Development of an open source power flow software for high voltage direct current grids and hybrid AC/DC systems: MATACDC — Source link ⌂

Jef Beerten, Ronnie Belmans

Institutions: Katholieke Universiteit Leuven

Related papers:

- Generalized Steady-State VSC MTDC Model for Sequential AC/DC Power Flow Algorithms

- MATPOWER: Steady-State Operations, Planning, and Analysis Tools for Power Systems Research and Education

- A New Approach for Benefit Evaluation of Multiterminal VSC–HVDC Using A Proposed Mixed AC/DC Optimal Power Flow

- MatACDC - an open source software tool for steady-state analysis and operation of HVDC grids

- Second-Order Cone Programming for Optimal Power Flow in VSC-Type AC-DC Grids

Development of an Open Source Power Flow Software for HVDC Grids and Hybrid AC/DC Systems: MatACDC

Jef Beerten, Ronnie Belmans

University of Leuven (KU Leuven) Dept. Electrical Engineering (ESAT), Division ELECTA

Kasteelpark Arenberg 10, bus 2445; 3001 Heverlee, Belgium

email: jef.beerten@esat.kuleuven.be, ronnie.belmans@esat.kuleuven.be

Abstract

This paper presents the creation of a new simulation tool, MATACDC. It is the first open source program for power flow analysis of HVDC grids and hybrid AC/DC systems and uses state-of-the-art developments in the field of HVDC grids research. MATACDC is based on MATLAB and has been fully integrated with the AC system power flow routines from MATPOWER. The software includes all the models needed to study the steady-state interaction of AC and DC systems for a wide range of converter representations and control functions. Any combination of multiple non-synchronised AC systems and multiple DC systems can be solved. The code is freely available and is intended for researchers and students working in the field of HVDC grid steady-state interactions and HVDC grid operation. MATACDC can also easily be extended with user-defined functionality. The paper focuses on the program design and layout, the converter modelling, the practical implementation and the integration with AC power flow routines. Furthermore, different examples of possible user-defined functions show how the tool can be extended to include other control representations to study their effect on overall system interactions. Simulation results demonstrate the viability of the routines to simulate complex hybrid AC/DC systems.

I. Introduction

In recent years, the power engineering community is showing an increased interest in HVDC grids or so-called supergrids. This can be mainly attributed to the European plans of constructing a North-Sea grid to interconnect offshore wind farms using DC technology, which can evolve into a pan-European DC supergrid [1]. The introduction of HVDC grids fundamentally challenges the operation of the existing AC systems: when using a distributed voltage control, contingencies such as converter outages will lead to a systemwide response and a change of the power flows in the entire system, influenced by the HVDC grid layout itself [2], [3]. After the fast initial response determined by the control characteristics, a slower secondary control can be added to redispatch the power flows [4]. It is therefore essential that the HVDC grid is adequately modelled in future Optimal Power Flow (OPF) formulations [5], [6]. The OPF problem can also be extended to include security constraints [7] or make the HVDC grid actively participate in power flow changes due to the unpredictability of wind power [8].

For the analysis of a future meshed HVDC grid, power flow software is thus needed. Such software must be capable of dealing with power flow problems in complex hybrid AC/DC systems. Commercial programs to calculate the power flow in AC networks including HVDC systems have been available for a long time. However, it is only in recent years that the inclusion of meshed Voltage Source Converter High Voltage Direct Current (VSC HVDC) systems in these packages is given full attention. A main issue is that these commercial packages are characterised by a steep learning curve and a considerable cost, impeding their use in basic research and education. The main disadvantage of these programs is that the models are usually closed, hence not leaving ample room for modifications to the models themselves.

As an alternative to commercial programs, different open-source alternatives has been reported in the literature. One of these tools is MATPOWER [9], a MATLAB-based open source software package with power flow and OPF routines for AC systems. However, the focus is on AC systems, hence it is not possible to study HVDC grids and to address the steady-state interactions in a hybrid AC/DC system, consisting of several AC and HVDC grids.

This paper presents the creation of MATACDC, a new open-source power flow program designed to address these challenges. The software builds on the idea of using the MATLAB software as a computational engine. MATACDC is the first program of its kind allowing to study the steady-state interactions between multiple non-synchronised AC grids and DC systems based on VSC HVDC technology. Besides regular power flow models, a variety of converter models and control representations, such as DC voltage droop control [10], [11] have been included, as well as a number of DC system contingencies (e.g. converter outages). This makes the program ideally suited for researchers and students working in the emerging field of HVDC grids. The program is freely available on the internet [12] and a fully-documented user manual and code are provided. MATACDC is fully compatible with the power flow routines in MATPOWER, which is available online [13]. The program has a similar syntax, code structure and command line usage, which makes its use straightforward for persons who are already familiar with MATPOWER.

The paper is structured as follows: Section II describes the implementation, Section III introduces the models and Section IV covers the program flow. Section V discusses the inclusion of user-defined functionality to the program. Finally, Section VI addresses simulation results.

II. IMPLEMENTATION

A. Design Criteria

In the design phase, as much care as possible has been taken to integrate the program seamlessly with MATPOWER, a power flow and optimal power flow program in MATLAB [9]. The package has been fully integrated with the existing AC power flow routines developed in MATPOWER, while keeping the MATPOWER original source code unaltered. The AC/DC power flow problem is solved sequentially, meaning that the program solves the AC/DC power flow by iterating between the AC systems and the DC systems. Doing so, the DC system quantities remain unaltered during the AC system power flow, and vice versa.

MATACDC shares the same philosophy as MATPOWER, and *MatDyn* [14], a MATLAB-based open source transient stability program for AC systems: "It is intended as a simulation tool for researchers and educators that is easy to use and modify." [13]. The program has been designed such that MATPOWER users can easily
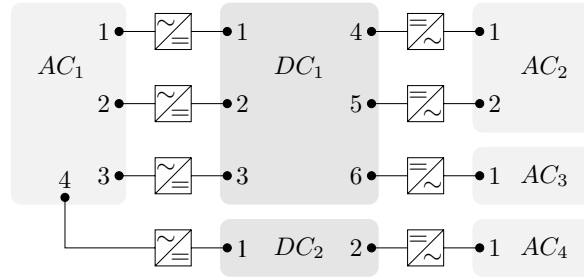
Fig. 1. Interconnected hybrid AC/DC system.

use MATACDC due to a similar program structure and many similarities, such as

- MATACDC uses the MATPOWER power flow data case files.
- MATACDC uses similar DC system case files, with three distinctive data matrices, `busdc`, `convdc` and `branchdc`, to store the information on the DC system and the interconnection with the AC system. The choice of these three data matrices is analogous to the subdivision in the `bus`, `gen` and `branch` matrices in MATPOWER's AC power flow data input.
- Similar to the power flow command `runpf` in MATPOWER, MATACDC uses the `runacdcpf` command to run an AC/DC power flow:

  ```
  >> runacdcpf(casefileac,casefiledc);
  ```

  in which `casefileac` is the MATPOWER AC power flow data case file and `casefiledc` the MAT-ACDC DC power flow data case file.
- MATACDC also stores the results in MATLAB structs:

  ```
  >> [mpc, mpcdc] = runacdcpf(...);
  ```

- MATACDC uses an option vector similar to MATPOWER. For example, printing the output can be suppressed by:

  ```
  >> opt = macdcoption;
  >> opt(13) = 0;
  >> [mpc, mpcdc] = runacdcpf(casefileac,casefiledc,opt);
  ```

As depicted in Fig. 1, MATACDC can be used to study complex interconnected systems with an arbitrary number of AC and DC systems. For convenience, the AC and DC bus numbers forming the interconnections to respectively the DC and the AC network have been sequentially numbered, starting from 1. Neither the AC nor DC systems have to be limited to these interconnected buses. In this example, AC systems 3 and 4 only share one converter with respectively HVDC grids 1 and 2. These systems can represent relatively small systems, e.g. island systems or (offshore) wind farms connected to the offshore systems, but they can also represent larger AC systems that only have one interconnection with the other AC systems in the grid.

B. Data Representation

Similar to the data representation in MATPOWER, MATACDC uses MATLAB m-files that return a MATLAB struct. The fields of this struct are `pol`, `baseMVAac`, `baseMVAdc`, `busdc`, `convdc` and `branchdc`. The
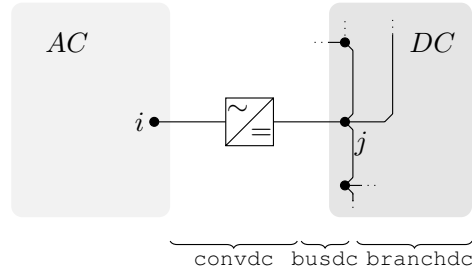
Fig. 2. AC/DC system implementation: conceptual representation.

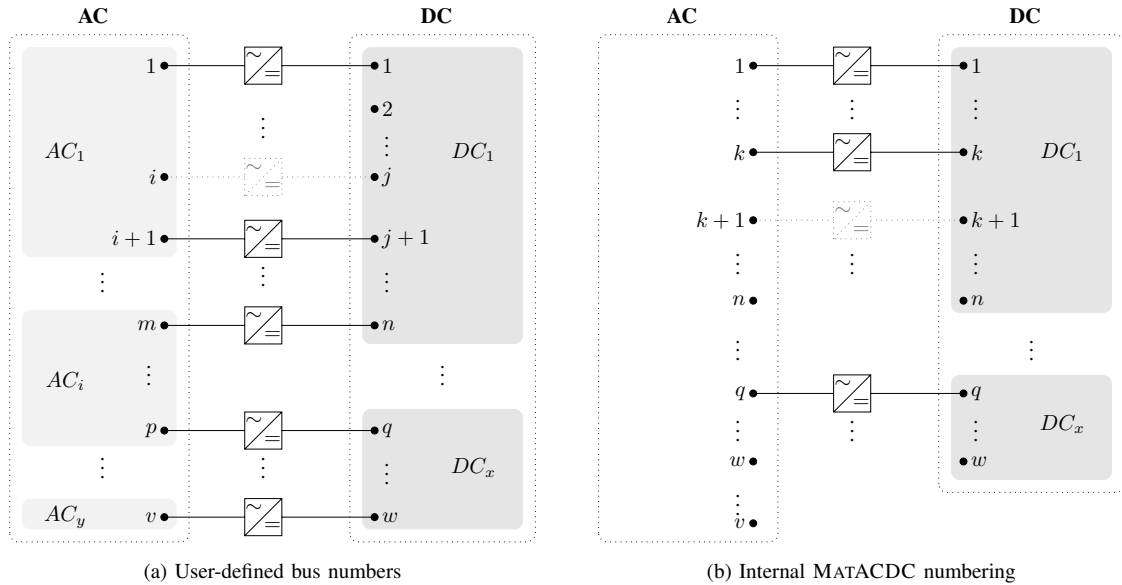(a) User-defined bus numbers

(b) Internal MATACDC numbering

Fig. 3. AC/DC system interconnections (a) before and (b) after internal bus renumbering.

fields `pol`, `baseMVAac` and `baseMVAdc` are scalar values, the other fields are data matrices containing the following data:

- `busdc`: DC bus data (bus numbers, active power withdrawals, voltages, ...)
- `convdc`: Converter station data (converter powers, voltages, loss data, converter limits, impedance values, status, control modes, ...)
- `branchdc`: DC branch data (bus numbers, line resistance, ...)

Full details of the named matrix indices are found in the MATACDC manual [12]. The aforementioned subdivision of the DC system data has been visualised in Fig. 2. The structure is similar to the AC power flow data division into a bus-, generator- and branchmatrix in MATPOWER.

MATACDC input data can have an arbitrary DC bus number associated with any AC bus number. Internally, the bus indices are redefined so that interconnected AC and DC buses have the same bus numbers. The bus renumbering allows full user flexibility while keeping the code straightforward. This renumbering algorithm is depicted graphically in Fig. 3 for an interconnected system of y AC grids with a total of v buses and x HVDC grids with a total of w DC buses.
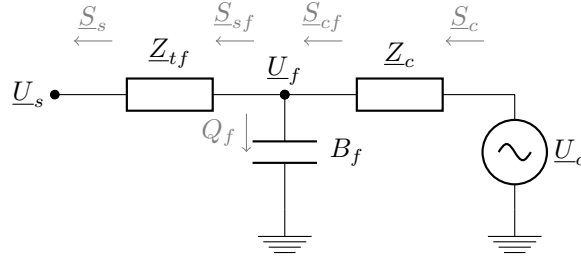
Fig. 4. Equivalent single phase power flow model of a converter station connected to the AC grid.

Each DC grid can have an arbitrary number of converter-connected DC buses, as well as DC buses without an AC interconnection (e.g. DC bus 2 in grid 1 from Fig. 3a) and DC buses facing a converter outage (e.g. DC bus j in grid 1 from Fig. 3a). Similarly, each AC bus can either have a connection to the DC system, or can be an AC bus without an interconnection.

As suggested in Fig. 3b, the MATACDC sorting algorithm puts an emphasis to the DC system and does not account for the different AC grids. The rationale for this approach relies in the fact that the MATPOWER power flow routine `runpf`, which is used for the AC system power flow, internally renumbers the buses for each AC system.

III. MODELS

This section introduces the modelling of the AC/DC system, with an emphasis on the implementation in MATACDC. The modelling is based on well-known state-of-the-art converter and DC system models. Further details are found in [15].

A. Converters

As seen from the AC Point of Common Coupling (PCC), the different components of the VSC are:

- the converter transformer
- the AC filters
- the phase reactor
- the converter

1) AC Side Model: Fig. 4 shows the most general format of the converter's AC side, consisting of a controllable voltage source $\underline{U}_c = U_c \angle \delta_c$ behind the phase reactor $\underline{Z}_c = R_c + \jmath X_c$, a susceptance B_f presenting the filter and a transformer represented as a complex impedance $\underline{Z}_{tf} = R_{tf} + \jmath X_{tf}$. $\underline{U}_s = U_s \angle \delta_s$ is the system voltage at the PCC and $\underline{U}_f = U_f \angle \delta_f$ is the filter bus voltage.

In MATACDC the possibility is included to omit the transformer impedance $\underline{Z}_{tf}$ and/or the filter susceptance B_f, leading to the different simplified situations depicted in Fig. 5. The model selection depends on the parameters specified by the user:

- **Full model** (Fig. 4)
- **No filter** ($B_f = 0$, Fig. 5a):
  - Omitting filters

(a) no filter

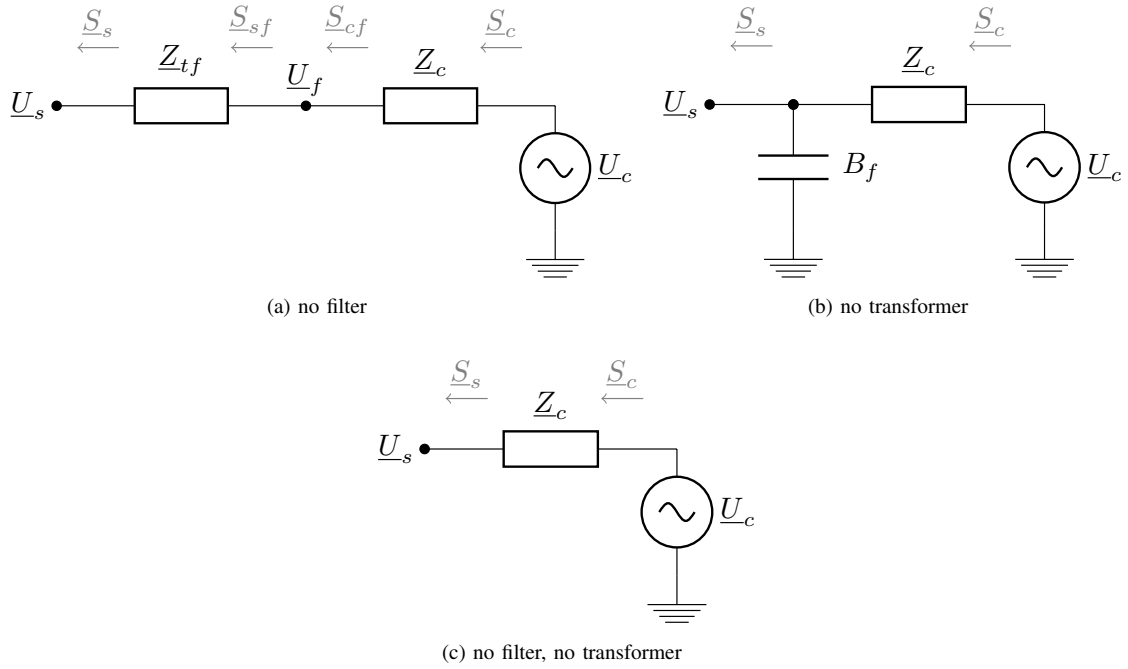(b) no transformer

(c) no filter, no transformer

Fig. 5. Single phase model of a converter station, (a) omitting the filter, (b) omitting the transformer and (c) omitting both filter and transformer.

   – Filterless design, e.g. modular multilevel converter topologies.

- **No transformer** ($Z_{tf} = 0$, Fig. 5b):

   – Transformerless design

- **No filter, no transformer** ($B_f = 0$ & $Z_{tf} = 0$, Fig. 5c):

   – Transformerless and filterless design

   – Simplified converter model (or pure inductive phase reactor when also omitting R_c).

Converter losses are taken into account using a generalised loss formula. The loss coefficients are defined independently for each converter in convdc. A distinction is made between rectifier and inverter operation.

2) Converter Control Representation: MATACDC includes three different converter representations with respect to the active power and two different representations with respect to the reactive power. These steady-state representations, depicted in Fig. 6 correspond to three distinctive active power control modes, namely

1) *P constant*: Constant active power injection P_s into the AC grid (Fig. 6a).

2) U_{dc} *constant*: Constant DC bus voltage U_{dc}, irrespective of the active power signal (Fig. 6b).

3) $P - U_{dc}$ *droop*: The active power injection into the HVDC system P_{dc} depends on DC bus voltage U_{dc} (Fig. 6c).

The droop control has setpoints for voltage and power, respectively $U_{dc,0}$ and $P_{dc,0}$, a voltage droop coefficient k_{dc} and can include a symmetric voltage deadband $\Delta U_{dc,0}$ (Fig. 6c), which can also be omitted.

The reactive power representation of the converter in MATACDC are:

1) *Q constant*: Constant reactive power injection Q_s.

2) *U constant*: Constant AC bus voltage magnitude U_s.

(a) P constant

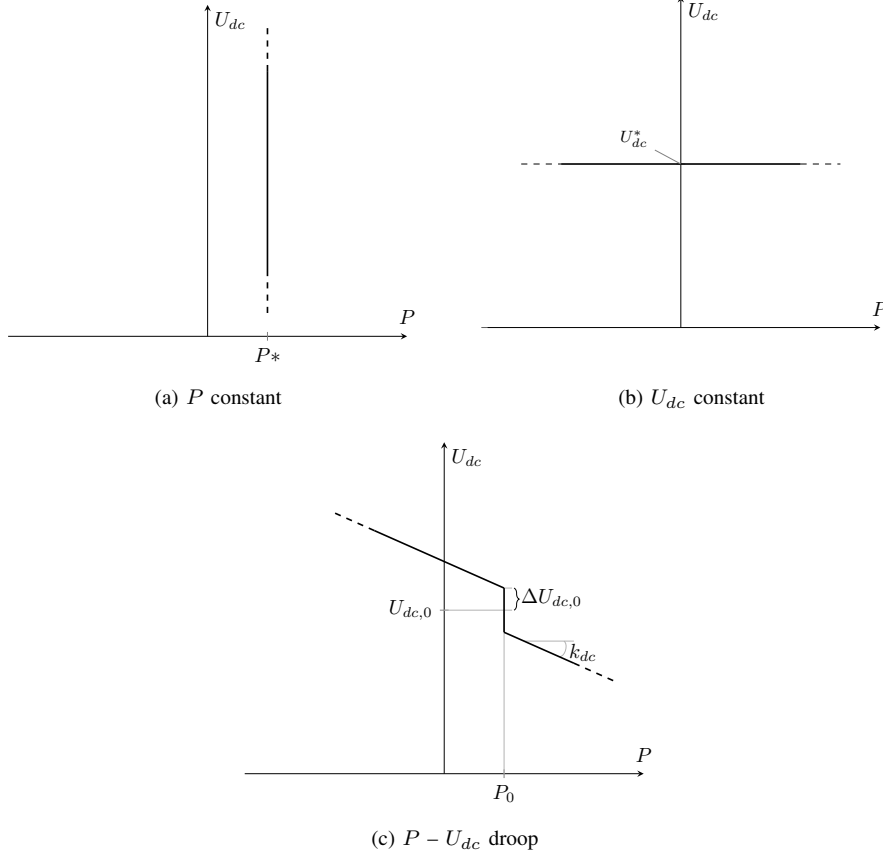(b) U_{dc} constant

(c) $P - U_{dc}$ droop

Fig. 6. Converter steady-state operation characteristics.

B. DC System Modelling

The DC networks is represented by a resistive network with current injections. The HVDC grid power flow equations can be written as

$$P_{dc_i} = p\, U_{dc_i} \sum_{\substack{j=1 \\ j \neq i}}^{n} Y_{dc_{ij}} \cdot (U_{dc_i} - U_{dc_j}), \tag{1}$$

with $Y_{dc_{ij}}$ equal to $1/R_{dc_{ij}}$ and $p = 1$ for a monopolar system or $p = 2$ for a monopolar symmetrically grounded or bipolar system. Combining these power injections for the n buses results in

$$\boldsymbol{P_{dc}} = p\, \boldsymbol{U_{dc}} \circ (\mathsf{Y}_{\mathsf{dc}} \boldsymbol{U_{dc}}), \tag{2}$$

with Y_{dc} is the DC bus matrix and $\circ$ the Hadamard or entrywise product.

In case of a $U_{dc} - P_{dc}$ droop with deadband, the relationship between the active power and the DC voltage from [16] can be generalised to

$$P_{dc_i} = P_{dc,0_i} - \frac{1}{k_{dc_i}}(U_{dc_i} - U'_{dc,0_i}), \tag{3}$$

with

$$U'_{dc,0_i} = \begin{cases} U^-_{dc,0_i} & \text{if } U_{dc_i} \leq U^-_{dc_i} \\ U_{dc_i} & \text{if } U^-_{dc_i} < U_{dc_i} < U^+_{dc_i} \\ U^+_{dc,0_i} & \text{if } U^+_{dc_i} \leq U_{dc_i} \end{cases}, \tag{4}$$

with $U^-_{dc,0_i} = U_{dc,0_i} - \Delta U_{dc,0}$ and $U^+_{dc,0_i} = U_{dc,0_i} + \Delta U_{dc,0}$.
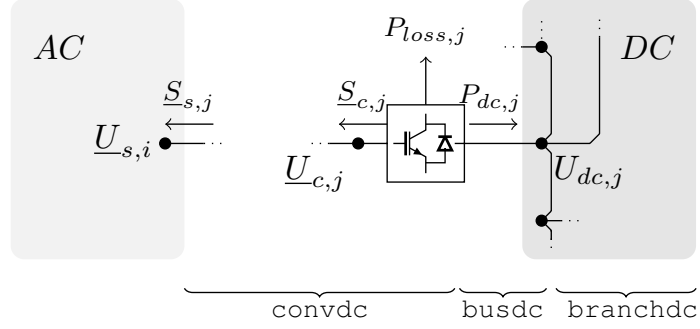
Fig. 7. AC/DC system interconnection.

IV. PROGRAM FLOW

To solve the AC/DC system, either a sequential [15] or combined power flow algorithm [17] can be used. MATACDC uses a sequential AC/DC power flow routine to solve the network. Fig. 7 summarises converter AC and DC powers, voltages and losses and how they are related to the data matrices. The AC representation of the converter stems with one of the implementations depicted in Figs. 4 – 5. The AC/DC power flow algorithm sequentially solves the AC and DC system power flow, thereby respectively keeping all converter powers and voltages constant. Fig. 8 shows the program flow structure. The remaining part of this section briefly addresses how MATACDC internally implements the sequential power flow algorithm.

In the mathematical description in this paper, the converter AC powers are considered to be positive when injected in the AC network. For convenience, DC powers are considered to be positive when injected into the DC system. In the MATACDC input and output format, however, the positive direction of all active powers stems with a power transfer from the DC to the AC system. The description in this section is limited to one HVDC grid and one AC grid for convenience, although the MATACDC algorithm does not pose any restrictions on the number of AC and HVDC grids.

A. AC Grids Power Flow

MATACDC relies on the MATPOWER `runpf` routine for the AC system power flow. There are two alterations to the standard usage of the `runpf` routine:

- **Non-synchronized zones** In case multiple AC grids are defined, MATACDC extracts the AC system data for each zone and solves the power flow of each AC grid one by one.
- **Infinite buses** In case only the DC system is of interest or in case an AC zone only has one node, infinite bus systems can be defined. When MATACDC runs the MATPOWER `runpf` routine these AC systems are left out of the analysis.

1) Initialization: To start the iteration, the system needs a first approximation of the converter active power injections, which can be done by assuming a lossless DC system and converters. Assuming the first converter to operate as a slack node, without lack of generality, we can write the vector of AC power injections as

$$\boldsymbol{P_s} = [\underbrace{P_{s_1}}_{\text{slack}}, \underbrace{P_{s_2} \ldots P_{s_m}}_{\text{voltage droop}}, \underbrace{P_{s_{m+1}} \ldots P_{s_k}}_{P-\text{control}}, \underbrace{\ldots 0}_{\text{outage}}]^T. \tag{5}$$
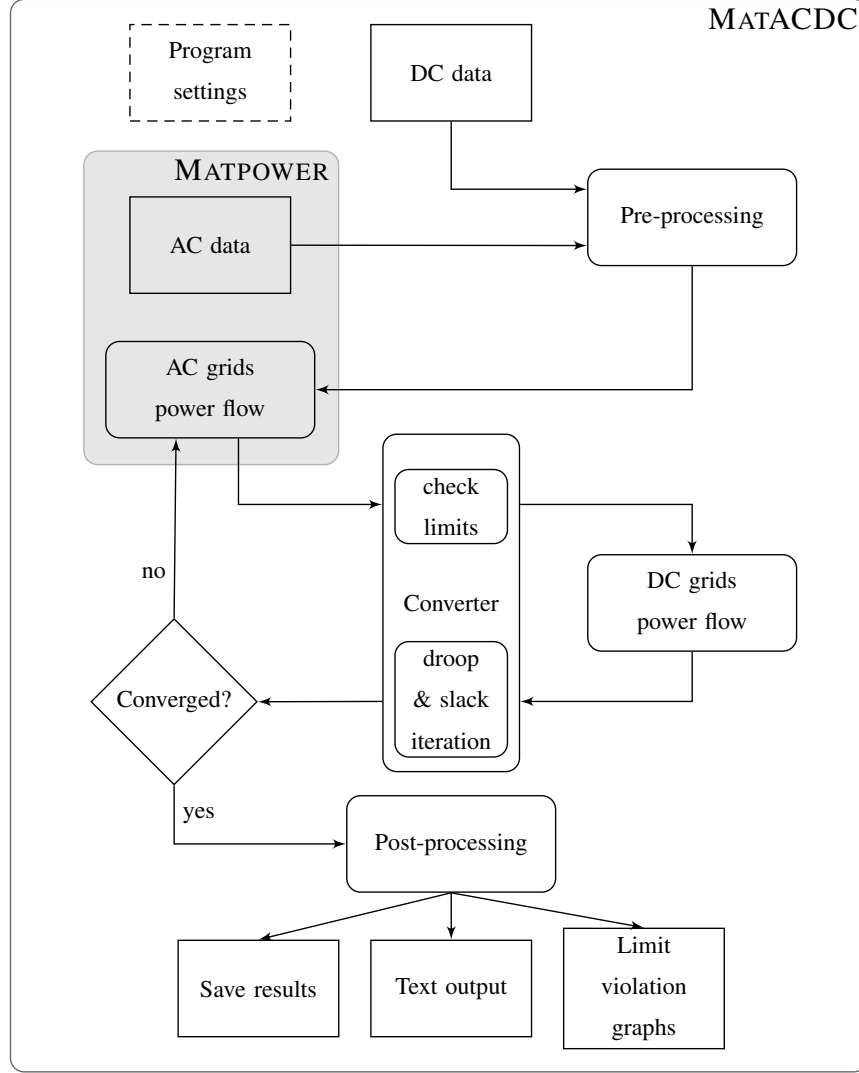
Fig. 8. MATACDC program flow.

The initialization of the active powers is given by

$$P_{s_i}^{(0)} = \begin{cases} -P_{dc,0_i} & \forall i\colon 2 \leq i \leq m \\ P_{s_i} & \forall i\colon m+1 \leq i \leq n \end{cases}. \tag{6}$$

The active power delivered by the DC slack bus is given by

$$P_{s_1}^{(0)} = -\sum_{i=2}^{m} P_{s_i}^{(0)} - \sum_{j=m+1}^{n} P_{s_j}, \tag{7}$$

or, in case of multiple converters with a constant voltage

$$P_{s_h}^{(0)} = -\frac{1}{l}\left(\sum_{i=l+1}^{m} P_{s_i}^{(0)} + \sum_{j=m+1}^{n} P_{s_j}\right) \quad \forall h\colon 1 \leq h \leq l. \tag{8}$$

Since multiple DC slack buses in a single HVDC grid might give rise to divergence in the DC power flow and are not likely to be implemented in real systems, this option has to be explicitly defined by the user. In general, a DC system can also operate with all converters under droop control, provided that not all converters include a deadband in the control.

Fig. 9. Representation of converters in the AC system.

2) Converter Representation: In the MATPOWER AC power flow routine `runpf`, the converters are represented as shown in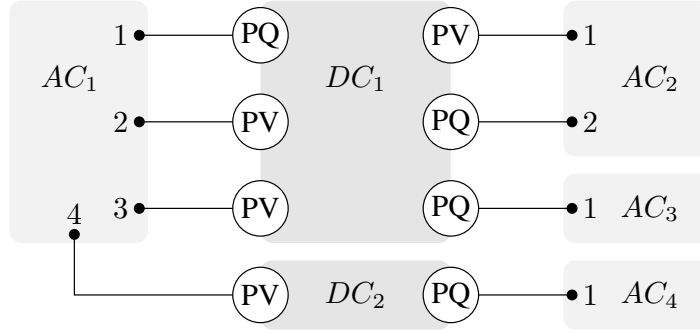 Fig. 9. Converters with a constant AC voltage are represented as dummy AC generators and their AC buses are changed from PQ-nodes to PV-nodes. Several situations can occur:

- No generator is present at the AC node: a dummy generator is added.
- A generator (PQ-node) is present: the node is changed to a PV-node and only the additional reactive power is considered to be delivered by the converter.
- A generator (PV-node) is present: the converter is set to constant reactive power control instead.

The converter power injections $\boldsymbol{P_s}$ and $\boldsymbol{Q_s}$ are included in the power mismatch vectors $\boldsymbol{\Delta P}^{(j)}$ and $\boldsymbol{\Delta Q}^{(j)}$ as negative loads

$$\Delta P_i^{(j)} = P_i^{gen} - (P_i^{dem} - P_{s_i}) - P_i(\boldsymbol{U}^{(j)}, \boldsymbol{\delta}^{(j)}), \tag{9}$$

$$\Delta Q_i^{(j)} = Q_i^{gen} - (Q_i^{dem} - Q_{s_i}) - Q_i(\boldsymbol{U}^{(j)}, \boldsymbol{\delta}^{(j)}). \tag{10}$$

B. Converter Limits

When the limits are enabled, the following situations may occur:

- When an **active power controlling converter** has an active power setpoint outside of the PQ-capability chart, the active power order is reduced to comply with the most stringent limit.
- When a **DC voltage droop controlling converter** hits a limit, the converter is set to a constant active power injection equal to the maximum converter active power limit.
- Similarly, when a **reactive power controlling converter** has a reactive power setpoint outside of the PQ-capability chart, the reactive power order is reduced to comply with the most stringent limit, provided that no active power limit is hit. If this happens to be the case, active power is prioritised when enforcing the limits.
- When an **AC voltage controlling converter** hits a reactive power limit, the converter is set to a constant reactive power injection determined by the most stringent limit. Similarly to the previous case, active power control is prioritised when the active power limit is violated.

The DC slack converter is excluded from the analysis, but is checked at the end of the power flow calculation. It is up to the user to redefine the converter setpoints when it is observed that a limit is hit in this converter.

C. HVDC Grid Power Flow

After calculating the AC power flow, all converter powers and losses are calculated to obtain the HVDC grid's injected powers $\boldsymbol{P_{dc}}$ for the k DC buses to which converters are connected. MATACDC uses a Newton-based iteration, based on (1) – (3) to calculate the HVDC grid power flow. For converters $m + 1$ to k the DC power injection P_{dc} is known as a result of the AC power flow. For the voltage droop converters 2 to m, only the DC power injection setpoints $P_{dc,0}$ are known. A modified active power vector $\boldsymbol{P'_{dc}}$ is defined such that

$$\boldsymbol{P'_{dc}} = [\underbrace{P_{dc_1}}_{\text{slack}}, \underbrace{P_{dc,0_2} \ldots P_{dc,0_m}}_{\text{voltage droop}}, \underbrace{P_{dc_{m+1}} \ldots P_{dc_k}}_{P-\text{control}}, \underbrace{\ldots 0}_{\text{outage}}]^T. \tag{11}$$

The DC bus voltages are calculated with a Newton-Raphson method

$$\left(\boldsymbol{U_{dc}} \frac{\partial \boldsymbol{P'_{dc}}}{\partial \boldsymbol{U_{dc}}}\right)^{(j)} \cdot \frac{\boldsymbol{\Delta U_{dc}}^{(j)}}{\boldsymbol{U_{dc}}} = \boldsymbol{\Delta P'_{dc}}^{(j)}, \tag{12}$$

with the modified power mismatch vector $\boldsymbol{\Delta P'_{dc}}^{(j)}$ given by

$$\Delta P_{dc_i}^{\prime(j)} = \begin{cases} P_{dc,0_i} - P_{dc,0_i}(\boldsymbol{U_{dc}}^{(j)}) & \forall i: \ 2 \le i \le m \\ P_{dc_i}^{(k)} - P_{dc_i}(\boldsymbol{U_{dc}}^{(j)}) & \forall i: \ m < i \le k \ , \\ -P_{dc_i}(\boldsymbol{U_{dc}}^{(j)}) & \forall i: \ k < i \le n \end{cases} \tag{13}$$

and $P_{dc,0_i}(\boldsymbol{U_{dc}}^{(j)})$ as

$$P_{dc,0_i}^{(j)} = P_{dc_i}(\boldsymbol{U_{dc}}^{(j)}) + \frac{1}{k_{dc_i}}(U_{dc_i}^{(j)} - U'_{dc,0_i}), \tag{14}$$

and superscripts (j) and (k) respectively referring to the inner NR iteration and the outer AC/DC power flow iteration.

D. DC Slack and Droop Buses Iteration

The AC bus active power injection P_s of the DC slack and droop buses is unknown and depends on the values of the DC power P_{dc} and the converter losses. As the converter losses depend on the yet unknown converter current, an additional iteration is needed to determine the active power injection P_s. If the converters are considered to be lossless, this additional iteration disappears.

V. USER-DEFINED FUNCTIONALITY

In MATACDC, it is relatively straightforward to add user-defined functions and models. As an illustration to an educational use of the program, we discuss how a DC/DC converter, a current-based droop converter and an alternative implementation of the slack bus can easily be added as models.

A. DC/DC Converter

In this example, we discuss a straightforward way to represent a DC/DC converter in the power flow algorithm, by making an analogy to the implementation of a combination of a line and a phase-shifting transformer in MATPOWER.

Fig. 10 represents the model of the DC/DC converter and the line, with N_{ij} the voltage ratio between the two sides of the DC/DC converter. The current injections can be written as a function of the voltages at the two ends as

$$\begin{bmatrix} I_{dc_i} \\ I_{dc_j} \end{bmatrix} = \mathsf{Y}_{\mathsf{dc,ij}} \begin{bmatrix} U_{dc_i} \\ U_{dc_j} \end{bmatrix}, \tag{15}$$
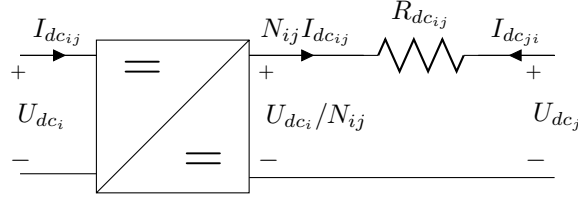
Fig. 10. DC line model with a DC/DC converter.

with the DC branch matrix $\mathsf{Y}_{\mathsf{dc,ij}}$ given by

$$\mathsf{Y}_{\mathsf{dc,ij}} = \begin{bmatrix} Y_{dc_{ij}}/N_{ij}^2 & -Y_{dc_{ij}}/N_{ij} \\ -Y_{dc_{ij}}/N_{ij} & Y_{dc_{ij}} \end{bmatrix}. \tag{16}$$

This can be implemented in the `makeYbusdc` routine by replacing the definition for the branch matrix elements `Ytt`, `Yff`, `Yft`, `Ytf` by

```
N = branchdc(:, 8);
Ytt = Ys;
Yff = Ys ./ N.^2;
Yft = - Ys ./ N;
Ytf = - Ys ./ N;
```

in which, as an example, column 8 of the `branchdc` matrix has temporarily been used to store the voltage ratios, with the default value for N_{ij} equal to 1. Alternatively to a constant voltage ratio, the converter can relatively easy be modeled as e.g. a constant voltage drop by rewriting the voltage ratio N_{ij} in terms of the fixed voltage drop $\Delta U_{DCDC_{ij}}$

$$N_{ij}^{(l)} = \left(1 - \frac{\Delta U_{DCDC_{ij}}}{U_{dc_i}^{(l)}}\right), \tag{17}$$

and adding an additional iteration to the DC network power flow, with (l) denoting this outer iteration index. The fixed voltage drops can e.g. temporarily be stored in column 7 of the `branchdc` matrix

```
dVdcdc = branchdc(:, 7);
```

The aforementioned outer iteration is added to the `runacdcpf` routine using

```
branchdc(:, 8) = 1./(1- dVdcdc./Vdc(branchdc(i,F_BUSDC)));
Ybusdc = makeYbusdc( busdc, branchdc );
[Vdc, Pdc] = dcnetworkpf(Ybusdc, Vdc, Pdc,...);
```

A DC/DC converter with constant current/power can be implemented either in a similar manner, or by splitting the DC network and defining opposite current (or alternatively power) references for the two buses.

B. Current-Based Voltage Droop Converter

MATACDC uses a power-based voltage droop control. The droop control can however relatively easy be changed to a current-based voltage droop control by using

$$I_{dc,0_i}^{(j)} = I_{dc_i}(\boldsymbol{U_{dc}}^{(j)}) + \frac{1}{k_{dc_i}}(U_{dc_i}^{(j)} - U_{dc,0_i}), \tag{18}$$

with

$$I_{dc,0_i} = \frac{P_{dc,0_i}}{pU_{dc,0_i}}, \tag{19}$$

the current reference. The DC current injections I_{dc_i} can be rewritten as

$$I_{dc_i} = \sum_{\substack{j=1 \\ j \neq i}}^{n} Y_{dc_{ij}} \cdot (U_{dc_i} - U_{dc_j}). \tag{20}$$

The modified power vector $\boldsymbol{P'_{dc}}$ from (11) is now replaced by a generalized vector with unknowns $\boldsymbol{X_{dc}}$ such that

$$\boldsymbol{X_{dc}} = [\underbrace{P_{dc_1}}_{\text{slack}}, \underbrace{I_{dc,0_2} \ldots I_{dc,0_m}}_{\text{voltage droop}}, \underbrace{P_{dc_{m+1}} \ldots P_{dc_k}}_{P-\text{control}}, \underbrace{\ldots 0}_{\text{outage}}]^T. \tag{21}$$

The DC power flow problem from (12) can now be rewritten as

$$\left(\boldsymbol{U_{dc}} \frac{\partial \boldsymbol{X_{dc}}}{\partial \boldsymbol{U_{dc}}} \right)^{(j)} \cdot \frac{\boldsymbol{\Delta U_{dc}}^{(j)}}{\boldsymbol{U_{dc}}} = \boldsymbol{\Delta X_{dc}}^{(j)}, \tag{22}$$

with the modified Jacobian elements given by

$$\left(U_{dc_j} \frac{\partial I_{dc,0_i}}{\partial U_{dc_j}} \right)^{(j)} = -U_{dc_j}^{(j)} Y_{dc_{ij}}, \tag{23}$$

$$\left(U_{dc_i} \frac{\partial I_{dc,0_i}}{\partial U_{dc_i}} \right)^{(j)} = \left(\sum_{\substack{j=1 \\ j \neq i}}^{n} Y_{dc_{ij}} + \frac{1}{k_{dc_i}} \right) U_{dc_i}^{(j)}. \tag{24}$$

The current-based droop characteristic can be implemented in the function `dcnetworkpf` by defining the new variable $\boldsymbol{X_{dc}}$ from (21) as

```
Xdc         = Pdc1;
Xdc(droop)=-Pdcset(droop)./(pol*Vdcset(droop));
```

Inside the DC network Newton-Raphson iteration, the values of $\boldsymbol{X_{dc}}$ are updated each iteration cycle

```
Pdccalc = pol*Vdc.*(Ybusdc*Vdc);
Xdccalc = Pdccalc;

Idccalc = Pdccalc./(pol*Vdc);
Xdccalc(droop) = Idccalc(droop)+...
  1./VIdroop(droop).*(Vdc(droop)-Vdcset(droop));
```

The modified Jacobian elements for the droop buses are given by (23) – (24)

```
J(droop,:)    = Ybusdc(droop,:).*...
              (ones(size(droop))*Vdc');
J(drooplidx) = J(drooplidx)+ ...
              1./VIdroop(droop).*Vdc(droop);
```

The incremental changes are solved by having

```
 dXdc = Xdc-Xdccalc;
 dV   = J\dXdc;
```

with \the MATLAB command to solve `J.dV = dXdc`. Also other droop implementations [18] and other converter control representations [19] can be accounted for similarly.

C. Predefined Average System Voltage

The slack bus in MATACDC has been implemented as a DC bus of which the voltage is known prior to the power flow. Alternatively, it is possible to include an additional equation to the DC system power flow and update the slack bus voltage each iteration cycle, e.g. to obtain a voltage profile resulting in an average system voltage equal to unity by defining

$$X_{dc_1} = \frac{1}{n} \sum_{j=1}^{n} U_{dc_j}, \tag{25}$$

assuming the first bus to be the slack bus. The Jacobian elements are given by

$$\left(U_{dc_j} \frac{\partial X_{dc_1}}{\partial U_{dc_j}} \right)^{(j)} = \frac{U_{dc_j}^{(j)}}{n}. \tag{26}$$

This is implemented in the function `dcnetwork` by predefining the slack bus element in the unknown vector $\boldsymbol{X_{dc}}$ from the previous example by

`Xdc(slack) = 1;`

In the NR iteration loop, the following command is added

```
if ~isempty(slack)
    J(slack,:) = Vdc/nb;
    Xdccalc(slack) = sum(Vdc)/nb;
end
```

D. Pilot Voltage Droop Control

In [20], it is discussed how a common pilot voltage can be used for droop control, as an alternative for only using a local voltage feedback. In [21], this concept has been used to distribute the power taking into account the available headroom at each converter. The concept of pilot voltage control can be accounted for in MATACDC by redefining the droop equation

$$P_{dc_i} = P_{dc,0_i} - \frac{1}{k_{dc_i}}(U_{dc}^+ - U_{dc,0}^+), \tag{27}$$

with U_{dc}^+ the pilot voltage. This leads to a redefinition of (14) such that

$$P_{dc,0_i}^{(j)} = P_{dc_i}(\boldsymbol{U_{dc}}^{(j)}) + \frac{1}{k_{dc_i}}(U_{dc}^{+\,(j)} - U_{dc,0}^+), \tag{28}$$

in the modified power mismatch vector $\boldsymbol{\Delta P'_{dc}}^{(j)}$ from (13) for $2 \leq i \leq m$. Assuming the voltage at node 2 U_{dc_2} to be used as pilot voltage, the Jacobian elements are redefined such that

$$\left(U_{dc_j} \frac{\partial P_{dc,0_i}}{\partial U_{dc_j}} \right)^{(j)} = \left(U_{dc_j} \frac{\partial P_{dc_i}}{\partial U_{dc_j}} \right)^{(j)}, \ \forall i\colon 2 < i \leq m \tag{29}$$

$$\left(U_{dc_2} \frac{\partial P_{dc,0_i}}{\partial U_{dc_2}} \right)^{(j)} = \left(U_{dc_2} \frac{\partial P_{dc_i}}{\partial U_{dc_2}} \right)^{(j)} + \frac{1}{k_{dc_i}} U_{dc_2}^{(j)}. \tag{30}$$

meaning that the contributions to the Jacobian from the droop-controlled buses, which are normally put on the diagonal, or now moved to the column corresponding to the pilot voltage, in this case column 2. In MATACDC,

this can be implemented by changing the function `dcnetworkpf` by replacing the calculation of the droop bus powers and the corresponding Jacobian elements as

```
Pdccalc(droop) = Pdccalc(droop) + ...
    1./PVdroop(droop).*(Vdcset(pilot)-Vdc(pilot));
J(droop,pilot)    = J(droop,pilot) +...
    1./PVdroop(droop).*Vdc(pilot);
```

with `pilot` the internal bus number of the pilot voltage controlled bus. Redefining the `dcnetworkpf` m-file, it is thus relatively straight-forward to include other HVDC grid control lay-outs and to study their effect on the power flow redistribution after an outage.

VI. CASE STUDY

MATACDC includes different case studies based on a 5-bus AC test system including a 3-terminal DC system from [16] and a modified version of the IEEE RTS test system extended with 2 MTDC systems from [15]. In [22], MATACDC has also been used to obtain steady-state values as a start for dynamic studies.

As an example, Fig. 11 shows the implementation of a hybrid AC/DC system, namely the CIGRE B4 DC grid test system [23]. The power flow results and inputs are based on the ongoing work in the CIGR Working Group B4-58.
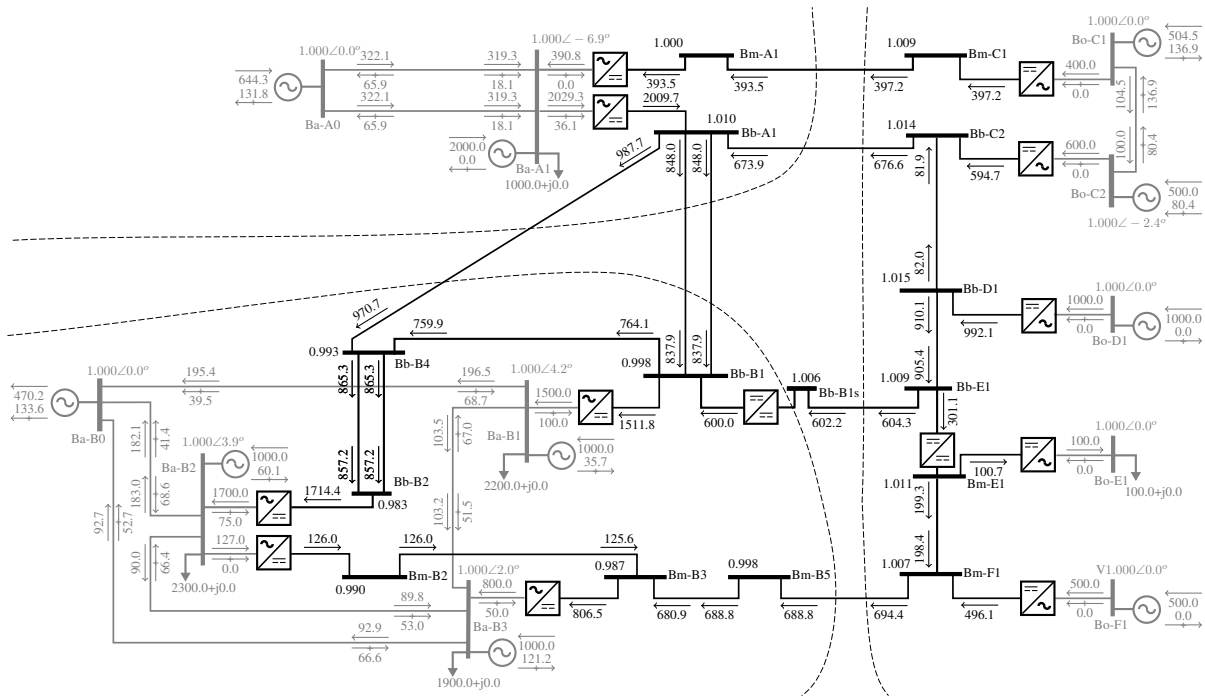

Fig. 11. MATACDC power flow solution of the CIGRE B4 DC Grid Test System. **Legend**: → Active power (MW) and ⇸ Reactive power (MVAr).

The HVDC networks (depicted in black) consist of a symmetric monopolar link (±200 kV), a meshed 5-terminal bipolar HVDC grid (±400 kV) and a symmetric monopolar radial multi-terminal system with 4 converters (±200 kV). The networks consist of both overhead lines and cables. The three AC networks (depicted

in gray) Ba-A0 to Ba-A1, Ba-B0 to Ba-B3 and Bo-C1 to Bo-C2 have operating voltages of respectively 380 kV, 380 kV and 145 kV. Buses Ba-A0, Ba-B0 and Bo-C1 have been chosen as the AC slack nodes for the different AC grids. The converter control modes, setpoints, loss model and other data are taken from [23].

The converter connected to Bm-A1 is under constant DC voltage control (slack node) and the AC/DC converters connected to Bm-C1, Bb-C2, Bb-D1, Bm-E1 and Bm-F1 have a constant active power injection. All remaining converters have a power-voltage droop [23]. The DC/DC converters has been modeled as constant voltage ratio to obtain a steady-state power flows of respectively 600 MW for the converter between Bb-B1 and Bb-B1s and 300 MW for the converter between Bm-E1 and Bb-E1. The modelling corresponds to the one discussed in Section V. The AC buses Bo-D1, Bo-E1 and Bo-F1 have been modeled as infinite buses.

In this example, the program needs three overall iterations to converge. When the AC grid is left out of the analysis by replacing all AC converter buses by infinite buses, the execution time roughly changes to about 74%.

VII. CONCLUSION

In this paper, the software program MATACDC has been introduced. This MATLAB-based program is the first of its kind by providing an open source tool to study steady-state interactions in hybrid AC/DC system for a variety of converter control strategies and contingencies. The program allows to study non-synchronised AC grids interconnected by different DC systems. MATACDC is flexible and seamlessly integrated with the AC power flow routines in MATPOWER. The code has been written to provide a maximum user flexibility to define the AC and HVDC grids with arbitrary topologies, while keeping the code easy to understand. Different examples show that user-added functions can easily be embedded, enabling the steady-state study of various system control strategies. The case study shows that the program can be used to simulate complex AC/DC systems.

ACKNOWLEDGEMENT

REFERENCES

[1] D. Van Hertem and M. Ghandhari, "Multi-terminal VSC HVDC for the European supergrid: Obstacles," *Renewable and Sustainable Energy Reviews*, vol. 14, no. 9, pp. 3156–3163, Dec. 2010.

[2] T. M. Haileselassie and K. Uhlen, "Impact of DC line voltage drops on power flow of MTDC using droop control," *IEEE Trans. Power Syst.*, vol. 27, no. 3, pp. 1441–1449, Aug. 2012.

[3] J. Beerten and R. Belmans, "Analysis of power sharing and voltage deviations in droop-controlled DC grids," *IEEE Trans. Power Syst.*, vol. 28, no. 4, pp. 4588–4597, 2013.

[4] M. Aragüés-Peñalba, A. Egea-Àlvarez, S. G. Arellano, and O. Gomis-Bellmunt, "Droop control for loss minimization in HVDC multi-terminal transmission systems for large offshore wind farms," *Electric Power Systems Research*, vol. 112, pp. 48 – 55, 2014.

[5] M. Baradar, M. Hesamzadeh, and M. Ghandhari, "Second-order cone programming for optimal power flow in VSC-type AC-DC grids," *IEEE Trans. Power Syst.*, vol. 28, no. 4, pp. 4282–4291, Nov. 2013.

[6] R. Wiget and G. Andersson, "DC optimal power flow including HVDC grids," in *Proc. IEEE EPEC 2013*, Aug. 2013, 6 pages.

[7] J. Rimez, "Optimal operation of hybrid AC/DC meshed grids," Ph.D. dissertation, University of Leuven (KU Leuven), 2014.

[8] A.-K. Marten and D. Westermann, "Schedule for converters of a meshed HVDC grid and a contingency schedule for adaption to unscheduled power flow changes," in *Proc. IEEE PES GM 2013*, Vancouver, Canada, Jul. 2013, 5 pages.

[9] R. D. Zimmerman, C. E. Murillo-Sanchez, and R. J. Thomas, "MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education," *IEEE Trans. Power Syst.*, vol. 26, no. 1, pp. 12–19, Feb. 2011.

[10] O. Gomis-Bellmunt, J. Liang, J. Ekanayake, and N. Jenkins, "Voltage-current characteristics of multiterminal HVDC-VSC for offshore wind farms," *Electric Power Systems Research*, vol. 81, pp. 440–450, 2011.

[11] T. Haileselassie, K. Uhlen, and T. Undeland, "Control of multiterminal HVDC transmission for offshore wind energy," in *Proc. Nordic Wind Power Conference Presentation*, Rønne, Denmark, Sep. 10–11, 2009, 7 pages.

[12] MatACDC website. [Online]. Available: http://www.esat.kuleuven.be/electa/teaching/matacdc/

[13] MATPOWER website. [Online]. Available: http://www.pserc.cornell.edu/matpower/

[14] S. Cole and R. Belmans, "MatDyn, a new Matlab-based toolbox for power system dynamic simulation," *IEEE Trans. Power Syst.*, vol. 26, no. 3, pp. 1129–1136, 2011.

[15] J. Beerten, S. Cole, and R. Belmans, "Generalized steady-state VSC MTDC model for sequential AC/DC power flow algorithms," *IEEE Trans. Power Syst.*, vol. 27, no. 2, pp. 821–829, May 2012.

[16] J. Beerten, D. Van Hertem, and R. Belmans, "VSC MTDC systems with a distributed DC voltage control – a power flow approach," in *Proc. IEEE PowerTech '11*, Trondheim, Norway, Jun. 19–23, 2011, 6 pages.

[17] M. Baradar and M. Ghandhari, "A multi-option unified power flow approach for hybrid AC/DC grids incorporating multi-terminal VSC-HVDC," *IEEE Trans. Power Syst.*, vol. 28, no. 3, pp. 2376–2383, Aug 2013.

[18] W. Wang, M. Barnes, and O. Marjanovic, "Droop control modelling and analysis of multi-terminal VSC-HVDC for offshore wind farms," in *Proc. IET ACDC 2012*, Birmingham, UK, Dec. 4–6, 2012, 6 pages.

[19] T. K. Vrana, J. Beerten, R. Belmans, and O. B. Fosso, "A classification of DC node voltage control methods for HVDC grids," *Electric Power System Research*, vol. 103, pp. 137–144, Oct. 2013.

[20] B. Berggren, K. Lindén, and R. Majumder, "DC grid control through the pilot voltage droop concept – methods for establishing set-point tracking," in *Proc. IEEE EnergyCon 2014*, Dubrovnik, Croatia, May 2014, 8 pages.

[21] N. R. Chaudhuri and B. Chaudhuri, "Adaptive droop control for effective power sharing in multi-terminal DC (MTDC) grids," *IEEE Trans. Power Syst.*, vol. 28, no. 1, pp. 21–29, Feb. 2013.

[22] J. Beerten, S. Cole, and R. Belmans, "Modeling of multi-terminal VSC HVDC systems with distributed DC voltage control," *IEEE Transactions on Power Systems*, vol. 29, no. 1, pp. 34–42, Jan. 2014.

[23] T. K. Vrana, S. Dennetière, Y. Yang, J. Jardini, D. Jovcic, and H. Saad, "The CIGRE B4 DC grid test system," *ELECTRA*, no. 270, pp. 10–19, Oct. 2013.