Réseau de transport d'électricité

LILLE

Initialisation and Load Flow with EMTP RV for Multiterminal DC grids

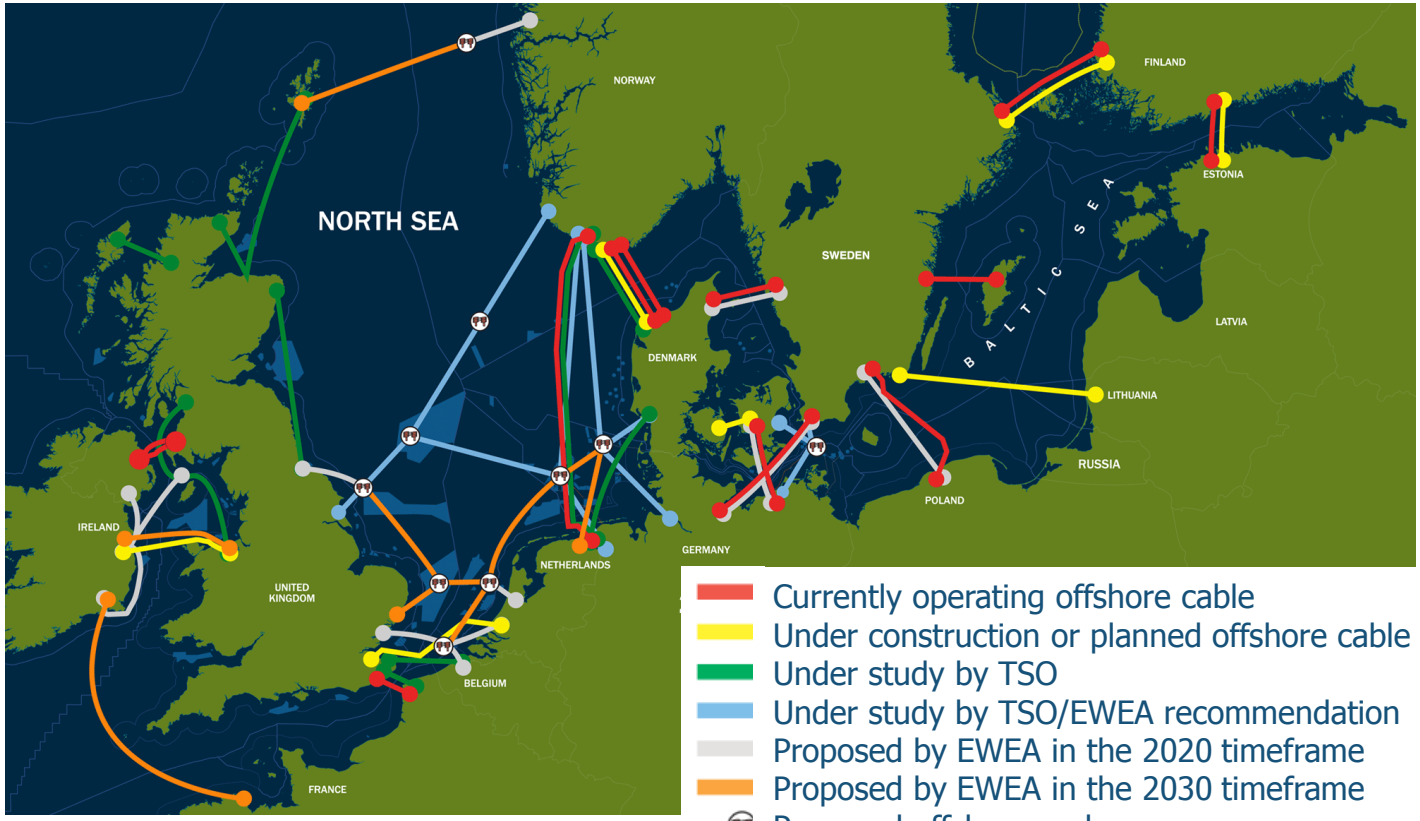
P. RAULT (PhD Student)
Supervisors:
F. Colas (L2EP)
X. Guillaud (L2EP)
S. Nguefeu (RTE)

Outline

1) Context
2) Find DC steady state solution
3) Initialize the DC network
4) Converter initialization
5) Initialize the overall system
6) Conclusion & Improvement

Context

EWEA's 20 year offshore network development plan



Source: EWEA 2009

The TWENTIES wind energy project

Secure large-scale integration of wind power into the European electricity grid

- Demonstration project
- Lunched by EU
- 62 M€ (32M€ Directly provided by EU)
- 26 Electrical companies & Research institutions
- 10 Member states are represented
- Coordinated by Red Eléctrica de España

➢ RTE task: Improving safety and security for offshore wind generation

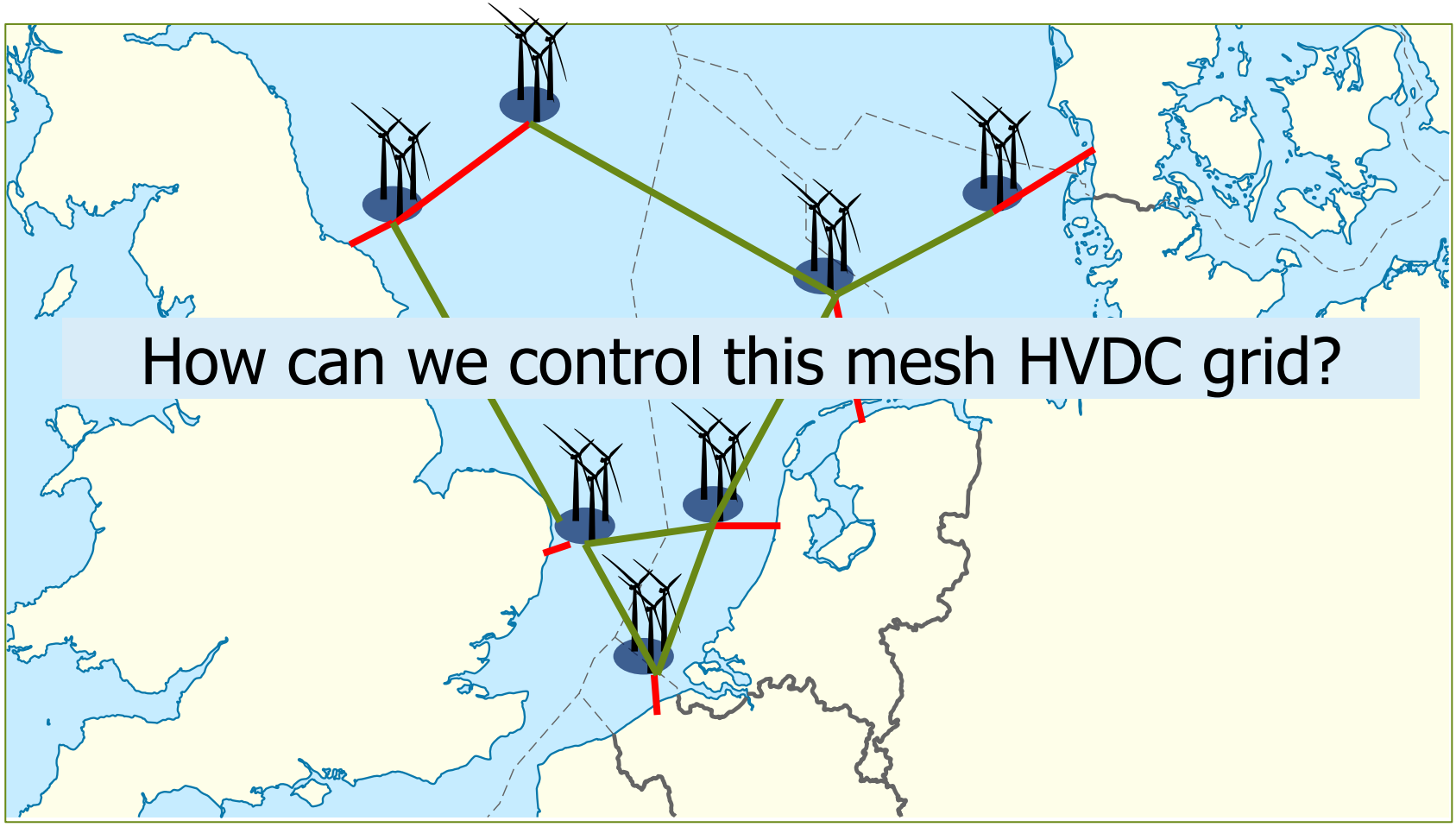
- Control & protection to roll out HVDC grid

➢ Further information

- http://www.twenties-project.eu

Context: My work

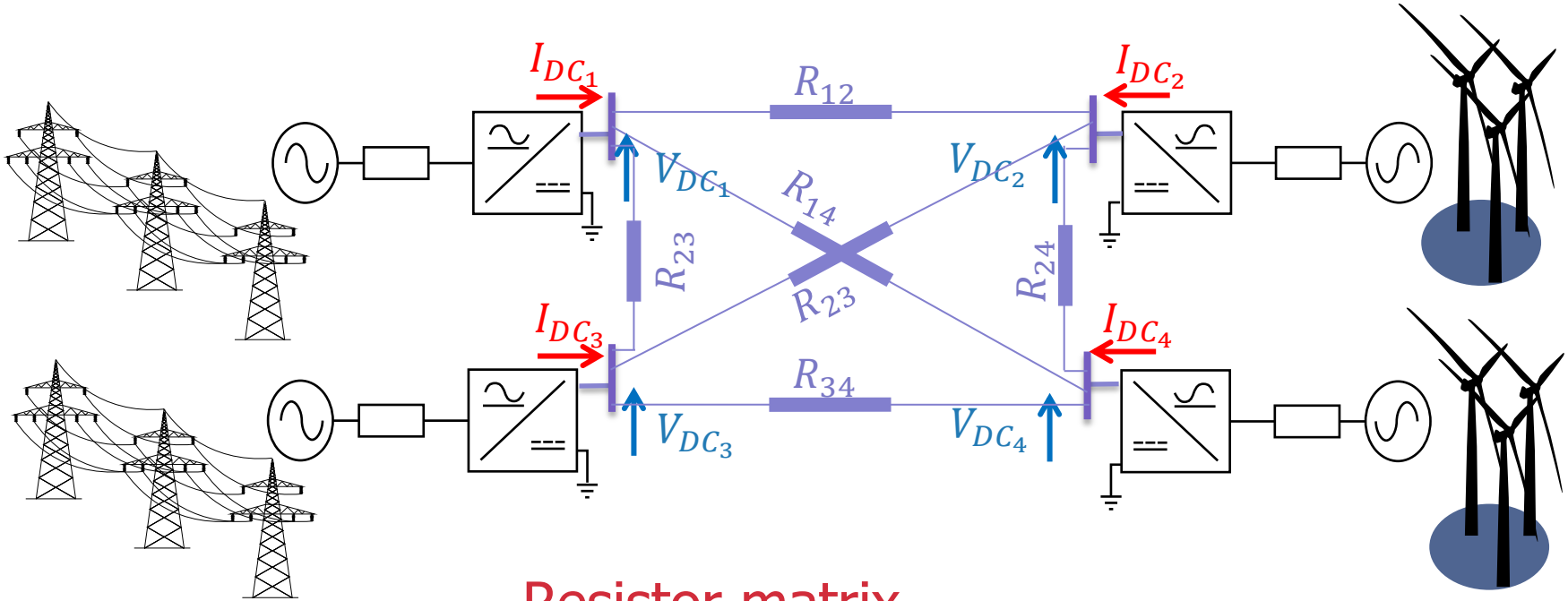
Multiterminal HVDC network



Outline

1) Context
2) Find DC steady state solution
3) Initialize the DC network
4) Converter initialization
5) Initialize the overall system
6) Conclusion & Improvements

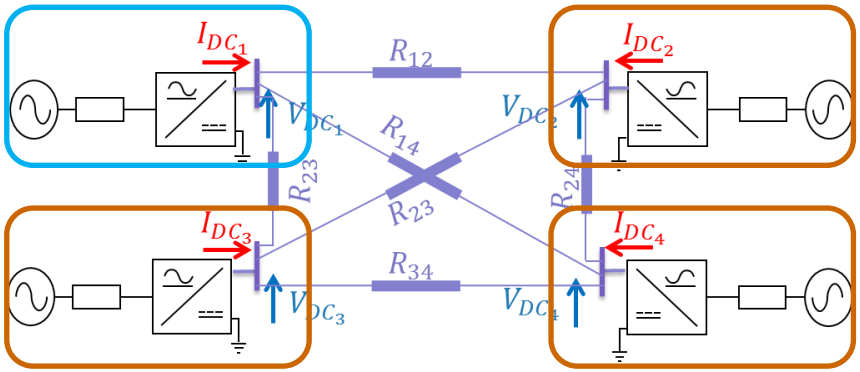
Example of 4 terminals



Resistor matrix

$$R = \begin{bmatrix} R_{11} & R_{12} & R_{13} & R_{14} \\ R_{21} & R_{22} & R_{23} & R_{24} \\ R_{31} & R_{32} & R_{33} & R_{34} \\ R_{41} & R_{42} & R_{43} & R_{44} \end{bmatrix}$$

Steady state equations

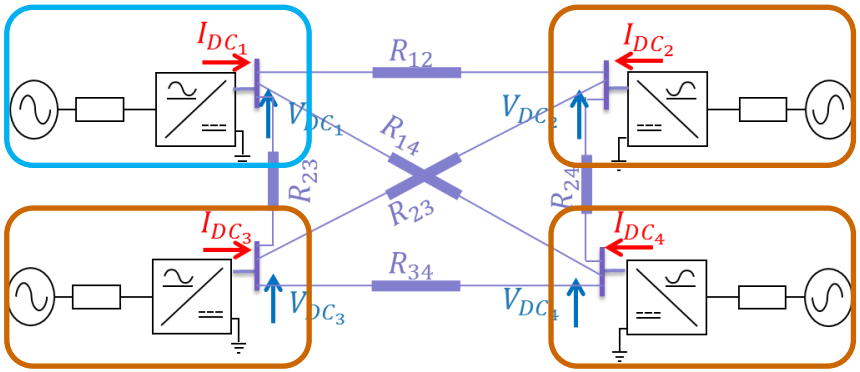


Voltage controlled

Current controlled

$$\begin{cases} I_{12} = (V_{DC_1} - V_{DC_2})/\ R_{12} \\ I_{13} = (V_{DC_1} - V_{\mathrm{DC}_3})/\ R_{13} \\ I_{14} = (V_{\mathrm{DC}_1} - V_{DC_4})/\ R_{14} \\ I_{23} = (V_{\mathrm{DC}_2} - V_{\mathrm{DC}_3})/\ R_{23} \\ I_{24} = (V_{DC_2} - V_{\mathrm{DC}_4})/\ R_{24} \\ I_{34} = (V_{\mathrm{DC}_3} - V_{DC_4})/\ R_{34} \end{cases} \quad \& \quad \begin{cases} I_{DC_1} = I_{12} + I_{13} + I_{14} + V_{\mathrm{DC}_1}/R_{11} \\ V_{\mathrm{DC}_2} = (I_{12} - I_{23} - I_{24} + I_{\mathrm{DC}_2}).\ R_{22} \\ V_{\mathrm{DC}_3} = (I_{13} + I_{23} - I_{34} + I_{DC_3}).\ R_{33} \\ V_{\mathrm{DC}_4} = (I_{14} + I_{24} + I_{34} + I_{\mathrm{DC}_4}).\ R_{44} \end{cases}$$

Find steady state solution = solve matrix equation



Voltage controlled

Current controlled

Input vector

$$\begin{bmatrix} I_{DC_1} \\ V_{DC_2} \\ V_{DC_3} \\ V_{DC_4} \\ I_{12} \\ I_{13} \\ I_{14} \\ I_{23} \\ I_{24} \\ I_{34} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & R_{22} & 0 & 0 & -R_{22} & -R_{22} & 0 \\ 0 & 0 & 0 & 0 & 0 & R_{33} & 0 & R_{33} & 0 & -R_{33} \\ 0 & 0 & 0 & 0 & 0 & 0 & R_{44} & 0 & R_{44} & R_{44} \\ 0 & -1/R_{12} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1/R_{13} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1/R_{14} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/R_{23} & -1/R_{23} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/R_{24} & 0 & -1/R_{24} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/R_{34} & -1/R_{34} & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} I_{DC_1} \\ V_{DC_2} \\ V_{DC_3} \\ V_{DC_4} \\ I_{12} \\ I_{13} \\ I_{14} \\ I_{23} \\ I_{24} \\ I_{34} \end{bmatrix} + \begin{bmatrix} 1/R_{11} & 0 & 0 & 0 \\ 0 & R_{22} & 0 & 0 \\ 0 & 0 & R_{33} & 0 \\ 0 & 0 & 0 & R_{44} \\ 1/R_{12} & 0 & 0 & 0 \\ 1/R_{13} & 0 & 0 & 0 \\ 1/R_{14} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} V_{DC_1} \\ I_{DC_2} \\ I_{DC_3} \\ I_{DC_4} \end{bmatrix}$$

Outline

1) Context
2) Find DC steady state solution
3) Initialize the DC network
4) Converter initialization
5) Initializing the overall system
6) Conclusion & Improvements

Strategy to start DC simulation from steady state

1. Input vector
   - Resistor matrix
2. Build the steady state matrix
   - Run DC steady state solution
2. Transmit SS solution to DC initializing bloc
3. Find Steady-state solution and start from steady-state
4. Display results

1 & 2 Build a Marix / LF DC

1) Develop method in JavaScript for matrix operation
2) Program a java script application for DC load flow
    1) Usable with any input data

23 Changing global data

1) Create new structure named “Station”

- AC voltage
- Active power
- Reactive power
- DC voltage

2) Get global data object

```
var cct = currentCircuit();
var attr=cct.getAttribute('GlobalDataTag');
oGlobalData=getGlobalValue(attr);
```

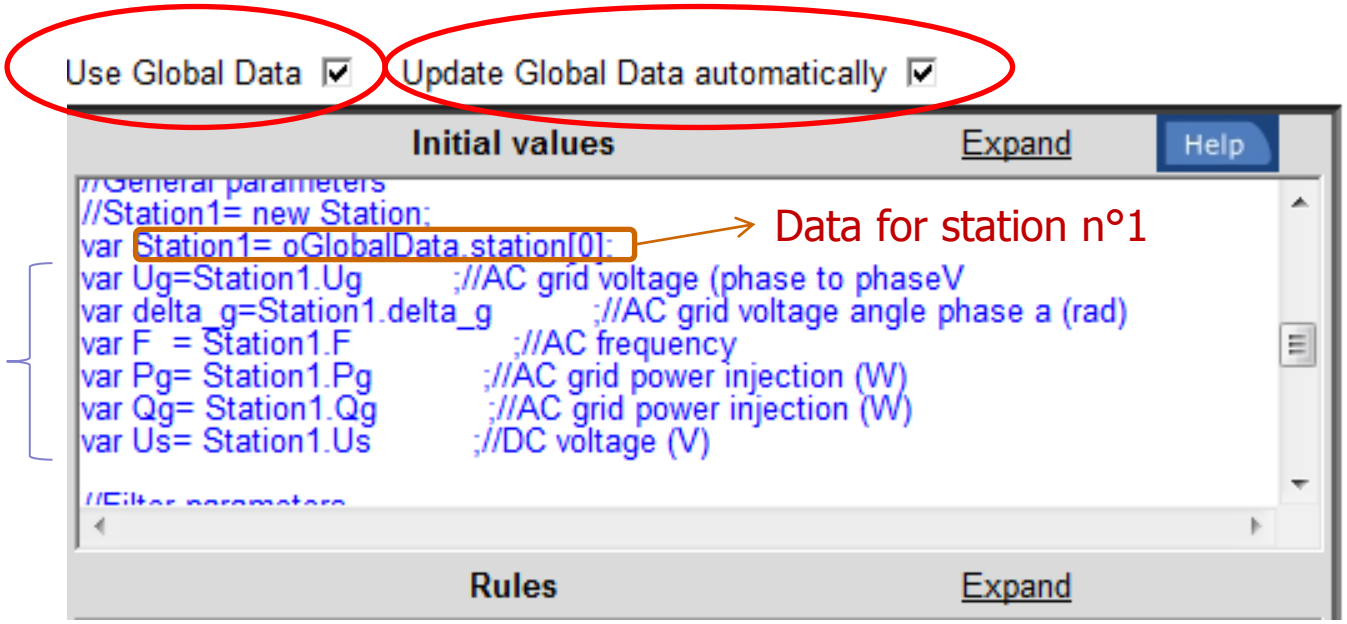
3) Save globale data

```
oGlobalData.station=station;//save new global data object
oGlobalData.confirm_device_updates=true;
cct.setAttribute('GlobalDataTag',attr);
```

4) update global data in all the circuit

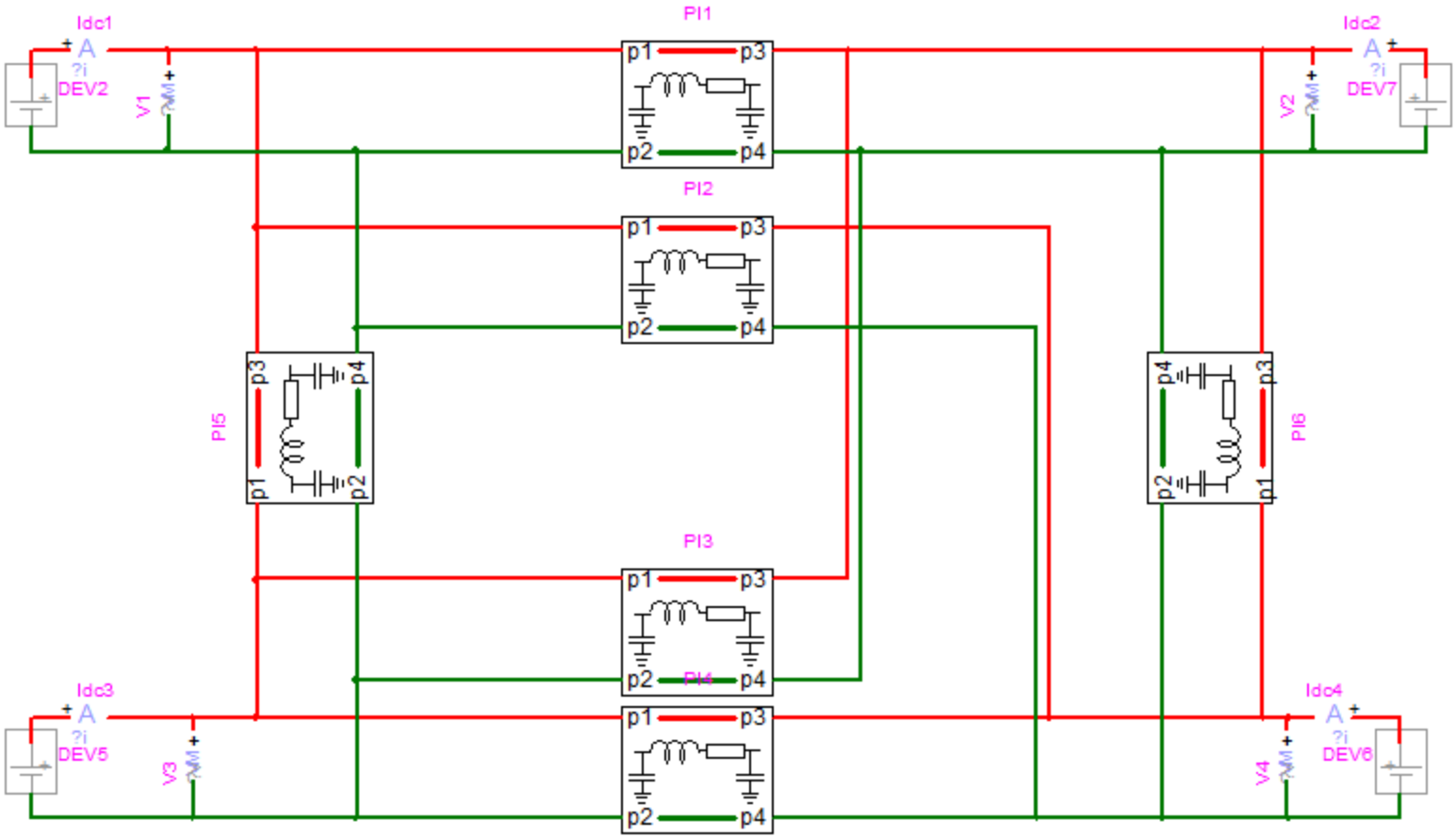
```
parseScriptFile('update_variables_in_black_boxes.dwj’)
```

23 Using a table of global data in « Black box device »



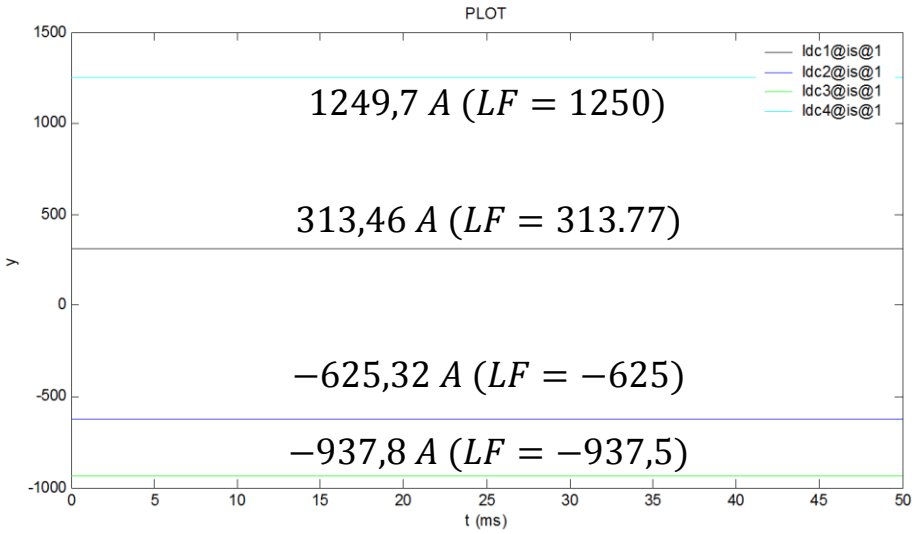
Initialization For each converter

Initialization establishing DC voltage (« Π »)

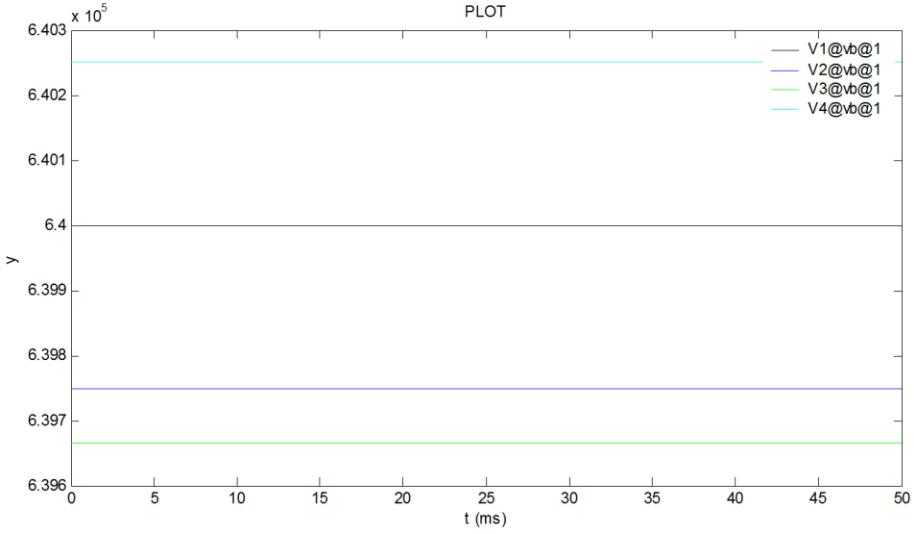


Initialization establishing DC voltage (« Π »)

Current



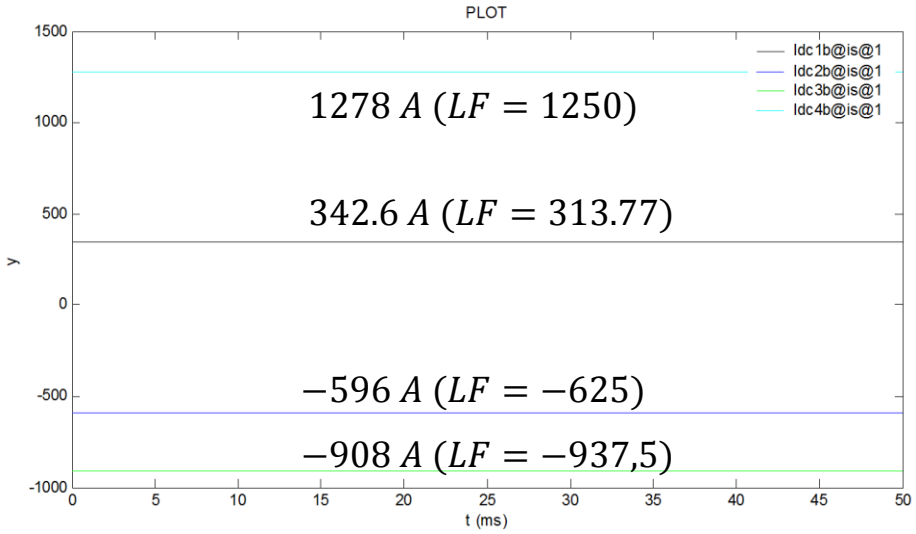
Voltage



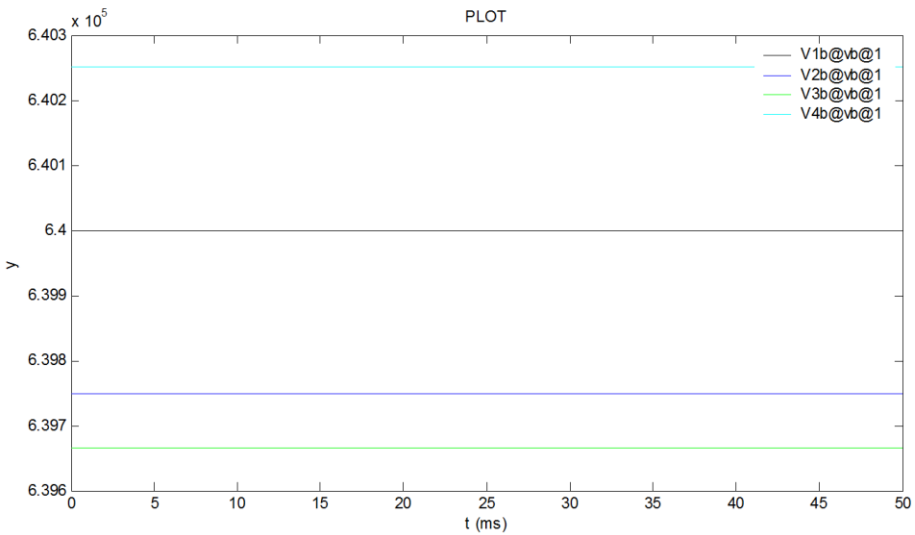
1) Currents are well initialized
2) There is no transient

Initialization establishing DC voltage (« FDQ »)

Current



Voltage



1) There is no transient
2) Small current error (<1,24%)

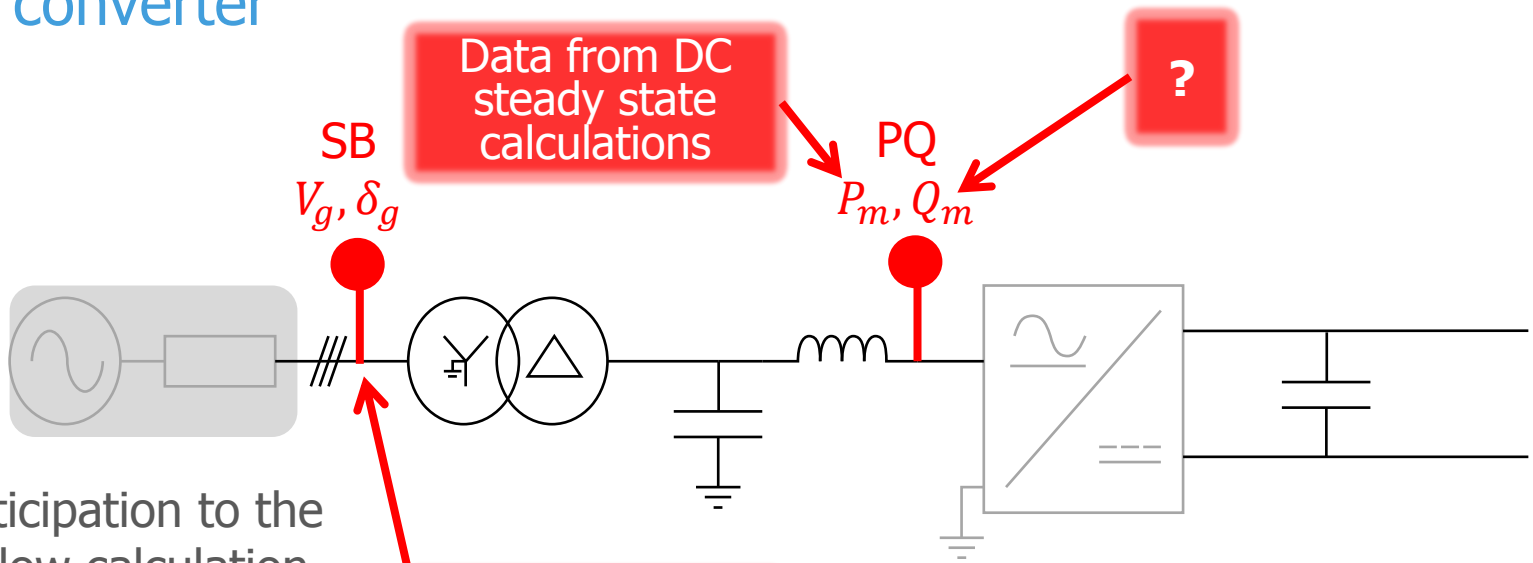
Outline

1) Context
2) Find DC steady state solution
3) Initialize the DC network
4) Converter initialization
5) Initialize the overall system
6) Conclusion & Improvements

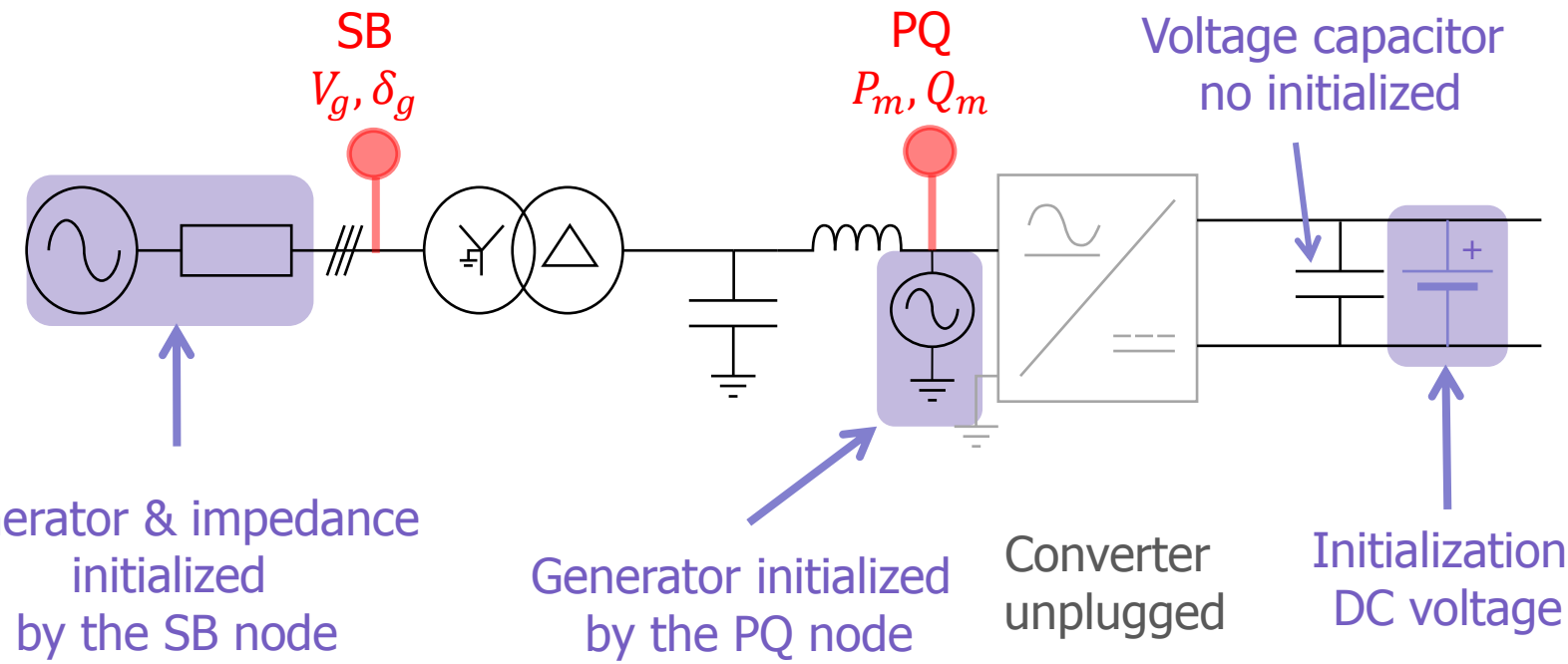
How start a time-domain solution using converter AC/DC?

1. Put AC load flow bus
   - Fill it with DC results
2. Start the EMTP-rv's AC load flow
3. Start Steady-state solution from Load-Flow solution
4. Step between steady-state solution and time simulation

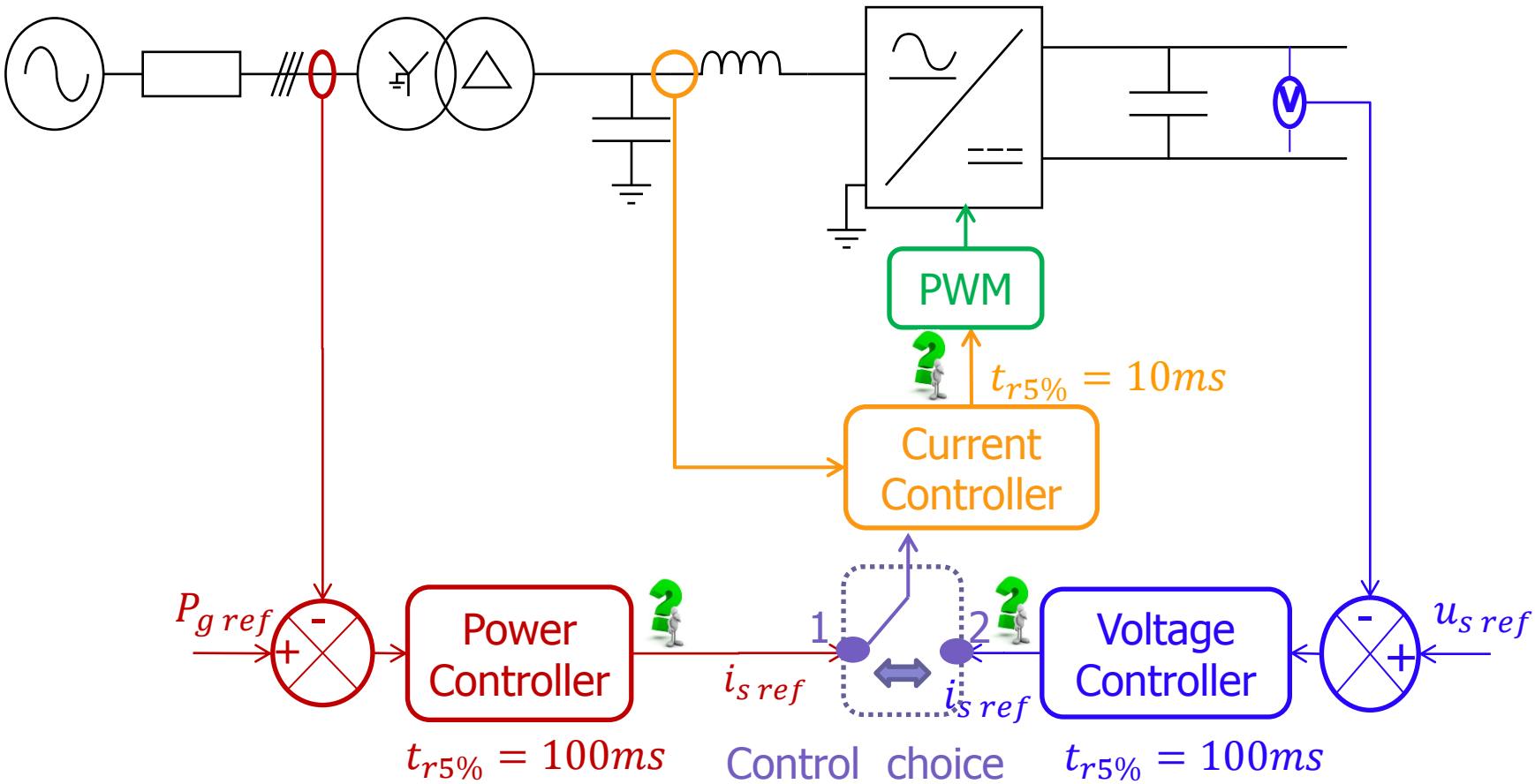
AC Load Flow for initializing the AC part of each converter



Starting simulation (Power part)



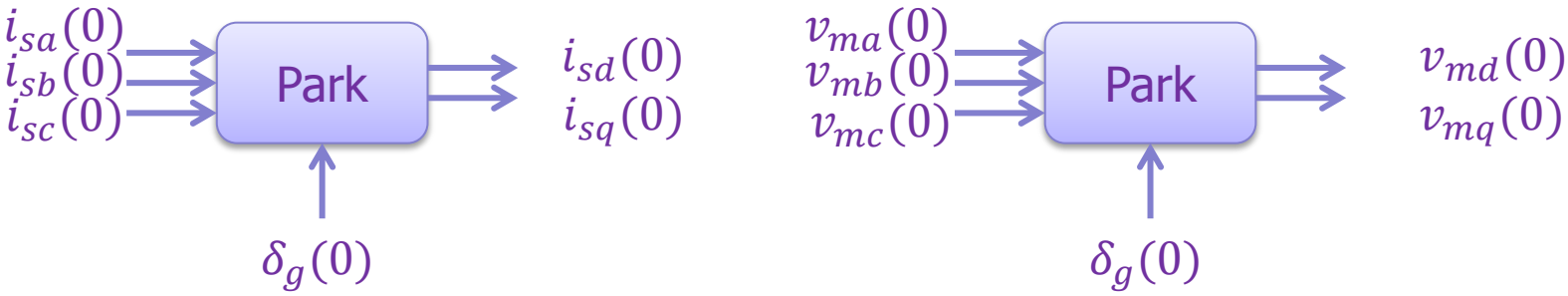
Initialization of the control part



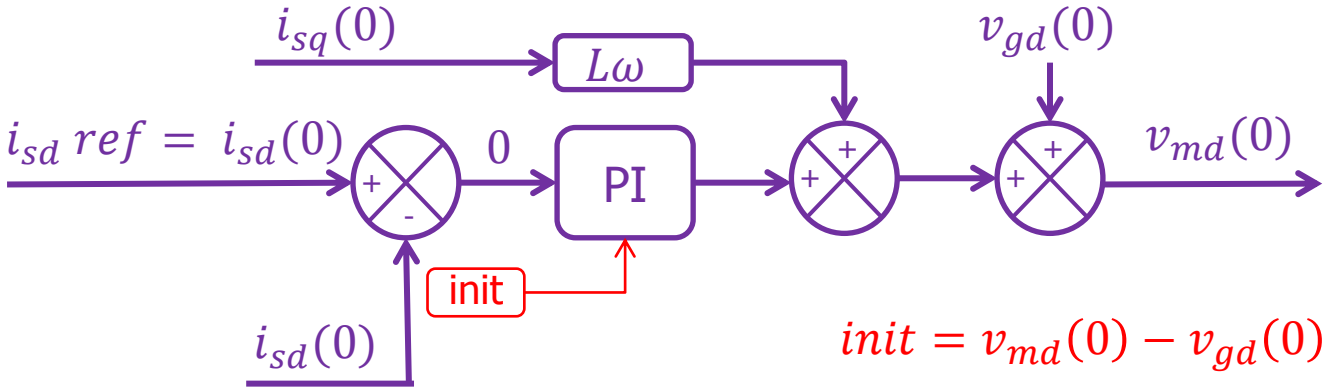
Controllers have to be initialized

How initialize a controller from EMTP steady state solution?

Measures & transformations



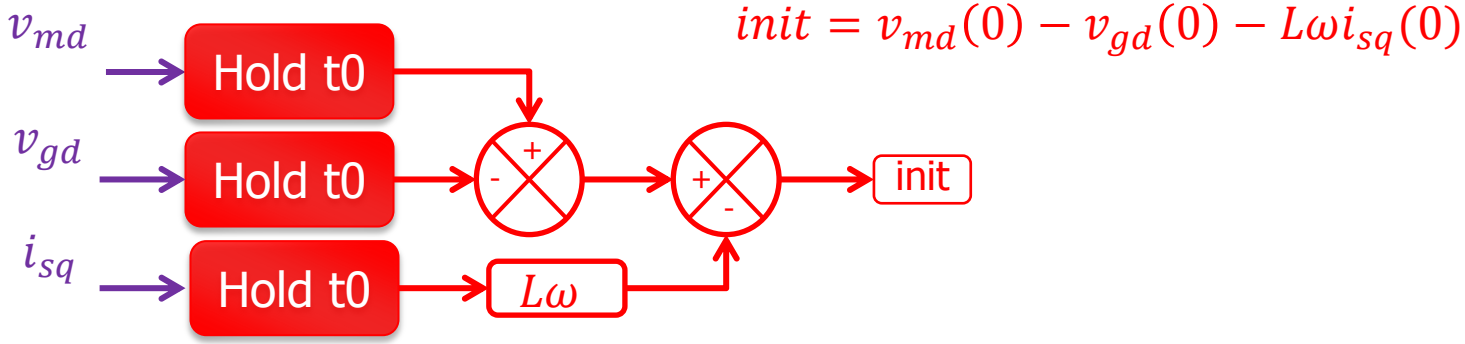
Example : Cuurent controller



$$init = v_{md}(0) - v_{gd}(0) - L\omega i_{sq}(0)$$

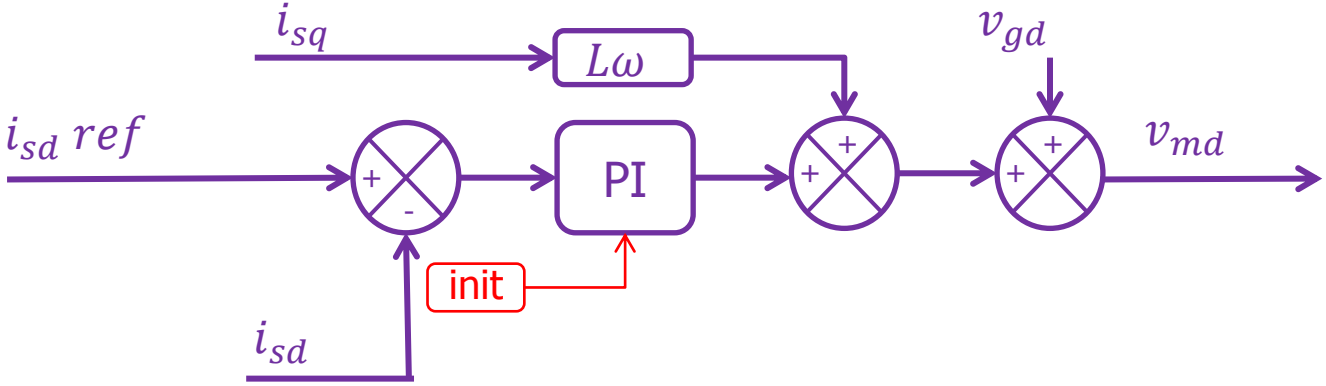
Initializing example: Current controller

Initialization block diagram



$$init = v_{md}(0) - v_{gd}(0) - L\omega i_{sq}(0)$$

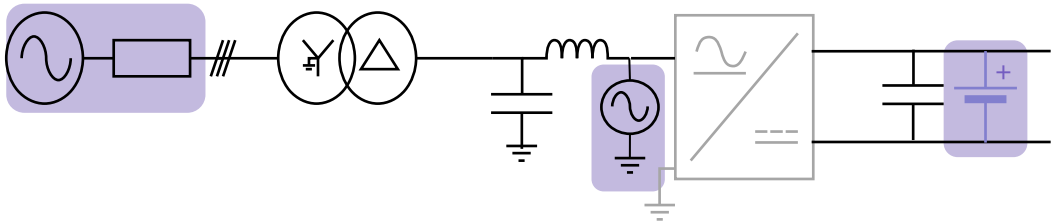
Current controller



Transition initialization/simulation

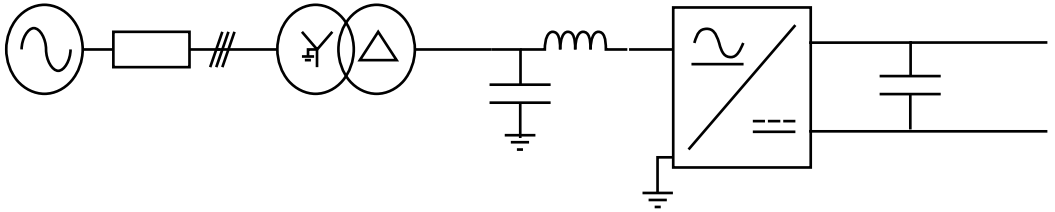
t=0 (steady state solution)

Converter unplugged



t>0

Converter plugged



- Ideal switches are used between 2 configurations
- Change before the first calculation step

Outline

1) Context
2) Find DC steady state solution
3) Initialize the DC network
4) Converter initializing
5) Initialize the overall system
6) Conclusion & Improvements

Time-domain simulation with MTDC and AC grids

Procedure to follow

1. Start DC steady state solution
   - Put results in global data
2. Fill PQ node with corresponding global data
   - Start AC load flow
3. Start Steady-state solution from Load-Flow solution

Outline

1) Context
2) Find DC steady state solution
3) Initialize the DC network
4) Converter initializing
5) Initializing the overall system
6) Conclusion & Improvements

Conclusion

1) Calculation of a DC steady state in JavaScript (Matrix operations)

2) Creation of global data in a JavaScript file

3) Use of these data to initialize a AC load flow

4) Initialization of controllers from AC load flow results

5) Startup of time-domain simulation AC/DC from steady state

Improvement & further work

1) Initialize measure filters

2) Initialize a detailed converter

3) Use a file to initialize simulation data

4) Use EMTP-rv features to initialize DC

Réseau de transport d'électricité

LILLE

Thank you for your attention!

APPENDIXES

Structure « Station »

```
function Station(){
    this.Ug=320e3;        //AC grid voltage phase to phase (V)
    this.delta_g=0;       //AC grid voltage angle phase a (rad)
    this.F=50;            //AC frequency (Hz)
    this.Pg=400e6;        //AC grid power injection (W)
    this.Qg=200e6;        //AC grid power injection (W)
    this.Us=640e3;        //DC voltage (V)
}
```

Put LF results in table of structure

```
var station=[];
for(var i=0;i<Vdc.dimensions();i++){
    station[i]= new Station();//creat a variable station
    station[i].Us=2*Vdc.e(i+1);//change DC voltage value
    station[i].Pg=2*Vdc.e(i+1) * Idc.e(i+1) * (-1); //change AC active power
}
```

Each station is represented by a structure of index i