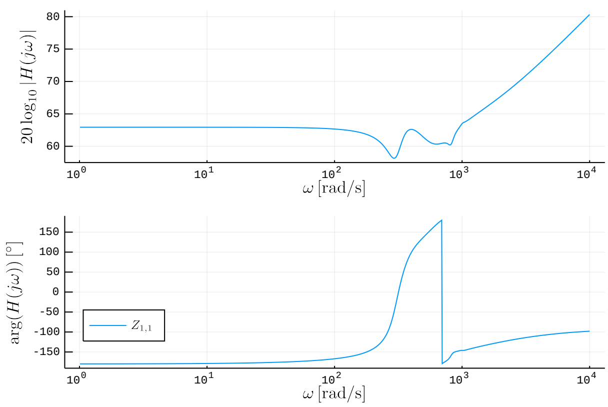
In the case when the impedance between two out of three pins denoted as x and y ($x, y \in \{d, q, z\}$) is estimated, while the third pin is short connected to the ground, the expected impedance is $Z_{eq} = \frac{1}{Y_{xx}}$, where $Y_{xx} \in \{Y_{dd}, Y_{qq}, 3Y_{zz}\}$. The same result is obtained in the simulation and depicted in Fig. 37.

For the case when one pin is considered as “open” and the impedance is determined between other pins, the simulated results are depicted in Fig. 38. The impedance between pins x ($x = d, q$) and z for the short connected third pin, the expected impedance is:

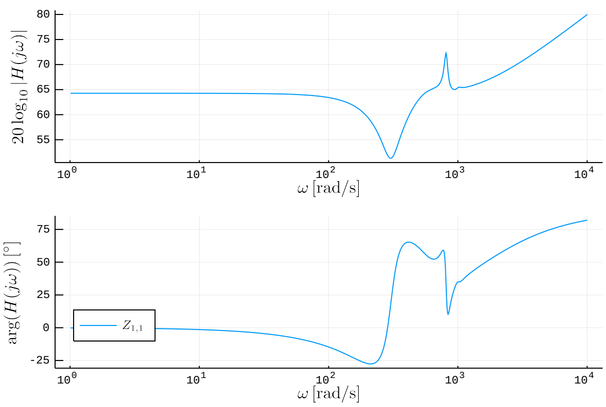
$$Z_{eq} = \frac{1}{3\left(Y_{zz} - \frac{Y_{zx}Y_{xz}}{Y_{xx}}\right)}.$$

Impedance between pins x and y for $x, y = d, q$ with open DC pin z is

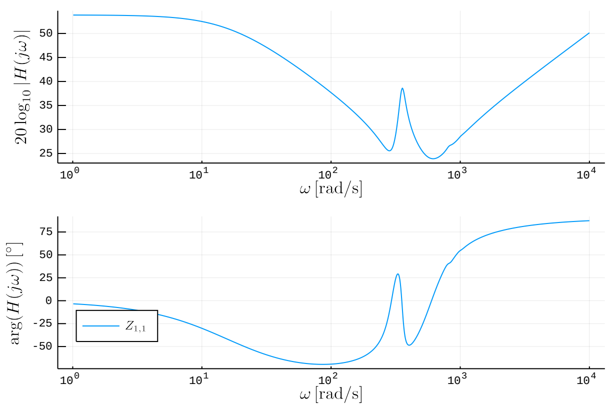
$$Z_{eq} = \frac{1}{Y_{xx} - \frac{Y_{zx}Y_{xz}}{Y_{zz}}}.$$



(a)



(b)



(c)

Figure 37: Impedance between two pins when the third is short connected: (a) between d and q pins; (b) between q and d pins; and (c) between dc pin and short connected d and q pins.

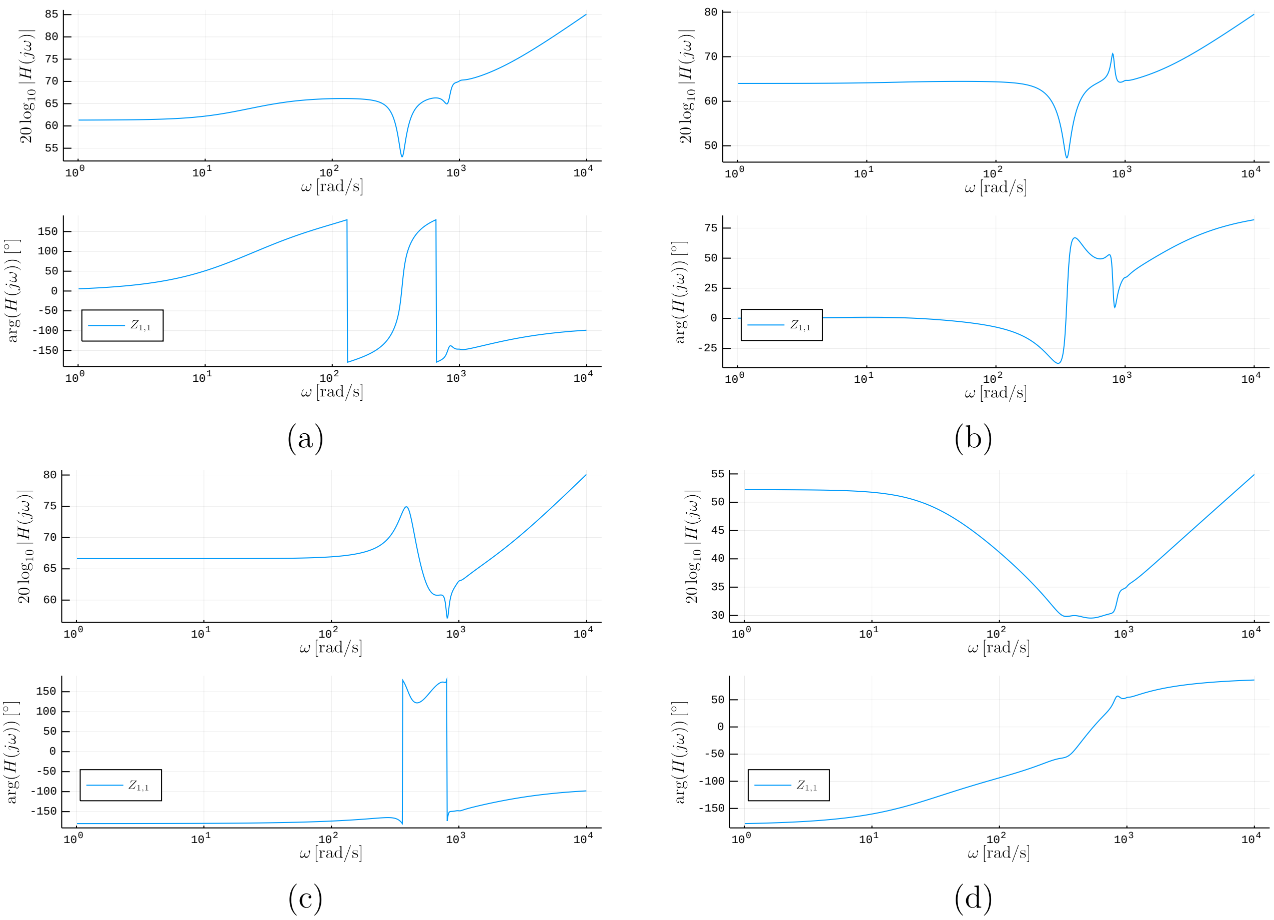


Figure 38: Impedance between two pins when the third is open connection: (a) between d and q pins; (b) between q and d pins; (c) between z and d; and (d) between z and q.

An example for the simulation of the MMC with an incorporated PLL is given with the following listing.

```
mmc(S_base = 1000e6, V_base = 333e3,
    energy = PI_control(Kp = 120, Ki = 400),
    occ = PI_control(ζ = 0.7, bandwidth = 1000),
    ccc = PI_control(ζ = 0.7, bandwidth = 300),
    zcc = PI_control(ζ = 0.7, bandwidth = 300),
    power = PI_control(Kp = 2.0020e-07, Ki = 1.0010e-04),
    pll = PI_control(Kp = 2e-3, Ki = 2))
```

The admittances of the MMC can be displayed using the function `plot_data` and are presented in Figs. 39, 40 and 41.

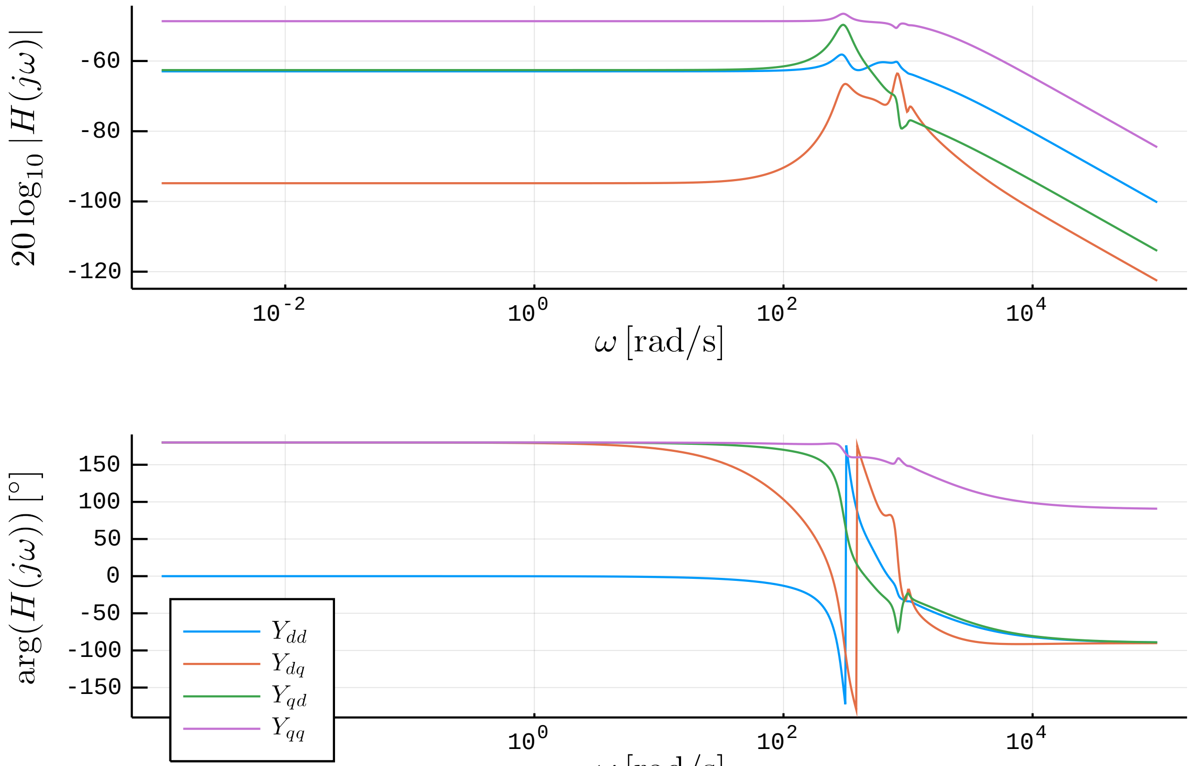


Figure 39: AC side admittance of the MMC with PLL.

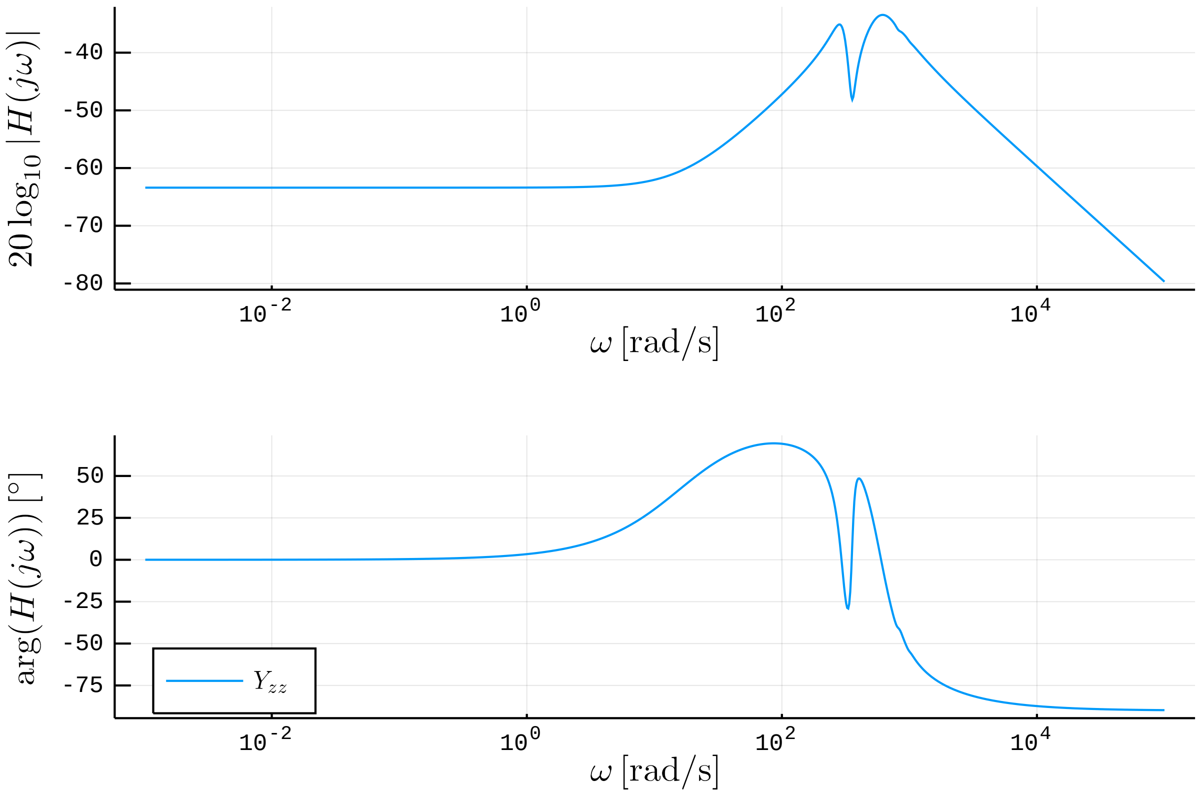


Figure 40: DC side impedance of the MMC with PLL incorporated.

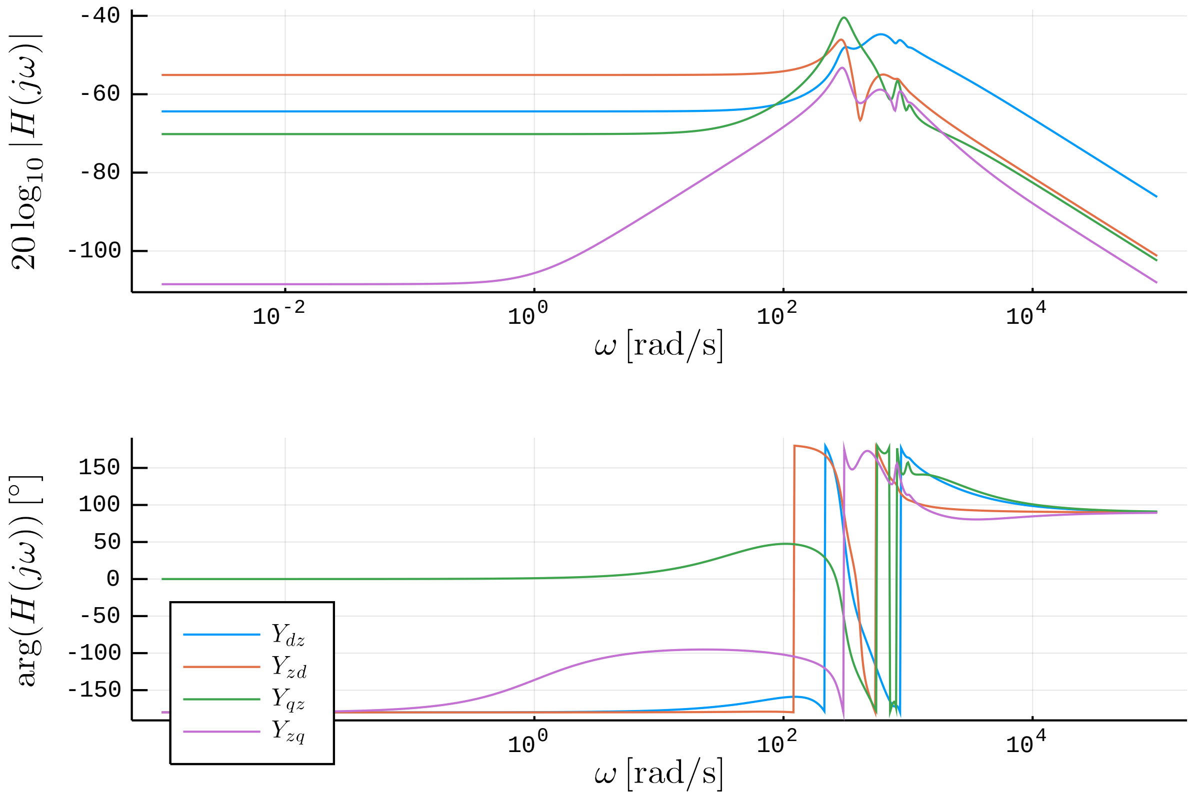


Figure 41: Admittance interconnection between AC and DC side of the MMC with PLL applied.

6.4 Quick start - Implementation of a new component

1. Make a new folder in `/Network/Components/new_folder` with the name of the new component;

2. Make a file in the folder with the name `new_component_name.jl`;

3. Implement the component as the struct type. The struct can contain any number of required fields with filled in initial values. The following lines provide universal struct format, whose contructor is generated automatically by the compiler. The constructor is then called by `New_component_name()` and it gives as an output one instance of the component with the initialized all component values to the default ones.

```
export new_component_name

@with_kw mutable struct New_component_name
    field :: Type_field = initial_value
end
```

4. Make a new function called `new_component_name`:

```
"""
Comments and description of the function
"""
function new_component(;args... )

    # Insert here the code that builds the ABCD representation
    # or the equivalent component parameters
    # used later for ABCD representation
    # of the component

    elem = New_component_name()
    element = Element(element_value = elem, input_pins = nb_input_pins,
        output_pins = nb_output_pins)
    return element
end
```

5. Make function for the estimation of ABCD and Y parameters;

```
function eval_abcd(elem :: New_component_name, s :: Complex)

    # Number of instructions for evaluating ABCD parameters

    return ABCD_parameters
end
```

6. If the component is intended to be added to the power flow, then the two functions for setting power flow parameters are added;

```
function make_power_flow_ac!(elem :: New_component_name , dict :: Dict{String,
    global_dict :: Dict{String, Any})

    # Insert here the code to fill-in the powerflow dictionary.
    # Example:
```

```
    key = length(dict["new_powerflow_component"])
    dict["new_powerflow_component"][string(key)]["parameter"] = value
    # where new_powerflow_component, parameter and value must be adapted.
end

function make_power_flow_dc!(elem :: New_component_name , dict :: Dict{String,
        global_dict :: Dict{String, Any})

    # Insert here the code to fill-in the powerflow dictionary.
end
```

7. Add functions for plotting and saving component related data:

   `function save_data(comp :: New_component_type, file_name :: String, omegas)`,

   `function plot_data(comp :: New_component_type, omegas)`;

8. In the file `element_types`, add a new line `include("new_folder/new_component_name.jl")` to link it with the power system structures.

A Appendices

A.1 Future work

Concerning the implementation:

- Transformer models with grounded neutral point for Wye configurations;
- Current sources, as only voltages sources are implemented at the moment;
- Nyquist plot for stability assessment;
- Harmonic state space and transfer functions (HSS);
- Harmonic power flow for initialisation of HSS.

Concerning the report:

- Table of differences between the Simulator and PSCAD concerning the implementation of transmission lines, in case of different naming conventions, unavailable features, etc. For instance, the ideal transposition is not yet implemented for overhead lines.

A.2 Park's transformation

- Park's transformation

$$\mathbf{P}_{\omega_0}(t) = \frac{2}{3} \begin{bmatrix} \cos(\omega_0 t) & \cos\left(\omega_0 t - \frac{2\pi}{3}\right) & \cos\left(\omega_0 t - \frac{4\pi}{3}\right) \\ \sin(\omega_0 t) & \sin\left(\omega_0 t - \frac{2\pi}{3}\right) & \sin\left(\omega_0 t - \frac{4\pi}{3}\right) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \tag{96}$$

- Inverse Park's transformation

$$\mathbf{P}_{\omega_0}^{-1}(t) = \frac{3}{2}\,\mathbf{P}_{\omega_0}^{T}(t) + \frac{1}{2} \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \tag{97}$$

- Fourier transform applied to Park's transform:

$$\mathbf{P}_{\omega_0}(j\omega) = \frac{2\pi}{3} \left(\mathbf{a}\,\delta(\omega + \omega_0) + \overline{\mathbf{a}}\,\delta(\omega - \omega_0) + \mathbf{c}\,\delta(\omega)\right), \tag{98}$$

  where $\overline{\mathbf{a}}$ denotes conjugate of the matrix $\mathbf{a}$ and

$$\mathbf{a} = \begin{bmatrix} 1 & \exp(j\varphi) & \exp(2j\varphi) \\ j & j\exp(j\varphi) & j\exp(2j\varphi) \\ 0 & 0 & 0 \end{bmatrix},$$

$$\mathbf{c} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix},$$

  for $\varphi = \frac{2\pi}{3}$.

- Fourier transform applied to the inverse Park's transform

$$\mathbf{P}_{\omega_0}^{-1}(j\omega) = \pi\left(\mathbf{a}^T\,\delta(\omega + \omega_0) + \overline{\mathbf{a}}^T\,\delta(\omega - \omega_0) + 2\mathbf{c}^T\,\delta(\omega)\right) \tag{99}$$

A.3 Fourier transform

A.3.1 Fourier transform theorems

Time domain $x(t)$	**Spectral domain** $X(j\omega) = X(s)$, $s = j\omega$
$ax(t) + by(t)$	$aX(j\omega) + bY(j\omega)$
$x(t - t_0)$	$X(j\omega)\,\mathrm{e}^{-j\omega t_0}$
$x(t)\,\mathrm{e}^{j\alpha t}$	$X(j(\omega - \alpha))$
$x^*(t)$	$X^*(-j\omega)$
$x(-t)$	$X(-j\omega)$
$x(at)$	$\dfrac{1}{\lvert a\rvert}\, X\left(\frac{\omega}{a}\right)$
$x(t) * y(t)$	$X(j\omega)\,Y(j\omega)$
$x(t)\,y(t)$	$\frac{1}{2\pi}\, X(j\omega) * Y(j\omega)$
$\dfrac{d}{dt}x(t)$	$j\omega\, X(j\omega)$
$\displaystyle\int_{-\infty}^{t} x(\tau)\, d\tau$	$\left(\frac{1}{j\omega} + \pi\,\delta(\omega)\right)\, X(j\omega)$
$t\, x(t)$	$j\,\frac{d}{d\omega}X(j\omega)$
$x(t) \in \mathbb{R}$	$X(j\omega) = X^*(-j\omega)$ $\Re\{X(j\omega)\} = \Re\{X(-j\omega)\}$ $\Im\{X(j\omega)\} = -\Im\{X(-j\omega)\}$ $\lvert X(j\omega)\rvert = \lvert X(-j\omega)\rvert$ $\arg(X(j\omega)) = -\arg(X(-j\omega))$
$x_e(t)$ $x_o(t)$	$\Re\{X(j\omega)\}$ $-\Im\{X(j\omega)\}$
$x(t)$ $x(jt)$ $x(-jt)$	$X(j\omega)$ $2\pi\, X(-\omega)$ $2\pi\, X(\omega)$

A.3.2 Fourier transform table of the common signals

$x(t)$	$X(j\omega)$
1	$2\pi\,\delta(\omega)$
$\delta(t)$	1
$\mathrm{u}(t)$	$\dfrac{1}{j\omega} + \pi\,\delta(\omega)$
$\mathrm{e}^{j\omega_0 t}$	$2\pi\,\delta(\omega - \omega_0)$
$\mathrm{rect}(t)$	$\mathrm{sinc}\left(\frac{\omega}{2\pi}\right)$
$\mathrm{sinc}(t)$	$\mathrm{rect}\left(\frac{\omega}{2\pi}\right)$
$\mathrm{comb}(t)$	$\mathrm{comb}\left(\frac{\omega}{2\pi}\right)$
$\cos(\omega_0 t)$	$\pi\left(\delta(\omega + \omega_0) + \delta(\omega - \omega_0)\right)$
$\sin(\omega_0 t)$	$j\pi\left(\delta(\omega + \omega_0) - \delta(\omega - \omega_0)\right)$
$\mathrm{sinc}^2(t)$	$\mathrm{tri}\left(\frac{\omega}{2\pi}\right)$
$\mathrm{tri}(t)$	$\mathrm{sinc}^2\left(\frac{\omega}{2\pi}\right)$
$\mathrm{e}^{-at}\,\mathrm{u}(t), \quad \Re\{a\} > 0$	$\dfrac{1}{a + j\omega}$
$\mathrm{e}^{-\pi t^2}$	$\mathrm{e}^{-\frac{\omega^2}{4\pi}}$
$\dfrac{t^{n-1}}{(n-1)!}\,\mathrm{e}^{-at}\,\mathrm{u}(t), \quad \Re\{a\} > 0$	$\dfrac{1}{(a + j\omega)^n}$
$\mathrm{e}^{-at}\cos(\omega_0 t)\,\mathrm{u}(t), \quad \Re\{a\} > 0$	$\dfrac{a + j\omega}{(a + j\omega)^2 + \omega_0^2}$
$\mathrm{e}^{-at}\sin(\omega_0 t)\,\mathrm{u}(t), \quad \Re\{a\} > 0$	$\dfrac{\omega_0}{(a + j\omega)^2 + \omega_0^2}$

A.4 Glossary

. .

- `Element`: (...)
- `Network`: A set of components. The elements of type `Element` are gathered in the structure `Elements` and their interconnections are gathered in the dictionary `Net`
- node: A point of connection between two or more pins
- pin: An input or output port of an element. E.g. a three-phase transformer has three input pins and three output pins

References

[1] X. Wang, F. Blaabjerg, and W. Wu, "Modeling and analysis of harmonic stability in an ac power-electronics-based power system," *IEEE Transactions on Power Electronics*, vol. 29, no. 12, pp. 6421–6432, 2014.

[2] P. Kundur, N. J. Balu, and M. G. Lauby, *Power system stability and control.* McGraw-hill New York, 1994, vol. 7.

[3] V. K. Sood, *HVDC and FACTS controllers: applications of static converters in power systems.* Springer Science & Business Media, 2006.

[4] K. Padiyar, *HVDC power transmission systems: technology and system interactions.* New Age International, 1990.

[5] A. Bayo Salas, "Control interactions in power systems with multiple VSC HVDC converters," Ph.D. dissertation, KU Leuven, 2018.

[6] X. Wang and F. Blaabjerg, "Harmonic stability in power electronic-based power systems: Concept, modeling, and analysis," *IEEE Transactions on Smart Grid*, vol. 10, no. 3, pp. 2858–2870, 2018.

[7] R. Middlebrook *et al.*, "Design techniques for preventing input-filter oscillations in switched-mode regulators," in *Proc. Powercon*, vol. 5, 1978, pp. A3–1.

[8] K. Ji, G. Tang, J. Yang, Y. Li, and D. Liu, "Harmonic stability analysis of mmc-based dc system using dc impedance model," *IEEE Journal of Emerging and Selected Topics in Power Electronics*, 2019.

[9] A. J. Agbemuko, J. L. Domínguez-García, E. Prieto-Araujo, and O. Gomis-Bellmunt, "Dynamic modelling and interaction analysis of multi-terminal vsc-hvdc grids through an impedance-based approach," *International Journal of Electrical Power & Energy Systems*, vol. 113, pp. 874–887, 2019.

[10] J. Lyu, X. Zhang, X. Cai, and M. Molinas, "Harmonic state-space based small-signal impedance modeling of a modular multilevel converter with consideration of internal harmonic dynamics," *IEEE Transactions on Power Electronics*, vol. 34, no. 3, pp. 2134–2148, 2018.

[11] L. Harnefors, M. Bongiorno, and S. Lundberg, "Input-admittance calculation and shaping for controlled voltage-source converters," *IEEE Transactions on Industrial Electronics*, vol. 54, no. 6, pp. 3323–3334, Dec 2007.

[12] F. Milano, *Power system modelling and scripting.* Springer Science & Business Media, 2010.

[13] G. Stamatiou, M. Beza, M. Bongiorno, and L. Harnefors, "Analytical derivation of the DC-side input admittance of the direct-voltage controlled modular multilevel converter," *IET Generation, Transmission & Distribution*, vol. 11, no. 16, pp. 4018–4030, 2017.

[14] L. Bessegato, L. Harnefors, K. Ilves, and S. Norrga, "A method for the calculation of the ac-side admittance of a modular multilevel converter," *IEEE transactions on power electronics*, vol. 34, no. 5, pp. 4161–4172, 2018.

[15] H. Liu, X. Xie, and W. Liu, "An oscillatory stability criterion based on the unified *dq*-frame impedance network model for power systems with high-penetration renewables," *IEEE Transactions on Power Systems*, vol. 33, no. 3, pp. 3472–3485, 2018.

[16] Y. Sun, L. Wu, X. Wang, E. C. de Jong, F. Blåbjerg, V. Cuk, and J. Cobben, "Adequacy analysis of overhead line model for harmonic stability analysis of grid-connected voltage-source converters," in *2018 IEEE 19th Workshop on Control and Modeling for Power Electronics (COMPEL)*. IEEE, 2018, pp. 1–8.

[17] L. Wu *et al.*, "Impact of ehv/hv underground power cables on resonant grid behavior," *Eindhoven: Eindhoven University of Technology*, 2014.

[18] Y. Levron and J. Belikov, "Modeling power networks using dynamic phasors in the dq0 reference frame," *Electric Power Systems Research*, vol. 144, pp. 233–242, 2017.

[19] J. Belikov and Y. Levron, "Integration of long transmission lines in large-scale dq0 dynamic models," *Electrical Engineering*, vol. 100, no. 2, pp. 1219–1228, 2018.

[20] S. D'Arco, J. A. Suul, and J. Beerten, "Time-invariant state-space model of an ac cable by dq-representation of frequency-dependent pi-sections," *Proceedings of the 2019 IEEE PES Innovative Smart Grid Technologies Europe*, 2019.

[21] C. T. Rim, D. Y. Hu, and G. H. Cho, "Transformers as equivalent circuits for switches: General proofs and dq transformation-based analyses," *IEEE Transactions on Industry Applications*, vol. 26, no. 4, pp. 777–785, 1990.

[22] T. Reveyrand, "Multiport conversions between s, z, y, h, abcd, and t parameters," in *2018 International Workshop on Integrated Nonlinear Microwave and Millimetre-wave Circuits (INMMIC)*. IEEE, 2018, pp. 1–3.

[23] D. A. Frickey, "Conversions between s, z, y, h, abcd, and t parameters which are valid for complex source and load impedances," *IEEE Transactions on microwave theory and techniques*, vol. 42, no. 2, pp. 205–211, 1994.

[24] F. Dorfler and F. Bullo, "Kron reduction of graphs with applications to electrical networks," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 60, no. 1, pp. 150–163, 2012.

[25] C.-W. Ho, A. Ruehli, and P. Brennan, "The modified nodal approach to network analysis," *IEEE Transactions on circuits and systems*, vol. 22, no. 6, pp. 504–509, 1975.

[26] J. A. Martinez-Velasco, *Power system transients: parameter determination*. CRC press, 2017.

[27] P. M. Anderson and P. M. Anderson, *Analysis of faulted power systems*. IEEE press New York, 1995, vol. 445.

[28] F. Castellanos and J. Marti, "Full frequency-dependent phase-domain transmission line model," *IEEE transactions on Power Systems*, vol. 12, no. 3, pp. 1331–1339, 1997.

[29] A. Morched, B. Gustavsen, and M. Tartibi, "A universal model for accurate calculation of electromagnetic transients on overhead lines and underground cables," *IEEE Transactions on Power Delivery*, vol. 14, no. 3, pp. 1032–1038, 1999.

[30] H. Manitoba, "Research centre inc," *PSCAD/EMTDC user's guide*, 2003.

[31] A. Ametani, “A general formulation of impedance and admittance of cables,” *IEEE transactions on power apparatus and systems*, no. 3, pp. 902–910, 1980.

[32] P. de Arizon and H. W. Dommel, “Computation of cable impedances based on subdivision of conductors,” *IEEE Transactions on Power Delivery*, vol. 2, no. 1, pp. 21–27, 1987.

[33] F. De Léon, M. L. Márquez-Asensio, and G. Álvarez-Cordero, “Effects of conductor counter-transposition on the positive-sequence impedance and losses of cross-bonded cables,” *IEEE Transactions on Power Delivery*, vol. 26, no. 3, pp. 2060–2063, 2011.

[34] R. A. Rivas and J. R. Marti, “Calculation of frequency-dependent parameters of power cables: Matrix partitioning techniques,” *IEEE transactions on power delivery*, vol. 17, no. 4, pp. 1085–1092, 2002.

[35] G. Bergna-Diaz, J. Freytes, X. Guillaud, S. D’Arco, and J. A. Suul, “Generalized voltage-based state-space modeling of modular multilevel converters with constant equilibrium in steady state,” *IEEE Journal of Emerging and Selected Topics in Power Electronics*, vol. 6, no. 2, pp. 707–725, 2018.

[36] G. Bergna-Diaz, D. Zonetti, S. Sanchez, R. Ortega, and E. Tedeschi, “Pi passivity-based control and performance analysis of mmc multi-terminal hvdc systems,” *IEEE Journal of Emerging and Selected Topics in Power Electronics*, 2018.

[37] Ö. C. Sakinci and J. Beerten, “Generalized dynamic phasor modeling of the mmc for small-signal stability analysis,” *IEEE Transactions on Power Delivery*, vol. 34, no. 3, pp. 991–1000, 2019.

[38] J. Freytes, “Analyse de stabilité en petit signaux des convertisseurs modulaires multiniveaux et application à l’étude d’interopérabilité des mmc dans les réseaux hvdc,” Ph.D. dissertation, Ecole centrale de Lille, 2017.

[39] J. Revels, M. Lubin, and T. Papamarkou, “Forward-mode automatic differentiation in julia,” *arXiv:1607.07892 [cs.MS]*, 2016. [Online]. Available: https://arxiv.org/abs/1607.07892

[40] H. Ergun, J. Dave, D. Van Hertem, and F. Geth, “Optimal power flow for ac/dc grids: Formulation, convex relaxation, linear approximation, and implementation,” *IEEE Transactions on Power Systems*, vol. 34, no. 4, pp. 2980–2990, July 2019.

[41] G. F. Ergun, H. and D. Van Hertem, ““powermodelsacdc repository,” *Department Electrical Engineering, University of Leuven*, 2018. [Online]. Available: https://github.com/hakanergun/PowerModelsACDC.jl

[42] J. Beerten, “Matacdc 1.0 user’s manual,” *Department Electrical Engineering, University of Leuven*, 2012. [Online]. Available: https://www.esat.kuleuven.be/electa/teaching/matacdc/MatACDCManual

[43] R. D. Zimmerman and C. E. Murillo-Sánchez, “Matpower 6.0 user’s manual,” *PSERC: Tempe, AZ, USA*, 2016.