MATACDC 1.0
User's Manual

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

July 4, 2012

www.esat.kuleuven.be/electa/teaching/matacdc/

Contents

1 Introduction

MATACDC is a free MATLAB based open source program for AC/DC power flow analysis. The program uses a sequential AC/DC power flow algorithm and can be used to simulate interconnected AC systems and Multi-terminal Voltage Source Converter High Voltage Direct Current (VSC HVDC) systems. With respect to the AC system power flow, the program completely relies on MATPOWER, a power flow and optimal power flow program in MATLAB [1]. 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. The package has been fully integrated with the existing AC power flow routines developed in MATPOWER, while keeping the MATPOWER original source code unaltered.

MATACDC shares the same philosophy as MATPOWER and MATDYN [2, 3]: "It is intended as a simulation tool for researchers and educators that is easy to use and modify." [4]. The source code of MATACDC is available [5]. The code is well documented, structured and easy to understand and modify.

1.1 License

- MATACDC is free and open source software.
- MATACDC may be modified for personal use provided this license remains in force.
- MATACDC comes with no warranty whatsoever; not even the implied warranty of merchantability or fitness for a particular purpose.
- Any publications derived from the use of MATACDC must acknowledge MATACDC.
- MATACDC may not be redistributed without prior written permission.
- Modified versions of MATACDC, or works derived from MATACDC, may not be distributed without prior written permission.

1.2 Citing MATACDC

It is requested that publications using MATACDC explicitly acknowledge this by including a reference to the MATACDC-website. `www.esat.kuleuven.be/electa/teaching/matacdc/`

2 Getting started

This chapter briefly describes how to install MatACDC, how to run a simulation, change the input data and access the results. This chapter is merely intended for a first understanding of how to start working with the software. The different functions and components are explained in more detail in the subsequent chapters and can be accessed in MATLAB using the `help` command line.

2.1 Installation

In order to run the MATACDC files, it is required to have the following programs installed on your computer:

- MATLAB must be installed
- The open source power program MATPOWER must be installed and added to the MATLAB path. MATPOWER is available at `http://www.pserc.cornell.edu/matpower/`. Please consult the MATPOWER manual for more info on how to install the latest stable version of the program. MATACDC has been tested with the latest stable version MATPOWER 4.1.
- MATACDC must be installed[1], and preferably added to the MATLAB path. It is also advised to add the sub-directories `.../Cases/PowerflowAC` and `.../Cases/PowerflowDC` to the path.

To check whether all program components are installed correctly, you can run the `test_acdcpf.m` file.

2.2 Running a simulation

The MATACDC power flow routine is started by the command `runacdcpf`:

```
>> runacdcpf(...);
```

The `runacdcpf` command needs two types of input data: The MATPOWER AC power flow data and the MATACDC DC system data.

```
>> runacdcpf('case5_stagg','case5_stagg_MTDCslack');
```

2.3 Changing the input data

Similar to using the `loadcase` command with a MATPOWER case file, the dc power flow data can be loaded into a struct with the `loadcasedc` command.

```
>> mpcdc = loadcasedc(casefiledc);
```

It is also possible to load the individual data matrices by using

[1] MATACDC is available at `www.esat.kuleuven.be/electa/teaching/matacdc/`

```
>> [basemvaac, basemvadc, pol, busdc, convdc, branchdc] = loadcasedc(casefiledc);
```

By loading the input files first, it is possible to change the data:

```
>> define_constantsdc;
>> mpcdc = loadcasedc('case5_stagg_MTDCslack');
>> mpcdc.convdc(3,CONVTYPE_AC) = PVC;
>> runacdcpf('case5_stagg',mpcdc);
```

Similar to the `define_constants` script in MATPOWER, `define_constantsdc` loads the names of the column indices of the data matrices `busdc`, `convdc` and `branchdc`. In the example above, the third converter in the `case5_stagg_MTDCslack` file is changed to AC voltage control before starting the AC/DC power flow simulation.

2.4 Using the results

The results can be easily accessed after the AC/DC power flow:

```
>> [resultsac, resultsdc] = runacdcpf(casefileac,casefiledc);
```

The data struct `resultsac` contains all AC power flow results, and uses the standard MATPOWER data struct format, limited to the fields `baseMVA`, `bus`, `gen` and `branch`. The data struct `resultsdc` contains all DC power flow data.

Similarly to changing the input files in the example in the previous section, one can access the individual data elements using the named column indices:

```
>> define_constantsdc;
>> [mpc, mpcdc] = runacdcpf('case5_stagg','case5_stagg_MTDCslack');
>> Vdc = mpcdc.busdc(:,VDC);
```

In this example, the voltages at the different DC buses are stored in the vector `Vdc`.

2.5 Power flow options

When using the command lines from the previous sections, MATACDC uses the default program options. It is also possible to run a power flow with altered program settings.

```
>> [resultsac, resultsdc] = runacdcpf(casefileac,casefiledc,optionsdc);
```

The command `macdcoption` is used to define the default program options. The indices of the resulting option vector are defined in the file and can be easily accessed using the `help` command.

```
>> opt = macdcoption;
>> opt(10) = 1;     % enforce converter current and voltage limits
>> runacdcpf('case5_stagg','case5_stagg_MTDCslack',opt);
```

In this example, the converter limits are enforced during the power flow. As a result, the reactive power limit in converter 1 is hit and the converter control is changed from AC voltage control to constant Q injection.

3 System Input Data Format

This chapter introduces the hybrid AC/DC system from a conceptual point of view and discusses the DC system input format. The AC system data input format is the standard MATPOWER format. The DC system data is divided into three matrices: `busdc`, `convdc` and `branchdc`. This chapter discusses the DC system in terms of these three matrices and in terms of the AC/DC system interconnections.

For more details on the modeling of the system itself, the reader is referred to chapter 4.

3.1 AC/DC System

In general, a hybrid AC/DC system can have a structure similar to the one depicted in Fig. 2. 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

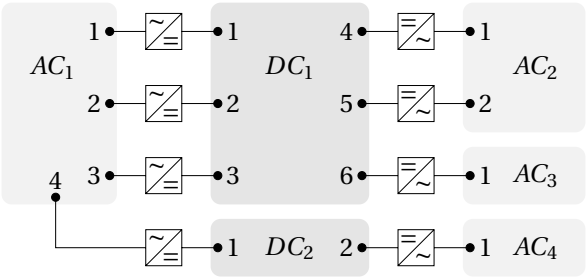


Figure 1: Interconnected AC/DC System

has to be limited to these interconnected buses. In this example, AC systems 3 and 4 only share one converter with respectively DC 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 1 interconnection with the other AC systems in the grid.

3.2 AC/DC System Data Representation

MATACDC uses three distinctive data matrices to store the information on the DC system and the interconnection with the AC system. The choice of these three data matrices is in analogy with the subdivision in the `bus`, `gen` and `branch` matrices in AC power flow. Fig. 2 represents this subdivision. The three data matrices are:

- `busdc` matrix: Contains all DC bus data (e.g. bus numbers, active power withdrawals, voltages, ...)
- `convdc` matrix: Contains all converter station data (e.g. loss data, impedance values, status, control modes, ...)
- `branchdc` matrix: Contains all DC branch data (e.g. line resistance, bus numbers, ...)

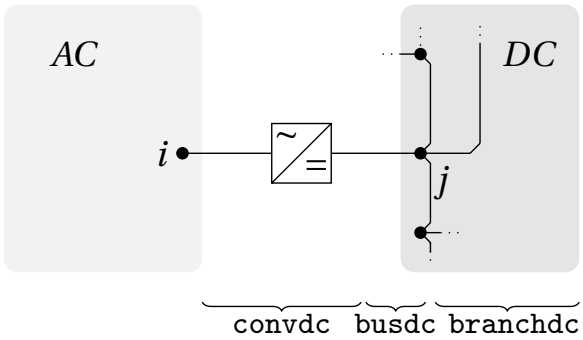


Figure 2: AC/DC System Implementation: Conceptual representation

This flexible definition of the DC system allows a generalized implementation of DC systems with multiple connections to different AC power systems as well as DC buses without connections to the AC system.

3.2.1 DC bus matrix

The different indices of the DC bus matrix `busdc` are given in Tab. 1.

Table 1: busdc data format

Index	NAME	Description
1	BUSDC_I	DC bus number
2	BUSAC_I	Corresponding AC bus number
		Note: Index BUSAC_I = 0 indicates no AC bus connection
3	GRIDDC	DC grid to which the DC bus is connected
4	PDC	P_{dc}, power withdrawn from the DC grid (MW)
5	VDC	U_{dc}, DC voltage (p.u.)
6	BASE_KVDC	base DC voltage (kV)
7	VDCMAX	maximum DC voltage (p.u.)
8	VDCMIN	minimum DC voltage (p.u.)
9	CDC	C_{dc}, DC bus capacitor size (p.u.) (not used in power flow)

Each DC bus can either have a corresponding AC bus or can not be connected to the AC system, as shown in Fig. 3. This example shows a topology similar to the one from Fig. 2, but with an arbitrary AC bus numbering and a consecutive DC bus numbering in the two DC systems. The system also includes a DC bus without a connection to the AC grid (DC bus 7).

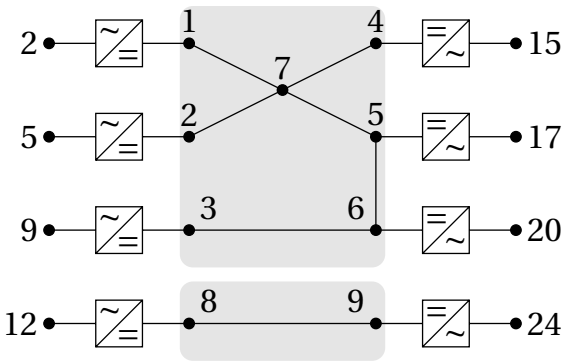


Figure 3: DC system bus numbering: example

The representation of the AC bus numbers, DC bus numbers and the DC grid index in the busdc matrix for this example is given by

```
%%          busdc_i  busac_i  griddc  ...
busdc = [      1        2        1     ...;
               2        5        1     ...;
               3        9        1     ...;
               4       15        1     ...;
               5       17        1     ...;
               6       20        1     ...;
               7        0        1     ...;
               8       12        2     ...;
               9       24        2     ...;] ;
```

The BUSAC_I = 0 indicates that their is no corresponding AC bus in any of the AC systems.

The AC buses can be interconnected by the same AC grid, or can alternatively be part of multiple non-synchronized zones. This can be accomplished by selecting different ZONE values in the MATPOWER bus matrix.

3.2.2 DC converter matrix

The different indices of the DC converter matrix convdc are given in Tab. 2. This convdc matrix contains all converter station data, such as losses, impedance values, status, control modes. More details on the different control options, the input- and the output data and the modelling can be found in chapter 4.

Table 2: convdc data format

Index	NAME	Description
Converter buses, control parameters and AC system data		
1	CONV_BUS	converter bus number (DC bus numbering)
2	CONVTYPE_DC	DC bus type (1 = constant power, 2 = DC slack, 3 = DC droop)
3	CONVTYPE_AC	AC bus type (1 = PQ, 2 = PV)
4	PCONV	P_s, active power injected in the AC grid (MW)

Table 2: convdc data format (continued)

Index	NAME	Description
5	QCONV	Q_s, reactive power injected in the AC grid (MVAr)
6	VCONV	U^{ref}, target voltage of converter connected AC bus (p.u.)
Impedance values		
7	RTF	R_{tf}, transformer resistance (p.u.)
8	XTF	X_{tf}, transformer reactance (p.u.)
9	BF	B_f, filter susceptance (p.u.)
10	RCONV	R_c, phase reactor resistance (p.u.)
11	XCONV	X_c, phase reactor reactance (p.u.)
12	BASEKVC	converter AC base voltage (kV)
13	VCMAX	$U_{c,max}$, maximum converter voltage magnitude (p.u.)
14	VCMIN	$U_{c,min}$, minimum converter voltage magnitude (p.u.)
15	ICMAX	I_{max}, maximum converter current (p.u.)
16	CONVSTATUS	converter status (1 = on, 0 = off)
Converter loss data		
17	LOSSA	a, constant loss coefficient (MW)
18	LOSSB	b, linear loss coefficient (kV)
19	LOSSCR	c_{rec}, rectifier quadratic loss coefficient (Ω)
20	LOSSCI	c_{inv}, inverter quadratic loss coefficient (Ω)
DC voltage droop constants (optional)		
21	DROOP	k, DC voltage droop (MW/p.u)
22	PDCSET	$P_{dc,set}$, voltage droop power set-point (MW)
23	VDCSET	$U_{dc,set}$, voltage droop voltage set-point (p.u.)
24	DVDCSET	$\Delta U_{dc,set}$ voltage droop deadband (p.u.) (optional)
Columns typically added after the power flow		
25	VMC	U_c, converter voltage magnitude (p.u.)
26	VAC	δ_c, converter voltage angle (degrees)
27	PCCONV	P_c, active power generated by the converter (MW)
28	QCCONV	Q_c, reactive power generated by the converter (MVAr)
29	PCLOSS	P_{loss}, active converter power losses (MW)
30	VMF	U_f, filter voltage magnitude (p.u.)
31	VAF	δ_f, filter voltage angle (degrees)
32	PFIL	P_{sf} or P_{cf}, active power at filter bus (MW)
33	QCONVF	Q_{sf}, reactive power through transformer at filter bus (MVAr)
34	QCCONVF	Q_{cf}, reactive power through reactor at filter bus (MVAr)

The CONV_BUS numbering corresponds to the BUSDC_I numbers defined in the busdc matrix.

The effect of a converter outage on the power flows in the AC/DC system can easily be adressed by changing the CONVSTATUS flag from 1 to 0. Internally, MATACDC handles DC buses without an AC connection differently from buses with a connection. For more information, please consult section 3.3.

Additionally, a number of named indices are defined for CONVTYPE_DC and CONVTYPE_AC, depending on the type of control. These are summarized in Tab. 3.

Table 3: convdc named constants

Index	NAME	Description
CONVTYPE_DC constants		
3	DCDROOP	DC voltage droop
2	DCSLACK	DC slack bus
1	DCNOSLACK	constant active power bus
CONVTYPE_AC constants		
2	PVC	constant voltage converter control
1	PQC	constant reactive power converter control

3.2.3 DC branch matrix

The different indices of the DC branch matrix branchdc are given in Tab. 4. Similar to a converter, a branch outage can be studied by having BRDC_STATUS equal to 0. In the DC system power flow, the only line parameters taken into account are $R_{dc_{ij}}$, the resistances of the lines.

Table 4: busdc data format

Index	NAME	Description
1	F_BUSDC	i, from bus number
2	T_BUSDC	j, to bus number
3	BRDC_R	$R_{dc_{ij}}$, resistance (p.u.)
4	BRDC_L	$L_{dc_{ij}}$, inductance (p.u./s) (not used in power flow)
5	BRDC_C	$C_{dc_{ij}}$, total line charging capacity (p.u.*s) (not used in power flow)
6	RATEDC_A	rateA, MVA rating A (long term rating, not used)
7	RATEDC_B	rateB, MVA rating B (short term rating, not used)
8	RATEDC_C	rateC, MVA rating C (emergency rating, not used)
9	BRDC_STATUS	initial branch status, (1 - in service, 0 - out of service)
Columns typically added after the power flow		
10	PFDC	$P_{dc_{ij}}$, real power injected at "from" bus end (MW)
11	PTDC	$P_{dc_{ji}}$, real power injected at "to" bus end (MW)

3.3 Bus renumbering

To allow a maximum flexibility towards the end user, the program allows having an arbitrary DC bus number associated with any AC bus number. Internally, the bus indices are redefined so that AC and DC buses that are interconnected have the same bus numbers as the double numbering

hampers a straight-forward implementation of the power flow algorithm. The MATACDC functions `ext2intdc` and `ext2intac` are used to redefine the bus numbers. Similarly, the functions `int2extac` and `int2extdc` are used to convert the internal bus data to the external bus numbers.

Before renumbering the buses, converters facing an outage are treated similar to existing DC buses without a connection to the AC system by using the function `convout`. The function sets `BUSAC_I = 0` in `busdc` for the converters that have `CONVSTATUS = 0` flag, and also splits the converter matrix into operational and non-operational converters. Similarly, the functions `brchdcout` and `brchout` remove the non-operational DC and AC branches from the analysis.

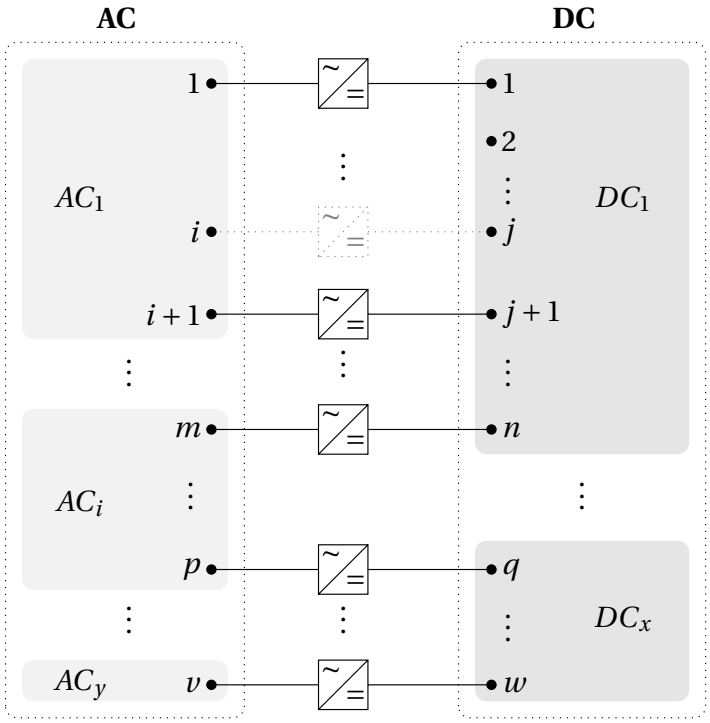


Figure 4: External AC/DC System interconnections.

Fig. 4 shows the generalized case the interconnection of y AC grids with a total of v buses and x DC grids with a total of w buses. Without lack of generality, we assume the last AC bus v to be connected to the x-th DC grid. With $v > w$, there are $v - w$ AC buses without a connection to a DC grid. The DC buses have been numbered in a logical order for convenience, although the algorithm does not impose any restrictions to the numbering of the DC buses as such. 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) and DC buses facing a converter outage (e.g. DC bus j in grid 1). Similarly, each AC bus can either have a connection to the DC system, or can be an AC bus without an interconnection.

The MATACDC bus numbering, allowing for different indices for AC and DC buses result in a maximum flexibility for the user to enter the different grid layouts in the power flow algorithm. After a per unit conversion of the AC and DC grid quantities, and the removal of converters and branches facing an outage, the overall sorting performed by the functions `ext2intdc` and `ext2intac` leads to the result in Fig. 5.

The DC buses are generally not consecutively numbered, contrary to what was shown in 4 for

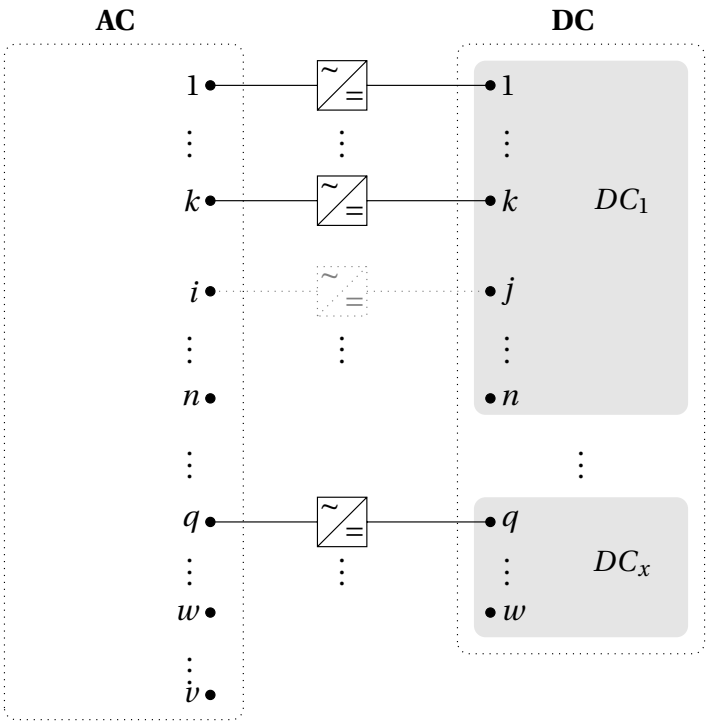


Figure 5: Internal AC/DC System interconnections after bus renumbering.

convenience. After renumbering the DC buses and sorting them per DC grid, the algorithm groups converter-connected DC buses and renames their AC buses correspondingly, e.g. buses 1 to k for the first DC grid in Fig. 5. This is done by the function `ext2intdc`.

Thereafter, arbitrary AC buses without DC connections are assigned to the DC buses without an AC connection, e.g. buses $k+1$ to n in Fig. 5. This is done by the function `ext2intac`. All converter outages are considered equally to DC buses without a converter. The renumbering allows an easy and straightforward access to both AC grid and DC grid variables during the power flow without compromising the flexibility of the user interface.

As indicated in Fig. 5, MATACDC does not take the AC zones into account when resorting the AC/DC system, as MATPOWER internally sorts the buses for each AC zone when an AC power flow is performed.

4 Modeling

This chapter introduces the modeling of the AC/DC system, from the software user's perspective. while linking the models to the implementation in MATACDC and to the DC system data matrices defined in chapter 3. For more details on the modeling itself, the reader is referred to [6], [7].

4.1 Converters

A general representation of a VSC HVDC converter station, showing the different components is depicted in Fig. 6. As seen from the AC point of common coupling (PCC), the different components

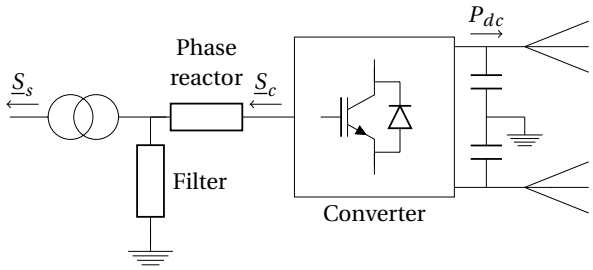


Figure 6: VSC HVDC converter station.

are [6]:

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

4.1.1 AC side model

In the most general format, the AC side of the converter is represented as depicted in Fig.7. The

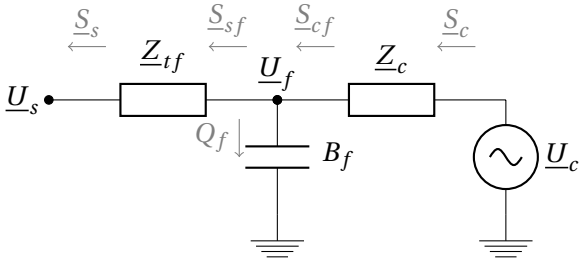


Figure 7: Equivalent single phase power flow model of a converter station connected to the AC grid.

model consists of a controllable voltage source $\underline{U}_c = U_c \angle \delta_c$ behind the phase reactor, represented as a complex impedance $\underline{Z}_c = R_c + \jmath X_c$. The low pass filter from Fig. 6 is represented as a susceptance

B_f at system frequencies. A transformer connects the filter bus to the AC grid and is represented by its complex impedance $\underline{Z}_{tf} = R_{tf} + \jmath X_{tf}$. The equations for active and reactive power at the grid side in terms of the complex voltages are

$$P_s = -U_s^2 G_{tf} + U_s U_f [G_{tf} \cos(\delta_s - \delta_f) + B_{tf} \sin(\delta_s - \delta_f)], \tag{1}$$
$$Q_s = \; U_s^2 B_{tf} + U_s U_f [G_{tf} \sin(\delta_s - \delta_f) - B_{tf} \cos(\delta_s - \delta_f)], \tag{2}$$

with $\underline{U}_s = U_s \angle \delta_s$ and $\underline{U}_f = U_f \angle \delta_f$ respectively the complex grid side and filter bus voltage. The equations at the converter side are

$$P_c = \; U_c^2 G_c - U_f U_c [G_c \cos(\delta_f - \delta_c) - B_c \sin(\delta_f - \delta_c)], \tag{3}$$
$$Q_c = -U_c^2 B_c + U_f U_c [G_c \sin(\delta_f - \delta_c) + B_c \cos(\delta_f - \delta_c)], \tag{4}$$

The reactive power at the filter is given by

$$Q_f = -U_f^2 B_f, \tag{5}$$

while the expressions for the filter side complex power flowing through the transformer are written as

$$P_{sf} = \; U_f^2 G_{tf} - U_f U_s [G_{tf} \cos(\delta_s - \delta_f) - B_{tf} \sin(\delta_s - \delta_f)], \tag{6}$$
$$Q_{sf} = -U_f^2 B_{tf} + U_f U_s [G_{tf} \sin(\delta_s - \delta_f) + B_{tf} \cos(\delta_s - \delta_f)], \tag{7}$$

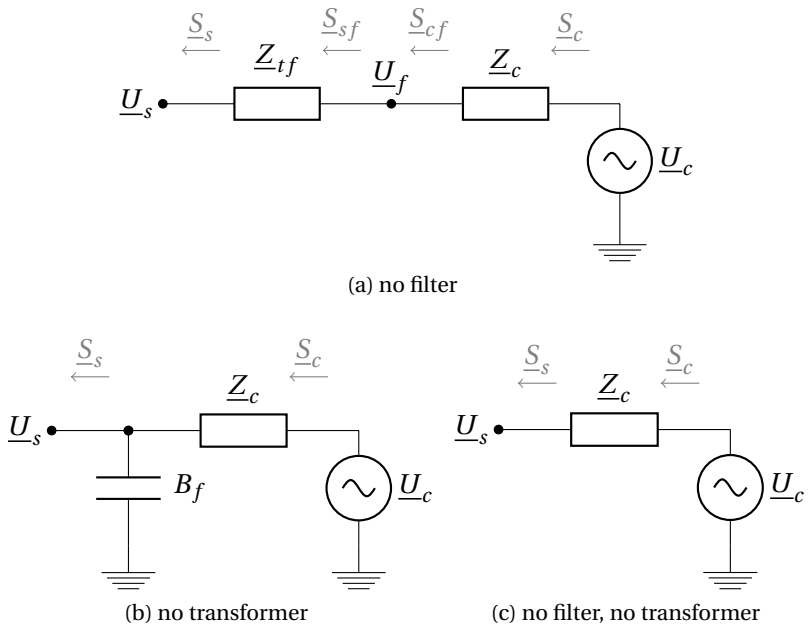
and those flowing through the phase reactor side are

$$P_{cf} = -U_f^2 G_c + U_f U_c [G_c \cos(\delta_f - \delta_c) + B_c \sin(\delta_f - \delta_c)], \tag{8}$$
$$Q_{cf} = \; U_f^2 B_c + U_f U_c [G_c \sin(\delta_f - \delta_c) - B_c \cos(\delta_f - \delta_c)]. \tag{9}$$

In this case, the user has to specify values for R_{tf}, X_{tf}, B_f, R_c and X_c. In MATACDC, both the transformer impedance $\underline{Z}_{tf}$ and the filter susceptance B_f can be omitted, leading to the different situations depicted in Fig. 8. Similarly, it is equally possible to only omit R_{tf} and/or R_c.

Depending on the parameters specified by the user, the MATACDC implementation corresponds to one of the situations depicted in Figs. 7 – 8. These implementations can be considered in the following situations:

- **No filter** ($B_f = 0$):
  - Omitting filters
  - Filterless design: Multilevel (modular) converter topologies. Contrary to earlier 2- or 3-level converter topologies using Pulse Width Modulation (PWM), the latest converter configurations synthesize the AC voltage waveform using a multilevel converter and do not require low-pass AC filters.
- **No transformer** ($Z_{tf} = 0$):
  - Transformerless design
- **No filter, no transformer** ($B_f = 0$ & $Z_{tf} = 0$):



(a) no filter

(b) no transformer

(c) no filter, no transformer

Figure 8: Single phase model of a converter station, (a) omitting the filter, (b) omitting the transformer and (c) omitting both filter and transformer.

- Transformerless and filterless design
- Simplified model of converter: voltage source behind a complex impedance (or pure inductive impedance when omitting R_c).

The different converter parameters R_{tf}, X_{tf}, B_f, R_c and X_c are respectively defined as the named indices `RTF`, `XTF`, `BF`, `RCONV` and `XCONV` in the `convdc` input matrix (see chapter 3). Depending on the input data defined, MATACDC automatically selects one of the corresponding representations from Fig. 8.

4.1.2 Converter losses

Converter losses can be taken into account using a generalized loss formula quadratically depending on the converter current I_c [8]:

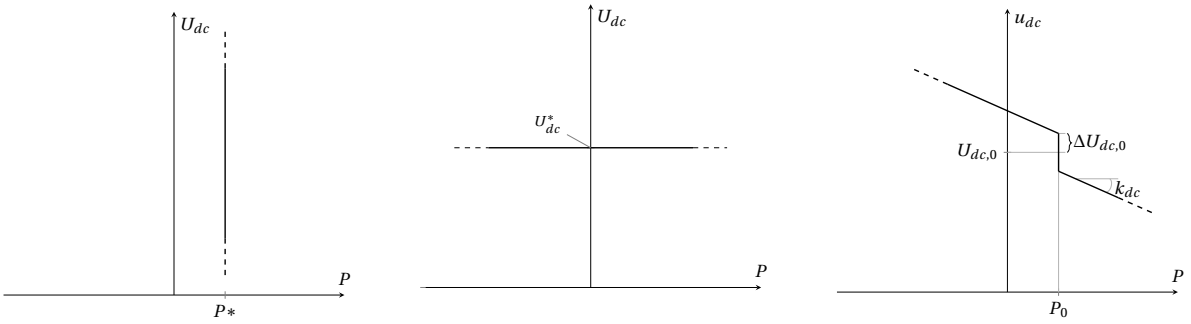
$$P_{loss} = a + b \cdot I_c + c \cdot I_c^2. \tag{10}$$

The loss coefficients are defined as the named indices `LOSSA`, `LOSSB`, and `LOSSCR` and `LOSSI` in the `convdc` matrix. A distinction can be made for the quadratic term in rectifier and inverter mode.

4.1.3 Converter control

Each converter can exhibit a number of different control functions. Due to the decoupled current control, the active and reactive power can be controlled independently. The different actual control options correspond to the steady-state representations used in MATACDC. The software includes 3 different converter representations with respect to the active power and 2 different converter representations with respect to the reactive power.

Fig. 9 shows the $P-U$ steady-state characteristics for the three control options.



(a) P constant (b) U_{dc} constant (c) $P-U_{dc}$ droop

Figure 9: Converter steady-state operation characteristics.

The active power operating mode can be changed using `convdc` index `CONVTYPE_DC` and the named constants from Tab. 3. The corresponding steady state representations are defined in MATACDC as follows:

1. *P constant*: The converter has a constant active power injection P_s into the AC grid.

```
convdc(..., CONVTYPE_DC) = DCNOSLACK;  %% set to active power control
convdc(..., PCONV) = ...;              %% active power set-point  P*_s
```

2. *U_{dc} constant*: The converter controls the DC bus voltage U_{dc} at the converter terminal to a constant value, irrespective of the active power signal.

```
convdc(..., CONVTYPE_DC) = DCSLACK;  %% set to dc voltage control
busdc( ..., VDC) = ...;              %% dc voltage set-point  U*_dc
```

3. *$P-U_{dc}$ droop*: The active power injection into the DC grid P_{dc} depends on the actual value of the DC bus voltage U_{dc}. The droop control has set-points for voltage and power, respectively $U_{dc,0}$ and $P_{dc,0}$, has a variable voltage droop k_{dc} and can include a symmetric voltage deadband $\Delta U_{dc,0}$ as depicted in Fig. 9c.

```
convdc(..., CONVTYPE_DC) = DCDROOP;  %% set to dc voltage droop control
convdc(..., DROOP  ) = ...;   %% voltage droop  k_dc
convdc(..., PDCSET ) = ...;   %% voltage droop power set-point  P_dc,0
convdc(..., VDCSET ) = ...;   %% voltage droop voltage set-point  U_dc,0
convdc(..., DVDCSET) = ...;   %% voltage droop deadband  ΔU_dc,0
```

The reactive power operating mode can be changed using `convdc` index `CONVTYPE_AC` and the named constants from Tab. 3. The converter can be represented in MATACDC as:

1. *Q constant*: The converter has a constant reactive power injection Q_s into the AC grid.

   ```
   convdc(...,CONVTYPE_AC) = PQC;   %% set to reactive power control
   convdc(..., QCONV) = ...;        %% reactive power set point Q*_s
   ```

2. *U constant*: The converter adapts the reactive power injection to obtain a constant AC bus voltage magnitude U_s.

   ```
   convdc(...,CONVTYPE_AC)=PVC;   %% set to ac voltage control
   convdc(..., VCONV) = ...;       %% AC voltage set point  U*_s
   ```

4.1.3.1 Case study: One DC voltage controller

In existing point-to-point VSC HVDC systems, it is common practice to have only one converter at the time controlling the DC voltage. The other converter is defined as constant power controller. In the `case5_stagg_HVDCptp` file, a two-terminal DC system is defined with

```
%%      busdc_i  type_dc type_ac    P_g   Q_g   Vtar    ...
convdc = [  1       1       1       -60    -40    1     ...;
            2       2       2         0      0    1     ...; ];
```

In this case, the first converter is under active power control and will be represented in the power flow algorithm as a constant P converter. The second converter, on the other hand, is the DC slack converter (constant U_{dc} control) and changes the active power to keep its DC bus voltage constant.

Similarly, the reactive power for converter 2 is unknown prior to the power flow, as this converter is set to constant AC voltage control.

Solving the power flow leads to the results for the active and reactive power:

```
>> define_constantsdc;
>> [mpc, mpcdc] = runacdcpf('case5_stagg','case5_stagg_HVDCptp');
>> mpcdc.convdc(:,[PCONV QCONV]) =
        -60.0000  -40.0000
         56.4959    0.0041
```

Please remark that the reactive power control can be set independently at each converter, whereas the active power settings are not independent.

4.1.3.2 Case study: Voltage droop control

As discussed in chapter 6, the effect of the converter control on the power flows in the system can be easily analyzed. This can be observed when using the `casedc` files `case5_stagg_MTDCslack` and `case5_stagg_MTDCdroop` for a power flow calculation.

The results for the constant power and constant DC voltage control are obtained using:

```
>> runacdcpf('case5_stagg','case5_stagg_MTDCslack');
```

The results for the DC voltage droop control are obtained using:

```
>> runacdcpf('case5_stagg','case5_stagg_MTDCdroop');
```

Both files lead to similar results, since the droop set-points in `case5_stagg_MTDCdroop` have been defined based on the power flow results from `case5_stagg_MTDCslack`. When, however, a converter outage is simulated, it can be observed that the power injections at all converters are reduced as a result of the droop control action.

The results for the constant power and constant DC voltage control are obtained using:

```
>> define_constantsdc;
>> mpcdc = loadcasedc('case5_stagg_MTDCslack');
>> mpcdc.convdc(3,CONVSTATUS) = 0;   %% outage of converter 3
>> runacdcpf('case5_stagg', mpcdc);
```

The results for the DC voltage droop control are obtained using:

```
>> define_constantsdc;
>> mpcdc = loadcasedc('case5_stagg_MTDCdroop');
>> mpcdc.convdc(3,CONVSTATUS) = 0;   %% outage of converter 3
>> runacdcpf('case5_stagg', mpcdc);
```

4.1.4 Converter limits

MATACDC allows to include converter current and voltage limits. Fig. 10 depicts the PQ capability chart of the converter and the limits included in MATACDC. The full lines include the effect of the

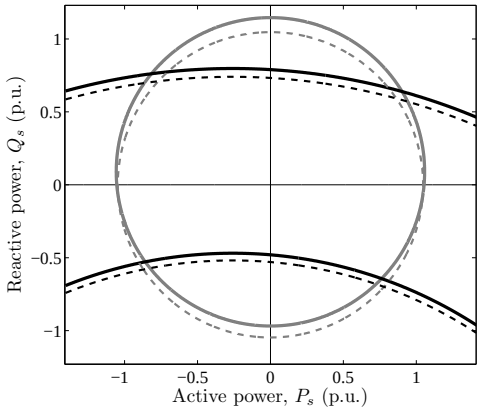


Figure 10: PQ-capability chart – Converter station current and voltage limits

filters, the dotted lines neglect the filters. The gray circle forms the converter current limit. The black arcs depict the upper and lower converter voltage limits. When a converter set-point is located outside of the converter limits, MATACDC prioritizes active over reactive power when enforcing the limits.

More details on the modelling of the converter limits can be found in [6].

4.1.4.1 Enabling/disabling converter limits

The converter upper and lower voltage limits and the converter current limit are defined in the `convdc` matrix in respectively the `VCMAX`, `VCMIN` and `ICMAX` index and can be easily changed:

```
>> define_constantsdc;
>> mpcdc = loadcasedc('case5_stagg_MTDCslack');
>> mpcdc.convdc(3,ICMAX) = 1.1; %% change current limit in converter 3
```

By default, the converter limits are not enforced. The limits can be enabled by changing index `LIMAC` of the default option vector defined in `macdcoption`:

```
>> opt = macdcoption;
>> opt(10) = 1; % LIMAC, enforce converter current and voltage limits
>> runacdcpf('case5_stagg','case5_stagg_MTDCslack',opt);
```

As a result, the reactive power at converter 1 is lowered.

When the limits are enabled by the user, the following situations may occur

- When an **active power controlling converter** has an active power set-point 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 at the maximum converter active power limit.
- Similarly, when a **reactive power controlling converter** has a reactive power set-point 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, on the other hand, this is the case, the active power is prioritized 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 prioritized when the active power limit is violated.

DC slack converters (converters with a constant U_{dc} control) are excluded from the analysis, but are checked at the end of the power flow calculation. It is up to the user to redefine the converter set-points when it is observed that a limit is hit in those converters.

As an example, it is possible to change the active power set-point of converter 1, so that at the same time, the reactive and active power limit are violated:

```
>> define_constantsdc;
>> opt = macdcoption;
>>opt(10) = 1; % enforce converter current and voltage limits
>> mpcdc = loadcasedc('case5_stagg_MTDCslack');
>> mpcdc.convdc(1,PCONV) = -130;
>> runacdcpf('case5_stagg',mpcdc,opt);
```

4.1.4.2 Visualization

It is also possible to visualize the converter limit violations, by enabling index `CONVPLOTOPT`. The three possible options are

- 0 - do not plot converter limit violations
- 1 - plot only converter limit violations
- 2 - plot converter limit violations and end situation

A graph similar to the one in Fig. 10 is plotted, showing the operational point with and without limit inclusion.

```
>> opt = macdcoption;
>> opt(10) = 1; % LIMAC, enforce converter current and voltage limits
>> opt(14) = 1; % CONVPLOTOPT, plot converter limit violations
>> runacdcpf('case5_stagg','case5_stagg_MTDCslack',opt);
```

4.2 DC System modelling

Under steady-state conditions, the DC networks can be represented by a resistive network with current injections and DC voltages at the different nodes, as depicted in Fig. 11 for the example from Fig. 3.

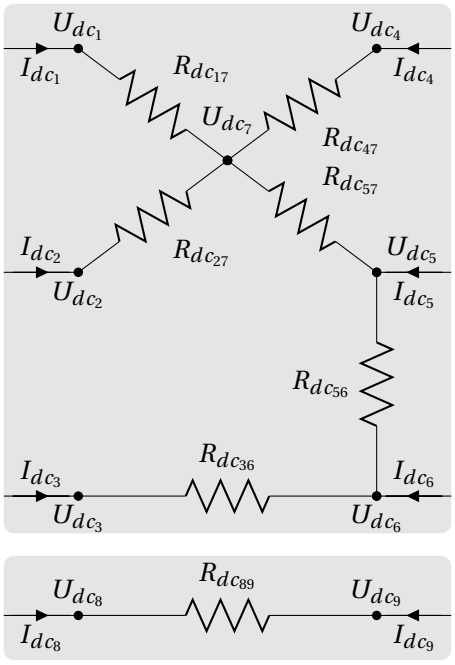


Figure 11: DC network modeling: example

The current injected at a DC node i can be written as the current flowing to the other $n-1$ nodes in the network:

$$I_{dc_i} = \sum_{\substack{j=1 \\ j \neq i}}^{n} Y_{dc_{ij}} \cdot (U_{dc_i} - U_{dc_j}), \tag{11}$$

with $Y_{dc_{ij}}$ equal to $1/R_{dc_{ij}}$.

Combining all currents injected in a DC grid with n buses results in

$$\boldsymbol{I_{dc}} = \mathsf{Y}_{\mathsf{dc}} \boldsymbol{U_{dc}}, \tag{12}$$

with the DC current vector $\boldsymbol{I_{dc}}$ given by

$$\boldsymbol{I_{dc}} = [\underbrace{I_{dc_1}, I_{dc_2} \dots I_{dc_k}}_{\text{working converters}}, \underbrace{0 \dots 0}_{\text{outage}}]^T, \tag{13}$$

with $n-k$ zero elements due to converter outages and DC buses without a connection to the AC system. The DC voltage vector is given by $\boldsymbol{U_{dc}} = [U_{dc_1}, U_{dc_2} \dots U_{dc_n}]^T$ and Y_{dc} is the DC bus matrix.

The DC grid power flow equations can be written as

$$P_{dc_i} = p\, U_{dc_i} \sum_{\substack{j=1 \\ j \neq i}}^{n} Y_{dc_{ij}} \cdot (U_{dc_i} - U_{dc_j}). \tag{14}$$

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. This parameter of the system is defined as the variable `pol` in the `casedc` data files.

In case of a $U_{dc} - P_{dc}$ droop, the DC power injected by the voltage droop controlled buses, can be written as

$$P_{dc_i} = P_{dc,0_i} - \frac{1}{k_{dc_i}} (U_{dc_i} - U_{dc,0_i}). \tag{15}$$

5 Sequential AC/DC Power Flow

This chapter will briefly summarize the sequential AC/DC power flow method used by MATACDC. Fig. 12 summarizes converter AC and DC powers, voltages and losses and how they relate to the `busdc`, `convdc` and `branchdc` data matrices. The AC representation of the converter stems with one of the implementations depicted in Fig. 8.

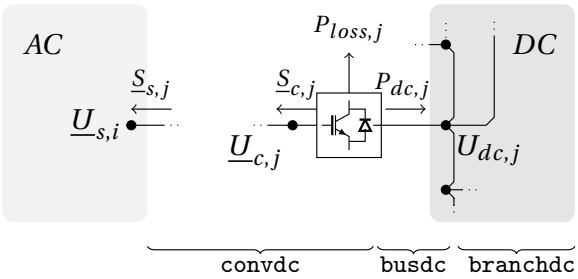


Figure 12: AC/DC System interconnection

The sequential AC/DC power flow algorithm sequentially solves the AC and DC system power flow, thereby respectively keeping all converter powers and voltages constant. Fig. 13 shows the flow chart of the AC/DC power flow algorithm. The remaining part of this chapter addresses how MATACDC internally implements the sequential power flow algorithm. More details on the mathematical modelling can be found in [6, 7].

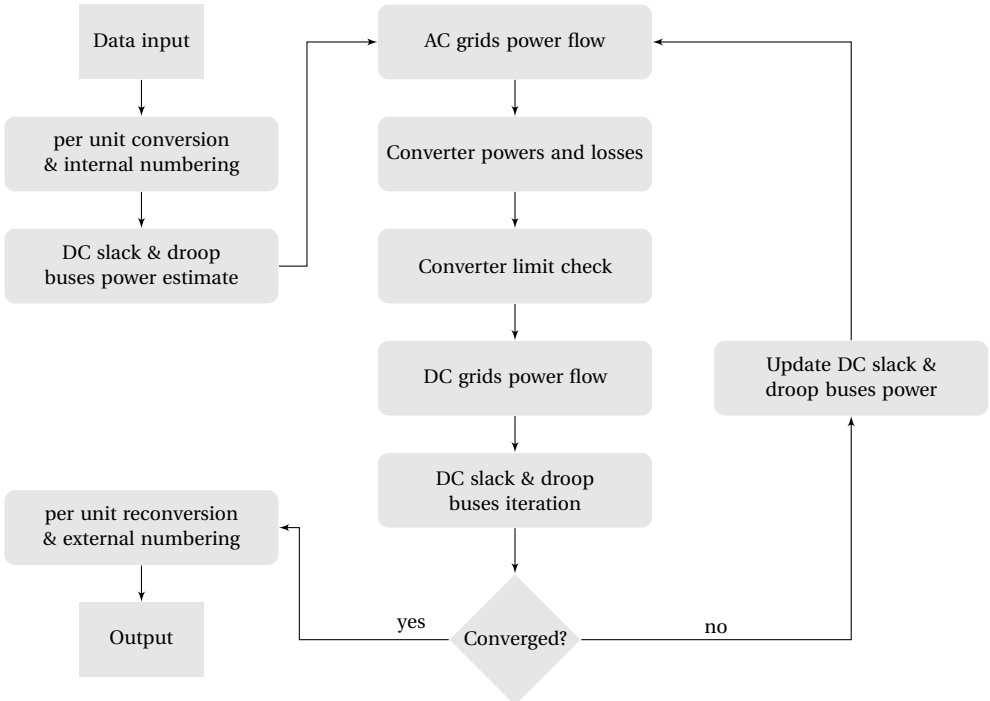


Figure 13: Flow chart of the sequential VSC AC/DC power flow algorithm

In the mathematical description in this chapter, 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 chapter limits itself to one DC grid, but can be easily generalized.

5.1 AC System 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:

1. **Non-synchronized zones:** In case multiple AC grids are defined (using the named index `ZONE` in the `bus` matrix), MATACDC extracts the AC system data for each zones and solves the power flow for each AC grid one by one.

2. **Infinite buses:** In case the user is only interested in the DC system power flow or in case an AC zone only has one node, the user can define infinite bus systems, by having `inf` for index `BUS_TYPE` in the `bus` matrix. When MATACDC runs the MATPOWER `runpf` routine these buses are taken out of the analysis.

5.1.1 Initialization

After having converted all converter data and the AC and DC grid data to the same per unit base, the AC and DC buses are renumbered as explained in 3.3. In the AC power flow, all DC system and converter data is kept constant. To start the iteration, the system needs a first approximation of the converter active power injections. Thereby, the DC system and converters are assumed to be lossless.

As a first estimate to initiate the overall iteration, the AC active power injections of the droop controlled converters are put equal to the negative of the DC power reference $P_{dc,0}$, thereby assuming that the DC voltage does not deviate from the reference value $U_{dc,0}$ and neglecting the converter losses P_{loss}. Without lack of generality, we assume [2] a n bus DC system with the first converter as the DC slack bus and the subsequent $m-1$ converters using a DC voltage droop control. The next $k-m$ buses are under constant active power control. The remaining $n-k$ buses do not have a connection to the AC grid or are facing a converter outage. The vector of AC power injections can thus be written as

$$\boldsymbol{P_s} = [\underbrace{P_{s_1}}_{\text{slack}}, \underbrace{P_{s_2} \dots P_{s_m}}_{\text{voltage droop}}, \underbrace{P_{s_{m+1}} \dots P_{s_k}}_{P-\text{control}}, \underbrace{0 \dots 0}_{\text{outage}}]^T. \tag{16}$$

The AC power injections of the $k-m$ power controlled buses are defined in the input data. The initialization can therefore be written as

$$P_{s_i}^{(0)} = P_{s_i} \quad \forall i\colon m+1 \le i \le n. \tag{17}$$

The active power injection estimate of the $m-1$ converters under voltage control is given by

$$P_{s_i}^{(0)} = -P_{dc,0_i} \quad \forall i\colon 2 \le i \le m. \tag{18}$$

[2] In MATACDC, the converters are not sorted based on their control, as done here for convenience.

The active power delivered by the DC slack bus, if present, is initiated as

$$P_{s_1}^{(0)} = -\sum_{i=2}^{m} P_{s_i}^{(0)} - \sum_{j=m+1}^{n} P_{s_j}, \tag{19}$$

with $P_{s_i}^{(0)}$ from (18) and P_{s_j} defined prior to the power flow as the active power injections of the constant power buses.

In case one DC grids contains several voltage controlling converters, the total deficit power is distributed amongst the different DC converters. In the general case of l DC voltage controlling converters, the initialization from (19) becomes

$$P_{s_h}^{(0)} = -\frac{1}{l}\left(\sum_{i=l+1}^{m} P_{s_i}^{(0)} + \sum_{j=m+1}^{n} P_{s_j}\right) \quad \forall h\colon 1 \le h \le l. \tag{20}$$

Please note that having a multiple number of DC slack buses in one DC grid (hence multiple DC voltage controlled nodes) can give rise to divergence in the DC power flow and has to be explicitly defined by the user in the option vector.

For subsequent iterations, the solution from the previous iteration is used for $P_{s_1}^{(k)}$.

5.1.2 Converter representation

The power flow equations for bus i in the AC grid can be written as

$$P_i(\boldsymbol{U},\boldsymbol{\delta}) = U_i \sum_{j=1}^{p} U_j \left[G_{ij}\cos(\delta_i - \delta_j) + B_{ij}\sin(\delta_i - \delta_j)\right], \tag{21}$$

$$Q_i(\boldsymbol{U},\boldsymbol{\delta}) = U_i \sum_{j=1}^{p} U_j \left[G_{ij}\sin(\delta_i - \delta_j) - B_{ij}\cos(\delta_i - \delta_j)\right]. \tag{22}$$

In the AC power flow routine, for which MATACDC relies on the MATPOWER file `runpf`, all converters are represented as constant active power inputs to the AC system. Converters in V-control are represented as dummy AC generators and their AC buses are changed from PQ-nodes to PV-nodes. Several situations can occur for a converter under V-control :

- When no generator is present at the AC node under consideration, a dummy generator is added to the bus.
- When a generator is present, but the node was a PQ node, the node is changed to a PV node and only the additional reactive power delivered is considered to be the converter reactive power.
- When a generator is present, and the AC node was a PV node, the converter is set to constant reactive power control instead.

Fig. 14 shows the representation of the converters in the AC power flow.

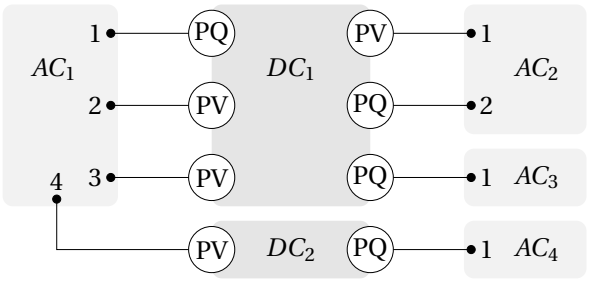


Figure 14: Representation of converters in the AC system

The converter power injections $\boldsymbol{P_s}$ and $\boldsymbol{Q_s}$ are included in the power mismatch vectors $\boldsymbol{\Delta P}^{(k)}$ and $\boldsymbol{\Delta Q}^{(k)}$ as negative loads. The power mismatch vectors can thus be rewritten as

$$\Delta P_i^{(j)} = P_i^{gen} - (P_i^{dem} - P_{s_i}) - P_i(\boldsymbol{U}^{(j)}, \boldsymbol{\delta}^{(j)}), \tag{23}$$

$$\Delta Q_i^{(j)} = Q_i^{gen} - (Q_i^{dem} - Q_{s_i}) - Q_i(\boldsymbol{U}^{(j)}, \boldsymbol{\delta}^{(j)}). \tag{24}$$

With respect to the active power, all converters are treated similar in the sense that also the DC slack and droop buses are introduced in the AC power flow algorithm as either PQ or PV buses. However, when a converter is under DC voltage control, either a slack bus or droop-based control, the active power injection in the AC grid is not known beforehand, since it depends on the active power at the DC side to control the DC voltage.

After calculating the AC power flow, all converter powers and losses are calculated to obtain the DC grid's injected powers $\boldsymbol{P_{dc}}$ for the k DC buses to which converters are connected, disregarding the ones facing outages.

$$P_{dc_i} = -P_{c_i} - P_{loss_i}, \ \forall i < k, \tag{25}$$

with P_c the active part of the complex power injected at the converter side, shown in Fig. 6. The converter powers are determined according to one of the implementations from Fig. 8. More details can be found in [6].

5.2 DC grid power flow

With the DC power injections calculated as a result of the AC power flow, MATACDC uses a Newton-based iteration, based on (14) – (15) to calculate the DC grid's power flow. For the converters under constant power control, $m+1$ to k, the DC power injection P_{dc} as defined by (14), is known as a result of the AC power flow. For the converters under distributed voltage control, 2 to m, the DC power injection set-points $P_{dc,0}$ are all known. A modified active power vector $\boldsymbol{P'_{dc}}$ is introduced to group these variables, hence

$$\boldsymbol{P'_{dc}} = [\underbrace{P_{dc_1}}_{\text{slack}}, \underbrace{P_{dc,0_2} \dots P_{dc,0_m}}_{\text{voltage droop}}, \underbrace{P_{dc_{m+1}} \dots P_{dc_k}}_{P-\text{control}}, \underbrace{0 \dots 0}_{\text{outage}}]^T. \tag{26}$$

Using this modified power vector, the DC bus voltages are calculated with a NR method:

$$\left(\boldsymbol{U_{dc}} \frac{\partial \boldsymbol{P'_{dc}}}{\partial \boldsymbol{U_{dc}}}\right)^{(j)} \cdot \frac{\boldsymbol{\Delta U_{dc}}^{(j)}}{\boldsymbol{U_{dc}}} = \boldsymbol{\Delta P'_{dc}}^{(j)}. \tag{27}$$

The equations and terms corresponding to the slack bus are removed since its voltage is known prior to the DC network power flow.

In this system of equations, the modified power mismatch vector $\boldsymbol{\Delta P'_{dc}}^{(j)}$ is given by

$$\Delta P'^{(j)}_{dc_i} = \begin{cases} P_{dc,0_i} - P_{dc,0_i}(\boldsymbol{U_{dc}}^{(j)}) & \forall i\colon 2 \le i \le m \\ P^{(k)}_{dc_i} - P_{dc_i}(\boldsymbol{U_{dc}}^{(j)}) & \forall i\colon m < i \le k\,, \\ -P_{dc_i}(\boldsymbol{U_{dc}}^{(j)}) & \forall i\colon k < i \le n \end{cases} \tag{28}$$

with $P_{dc,0_i}(\boldsymbol{U_{dc}}^{(j)})$ given by

$$P^{(j)}_{dc,0_i} = P_{dc_i}(\boldsymbol{U_{dc}}^{(j)}) + \frac{1}{k_i}(U^{(j)}_{dc_i} - U_{dc,0_i}), \tag{29}$$

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

More details on the DC grid power flow can be found in [6, 7].

5.3 DC slack and droop buses iteration

The AC bus active power injection P_s of the DC slack and droop bus is calculated from the DC powers P_{dc} and by accounting for the converter losses. As the converter losses from (10) depend on the yet unknown converter current, an additional iteration is needed to calculate the active power injection P_s. During this iteration, the grid side voltage $\underline{U}_s$ and reactive power injection Q_s are kept constant. Omitting the subscript to simplify notations, the value of P_c is iteratively updated according to

$$P^{(i)}_c = -P^{(k)}_{dc} - P^{(i)}_{loss}, \tag{30}$$

with the superscripts (i) and (k) respectively referring to the DC slack bus iteration and the outer AC/DC power flow iteration. The converter losses P_{loss} are calculated from (10). The previous AC network power flow results are used to provide an initial estimate for the converter losses $P^{(0)}_{loss}$.

A Newton iteration based on $\underline{U}_c$ and $\underline{U}_f$ as variables is internally used to update the converter state in order to obtain a new value for $P^{(i)}_{loss}$. For each DC slack or droop node, the iteration uses the values of Q_s, which is assumed to be constant during the iteration and the value of P_c, which is updated each iteration according to (30). Using (2) and (3), Q_s and P_c can be written as and in terms of $\underline{U}_c$ and $\underline{U}_f$. The power conservation at the filter bus leads to two additional equations in terms of $\underline{U}_c$ and $\underline{U}_f$, leading to four equations with four unknown variables

$$Q_s(\underline{U}_f, \underline{U}_c) \tag{31}$$
$$P_c(\underline{U}_f, \underline{U}_c) \tag{32}$$
$$F_1(\underline{U}_s, \underline{U}_c, \underline{U}_f) = P_{cf} - P_{sf}, \tag{33}$$
$$F_2(\underline{U}_s, \underline{U}_c, \underline{U}_f) = Q_{cf} - Q_{sf} - Q_f, \tag{34}$$

which leads to a four equations with four unknown variables. P_{cf}, P_{sf}, Q_{cf}, Q_{sf} and Q_f are defined in terms of $\underline{U}_c$ and $\underline{U}_f$ in (3) and (2).

The slack and droop bus iteration is depicted in Fig. 15.

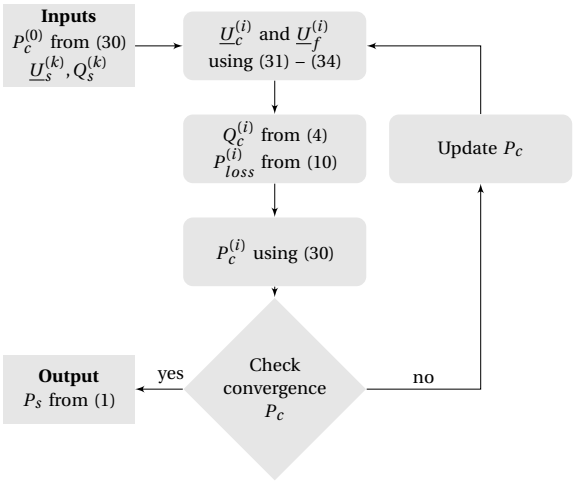


Figure 15: DC Slack & droop bus iteration flow chart

In the absence of a transformer, the active power P_c and reactive power Q_s are rewritten in terms of $\underline{U}_s$ and $\underline{U}_c$ as

$$P_c = U_c^2 G_c - U_s U_c [G_c \cos(\delta_s - \delta_c) - B_c \sin(\delta_s - \delta_c)] \tag{35}$$

$$Q_s = U_s^2 B_{cf} + U_s U_c [G_c \sin(\delta_s - \delta_c) - B_c \cos(\delta_s - \delta_c)], \tag{36}$$

with $B_c + B_f$ abbreviated as B_{cf}. Thereby, the number of unknowns is reduced and the problem is simplified to a system with two equations unknown variables and two unknown variables. More details can be found in [6].

6 Examples

MATACDC has a number of predefined data files that can be used to obtain AC/DC power flow results.

- In the `PowerflowAC` map:

  `case3_inf.m`

  `case5_stagg.m`

  `case24_ieee_rts1996_3zones.m`

  `case24_ieee_rts1996_3zones_inf.m`

- In the `PowerflowDC` map:

  `case5_stagg_HVDCptp.m`

  `case5_stagg_MTDCslack.m`

  `case5_stagg_MTDCdroop.m`

  `case24_ieee_rts1996_MTDC.m`

All power flows use the `runacdcpf` command, and have the AC and DC power flow data as input arguments, either as structs or as a string referring to the corresponding .m-file.

```
>> [resultsac, resultsdc] = runacdcpf(caseac, casedc);
```

Optionally, the `runacdcpf` command also returns a convergence flag (`converged`) and a the elapsed time (`timecalc`). It is also possible to define non-default options, both for the AC/DC power flow algorithm (`macdcopt`), as well as for the standard MATPOWER routines (`mpopt`):

```
>> [resultsac, resultsdc, converged, ...
        timecalc] = runacdcpf(caseac, casedc, macdcopt, mpopt);
```

Please consult `help runacdcpf` for more options and information on the input and output data and format.

6.1 Stagg 5-node test system

The Stagg test system, with the AC system based on [9] can be combined with the DC system based on [7]. The AC power flow solution can be obtained using the MATPOWER `runpf` command:

```
>> runpf('case5_stagg');
```

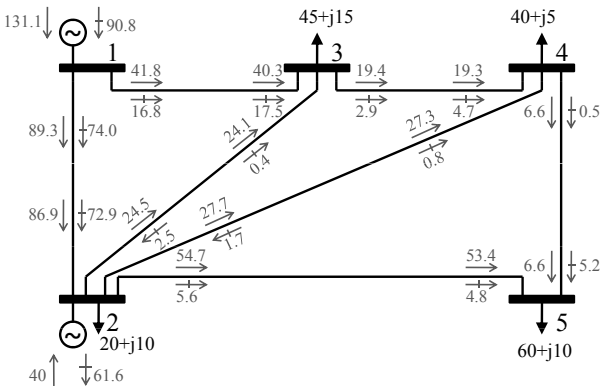


Figure 16: Stagg 5-node test system: AC power flow results (without DC system)

6.1.1 Constant power and DC voltage

The results for the constant power (converter 1 & 3) and constant DC voltage control (converter 2) can be obtained using the `case5_stagg_MTDCslack` file:

```
>> runacdcpf('case5_stagg','case5_stagg_MTDCslack');
```

The output files can be stored in two separate data structs:

```
>> [mpc,mpcdc] = runacdcpf('case5_stagg','case5_stagg_MTDCslack');
```

Alternatively, it is also possible to define data matrices as output files:

```
>> [baseMVA, bus gen ,branch, busdc, convdc, branchdc] = ...
            runacdcpf('case5_stagg','case5_stagg_MTDCslack');
>> [baseMVA, bus gen ,branch, busdc, convdc, branchdc, converged, timecalc] = ...
            runacdcpf('case5_stagg','case5_stagg_MTDCslack');
```

For more information on the input and output data and the options, please use `help runacdcpf`.

All previous commands results in a printed output which summarizes the AC and DC power flow results. The AC power flow results relies on the MATPOWER `printpf` command.

```
Sequential solution method converged in 3 iterations

Converged in 0.26 seconds
================================================================================
|     System Summary                                                          |
================================================================================

How many?                How much?              P (MW)            Q (MVAr)
---------------------    -------------------  -------------  -----------------
Buses               5    Total Gen Capacity     550.0        -800.0 to 800.0
```

Generators	2	On-line Capacity	550.0	-800.0 to 800.0
Committed Gens	2	Generation (actual)	173.6	51.5
Loads	4	Load	165.0	40.0
Fixed	4	Fixed	165.0	40.0
Dispatchable	0	Dispatchable	-0.0 of -0.0	-0.0
Shunts	0	Shunt (inj)	-0.0	0.0
Branches	7	Losses (I^2 * Z)	4.39	13.18
Transformers	0	Branch Charging (inj)	-	29.6
Inter-ties	0	Total Inter-tie Flow	0.0	0.0
Areas	1			

	Minimum	Maximum
Voltage Magnitude	0.991 p.u. @ bus 5	1.060 p.u. @ bus 1
Voltage Angle	-4.26 deg @ bus 4	0.00 deg @ bus 1
P Losses (I^2*R)	-	2.72 MW @ line 1-2
Q Losses (I^2*X)	-	8.15 MVAr @ line 1-2

Bus Data						
Bus #	Voltage Mag(pu)	Voltage Ang(deg)	Generation P (MW)	Generation Q (MVAr)	Load P (MW)	Load Q (MVAr)
1	1.060	0.000*	133.64	84.32	-	-
2	1.000	-2.383	40.00	-32.84	20.00	10.00
3	1.000	-3.895	-	-	45.00	15.00
4	0.996	-4.262	-	-	40.00	5.00
5	0.991	-4.149	-	-	60.00	10.00
		Total:	173.64	51.48	165.00	40.00

Branch Data								
Brnch #	From Bus	To Bus	From Bus Injection P (MW)	From Bus Injection Q (MVAr)	To Bus Injection P (MW)	To Bus Injection Q (MVAr)	Loss (I^2 * Z) P (MW)	Loss (I^2 * Z) Q (MVAr)
1	1	2	98.38	71.37	-95.66	-69.59	2.717	8.15
2	1	3	35.26	12.96	-34.20	-15.08	1.062	3.19
3	2	3	13.25	-6.22	-13.14	2.57	0.116	0.35
4	2	4	17.08	-5.18	-16.89	1.74	0.181	0.54
5	2	5	25.33	-1.85	-25.07	-0.35	0.257	0.77
6	3	4	23.09	4.64	-23.04	-6.47	0.057	0.17
7	4	5	-0.07	-0.27	0.07	-4.65	0.004	0.01
					Total:		4.393	13.18

```
================================================================================
|     DC bus data                                                              |
================================================================================
 Bus   Bus   Voltage    Power
 DC #  AC #  Mag(pu)    P (MW)
-----  ----  ---------  --------
   1     2     1.008    -58.627
   2     3     1.000     21.901
   3     5     0.998     36.186

================================================================================
|     VSC Converter Data                                                       |
================================================================================
 Bus     Bus injection          Converter Voltage                 Total loss
 DC#   P (MW)   Q (MVAr)        Mag(pu) Ang(deg)                    P (MW)
-----  -------  --------        ------- --------                  -----------
   1   -60.00   -40.00           0.890  -13.017                       1.37
   2    20.76     7.14           1.007   -0.655                       1.14
   3    35.00     5.00           0.995    1.442                       1.19
                                                                      ------
                                                            Total:     3.70

 Bus  Converter power   Filter   Transfo loss     Reactor loss    Converter loss
 DC#  P (MW) Q (MVAr)  Q (MVAr)  P (MW) Q (MVAr)  P (MW) Q (MVAr)    P (MW)
----- ------- -------  --------  ------ --------  ------ -------- --------------
   1   -59.92  -32.63   -8.12     0.08    5.83     0.01    9.66        1.29
   2    20.76   -0.65   -9.02     0.01    0.54     0.00    0.70        1.14
   3    35.02   -0.37   -8.83     0.02    1.43     0.00    2.03        1.17
                                 ------ --------  ------ -------- --------------
                      Total:      0.10    7.80     0.01   12.39        3.59

 Bus  Grid power       Traf Filt.Power  Filter    Conv Filt. Pwr   Converter Power
 DC#  P (MW) Q (MVAr)  P (MW) Q (MVAr)  Q (MVAr)  Q (MVAr)         P (MW) Q (MVAr)
----- ------ --------  ------ --------  --------  --------------   ------ --------
   1   -60.00  -40.00  -59.92   -34.17   -8.12     -42.29          -59.92  -32.63
   2    20.76    7.14   20.76     7.68   -9.02      -1.35           20.76   -0.65
   3    35.00    5.00   35.02     6.43   -8.83      -2.40           35.02   -0.37
================================================================================
|     DC branch data                                                           |
================================================================================
Brnch   From   To     From Bus    To Bus      Loss
  #     Bus    Bus    P (MW)      P (MW)      P (MW)
-----  -----  -----  --------    --------    --------
   1      1      2     30.66      -30.42        0.24
   2      2      3      8.52       -8.50        0.02
   3      1      3     27.96      -27.68        0.28
```

```
                                --------
                    Total:         0.54
```

Similarly, the results after an outage of the third converter (DC bus 3 or AC bus 5) are obtained using:

```
>> define_constantsdc;
>> mpcdc = loadcasedc('case5_stagg_MTDCslack');
>> mpcdc.convdc(3,CONVSTATUS) = 0   %% outage of converter 3
>> runacdcpf('case5_stagg', mpcdc);
```

6.1.2 Voltage droop control

The results for the DC voltage droop control are obtained using the file `case5_stagg_MTDCdroop`:

```
>> runacdcpf('case5_stagg','case5_stagg_MTDCdroop');
```

Similarly, the power flow results after an outage of converter 3 are obtained with the following command:

```
>> define_constantsdc;
>> mpcdc = loadcasedc('case5_stagg_MTDCdroop');
>> mpcdc.convdc(3,CONVSTATUS) = 0;   %% outage of converter 3
>> runacdcpf('case5_stagg', mpcdc);
```

Figs. 17–18 shows the results of the AC/DC power flow before and after the outage, both for a constant power & voltage control, and a voltage droop control.

The droop settings can be easily changed, by having:

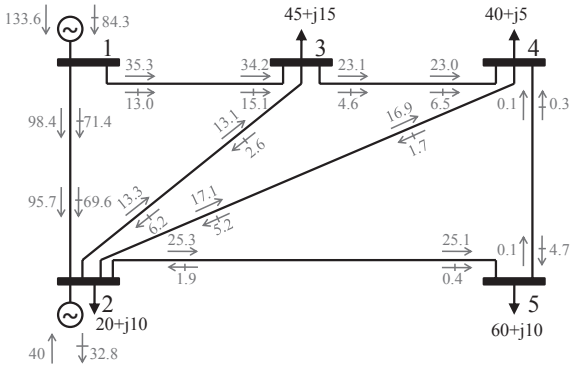
```
>> define_constantsdc;
>> mpcdc = loadcasedc('case5_stagg_MTDCdroop');
>> mpcdc.convdc(3,CONVSTATUS) = 0;   %% outage of converter 3
>> mpcdc.convdc(:,DROOP) = mpcdc.convdc(:,DROOP)/5;   %% changing all droop settings
>> runacdcpf('case5_stagg', mpcdc);
```

It can be observed that these lower droop settings result in much lower DC voltages after the outage of converter 3.

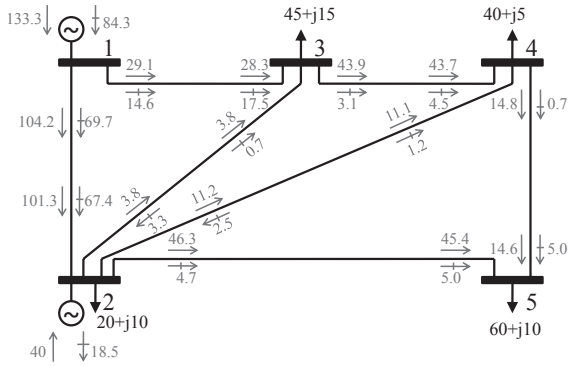
6.1.3 Infinite bus system

When not interested in the AC power flow results, MATACDC allows to have infinite buses defined in the AC system data. In file `case3_inf`, the Stagg 5-node test system has been replaced by 3 infinite nodes to which the converters are connected:

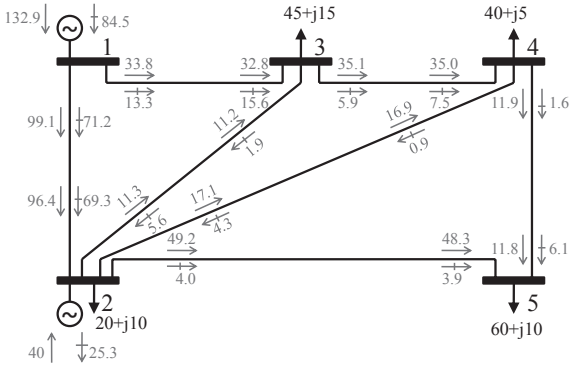
```
%% bus data
% bus_i type Pd Qd Gs Bs area Vm      Va baseKV zone Vmax Vmin
```



(a) Normal operation



(b) Outage: P-control



(c) Outage: DC voltage droop

Figure 17: Stagg 5-node test system: AC network power flow solution with a VSC MTDC system between buses 2, 3 and 5: (a) Normal operation, (b) Converter outage on bus 5 & P-control, (c) Converter outage on bus 5 & DC voltage droop on converters 2 and 3. **Legend**: → Active power (*MW*) and ⇢ Reactive power (*MVAr*)

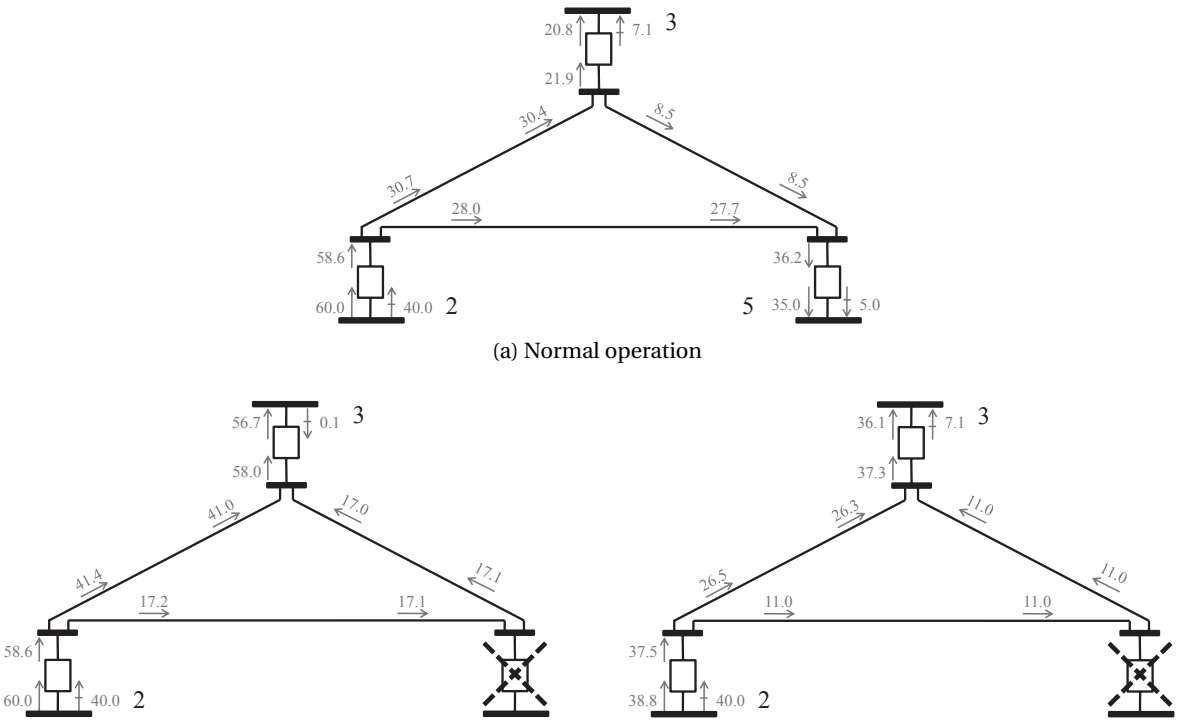
```
bus = [
2       inf       0     0 0    0    1    1.06     0 345     1       1.1     0.9;
3       inf       0     0 0    0    1    1        0 345     2       1.1     0.9;
5       inf       0     0 0    0    1    1        0 345     3       1.1     0.9;
];
```

The three nodes each belong to a different zone and have a voltage equal to the one defined in the `VM` index.

The power flow results can be obtained in a similar manner:

```
>> runacdcpf('case3_inf','case5_stagg_MTDCslack');
```

The power flow with the infinite buses leads to similar results. Please note that in case of infinite buses, the converters in V-control (in this example, bus 2) do not inject reactive power to support the voltage.



(a) Normal operation

(b) Outage: P-control

(c) Outage: DC voltage droop

Figure 18: DC network power flow solution with a VSC MTDC system between buses 2, 3 and 5 of Stagg 5-node test system: (a) Normal operation, (b) Converter outage on bus 5 & P-control, (c) Converter outage on bus 5 & DC voltage droop on converters 2 and 3. **Legend**: → Active power (*MW*)

6.2 RTS1996 test system

MATACDC also includes a more elaborated test system, based on the IEEE Two Area Reliability Test System-96. The two areas have been turned into two non- synchronized AC zones, interconnected by two DC systems, namely a 4-terminal meshed DC grid and a 3-terminal network. The 3-terminal network is connected to a third AC zone, which can represent e.g. an offshore wind farm.

```
>> runacdcpf('case24_ieee_rts1996_3zones','case24_ieee_rts1996_MTDC');
```

Similarly to the previous example, the third AC zone can be replaced by an infinite bus, which has been done in the data file `case24_ieee_rts1996_3zones_inf`.

```
>> runacdcpf('case24_ieee_rts1996_3zones_inf','case24_ieee_rts1996_MTDC');
```

Bibliography

[1] 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.

[2] 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.

[3] MatDyn website. [Online]. Available: http://www.esat.kuleuven.be/electa/teaching/matdyn/

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

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

[6] 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.

[7] 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.

[8] G. Daelemans, "VSC HVDC in meshed networks," Master's thesis, Katholieke Universiteit Leuven, Leuven, 2008.

[9] G. W. Stagg and A. H. El-Abaid, *Computer Methods in Power System Analysis*. Kogakusha, Japan: McGraw-Hill, 1968.