EASY-RES

SUMMER SCHOOL
"ENABLING DRES TO OFFER ANCILLARY SERVICES"
20TH – 24TH SEPTEMBER 2021

Technical constraints and optimization in distribution grids and converters

Álvaro Rodríguez del Nozal // Block 7.A

This project has received funding from the European Union´s Horizon 2020 Programme for research and innovation under Grant Agreement no 764090.

Agenda

- Active distribution model
  - Network description
  - Constraints for a raliable operation of the network
- ADN model impementation guidelines
  - Object-based program structure
  - Solving a power flow problem
- VSC electrothermal model
  - Power losses
  - Thermal model

Active distribution network model

- Definition of a model of an ADN with a single point of interconnection with the transmission system.
  - Immediate extension to more POIs.
  - The network model is defined by the Kirchhoff laws and power expression.
  - Power lines are modeled by a RL circuit (low voltage and short length).
- Control variables:
  - Active and reactive power injections of DRESs (grid-feeding).
  - Voltage fixed by the DRESs (grid-forming).
- Constraints: voltage limits, current constraints, active power restrictions, …
  - Depend on the control variables.

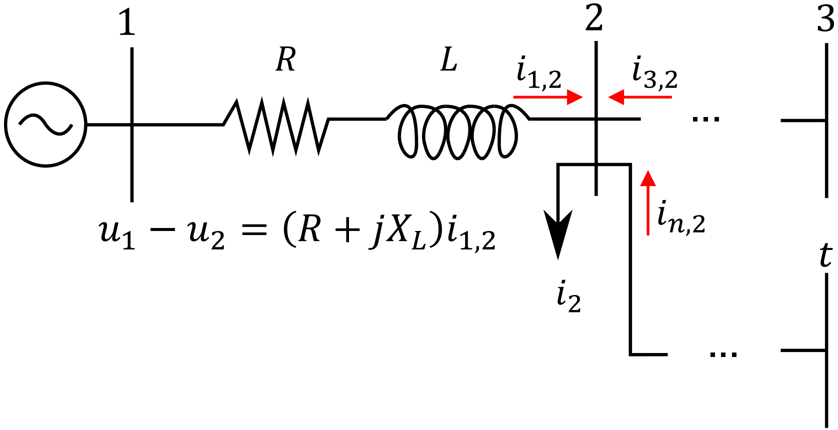
Control variables with a given value:
POWER FLOW
Outcome variables: voltage, currents, …

Control variables as a searched vector:
OPTIMAL POWER FLOW
Additional constraints come into play

Active distribution network model

Network description

- The equations that describe the model of the network are given by the voltage and current Kirchhoff laws



$$u_1 - u_2 = (R + jX_L)i_{1,2}$$

$$i_{1,2} + i_{3,2} + \cdots + i_{t,2} = i_2$$

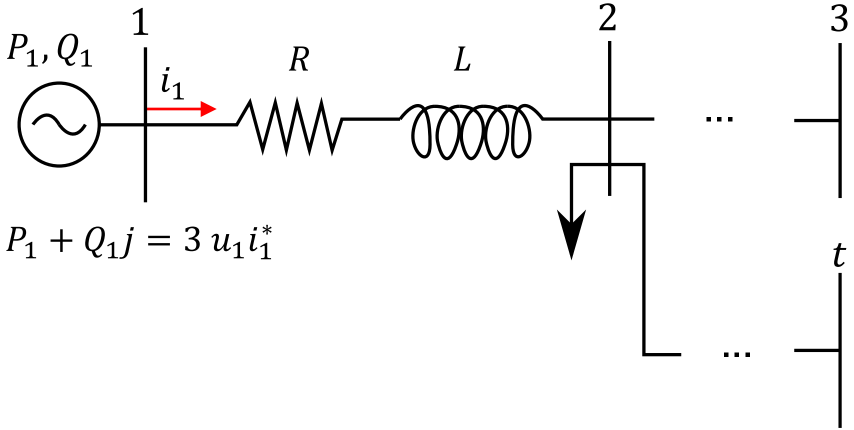
$$\sum_k i_{k,n} = i_n$$

$$u_k - u_n = (R + jX_L)i_{k,n}$$

Active distribution network model

Network description

- Power injections performed by the DRESs are also considered in the problem.



$$P_1 + Q_1 j = 3\, u_1 i_1^*$$

$$\sum_k i_{k,n} = i_n$$

$$u_k - u_n = (R + jX_L)\, i_{k,n}$$

$$P_k + Q_k j = 3\, u_k i_k^*$$

Power Flow problem

- **Given the active and reactive power injections** by each DRES and those demanded by the loads at every bus of the ADN, obtain the rest of variables of the problem:
  - Voltages.
  - Currents flowing through power lines.
  - Current injections by the DRESs.
  - Current demanded by the loads.

$$\sum_k i_{k,n} = i_n$$

$$P_k + Q_k j = 3\, u_k i_k^*$$

→ 4 equations/bus

$$u_k - u_n = (R + jX_L) i_{k,n}$$

→ 2 equation/line

Power Flow problem

- **Given the active and reactive power injections** by each DRES and those demanded by the loads at every bus of the ADN, obtain the rest of variables of the problem:
  - Voltages.
  - Currents flowing through power lines.
  - Current injections by the DRESs
  - Current demanded by the loads.

Data

$$\sum_k i_{k,n} = i_n$$

$$P_k + Q_k j = 3\, u_k i_k^*$$

$$u_k - u_n = (R + jX_L) i_{k,n}$$

4 equations/bus

2 equation/line

2 variables/line

2 variables/bus

2 variable/bus

- n buses
- m lines

System of non-linear equations with $4n + 2m$ equations and searched variables

Active distribution network model

Power Flow problem

- **Given the voltages fixed by each DRES** and the power demanded by the loads at every bus of the ADN, obtain the rest of variables of the problem:
  - **Rest** of the voltages.
  - Currents flowing through power lines.
  - Current injections by the DRESs and **active and reactive power injections**.
  - Current demanded by the loads.

Data

$$\sum_k i_{k,n} = i_n$$

$$P_k + Q_k j = 3\, u_k i_k^*$$

$$u_k - u_n = (R + jX_L) i_{k,n}$$

- 2 variables/line
- 4 equations/bus
- 2 variables/bus
- 2 equation/line
- 2 variable/bus without DRES
- 2 variable/bus with DRES

- n buses
- m lines

System of non-linear equations with $4n + 2m$ equations and searched variables

Optimal Power Flow problem

- **Determine the active and reactive power injections of the DRESs** in order to optimally operate the ADN.
  - Possible goal: minimize power losses in the network while ensuring an active power injection at the point of interconnection with the transmission system.

Data for loads

2 control variables for each DRES

$$\sum_k i_{k,n} = i_n$$

$$P_k + Q_k j = 3\, u_k i_k^*$$

$$u_k - u_n = (R + jX_L) i_{k,n}$$

- 4 equations/bus
- 2 equation/line
- 2 variables/line
- 2 variables/bus
- 2 variable/bus

- n buses, $\hat{n}$ buses with DRESs
- m lines

System of non-linear equations with
- $4n + 2m$ equations
- $4n + 2m + 2\hat{n}$ variables

$$\min P_{loss} = min \sum_k P_k$$

Optimal Power Flow problem

- **Determine voltage setpoints of the DRESs** in order to optimally operate the ADN.
  - Possible goal: minimize power losses in the network while ensuring an active power injection at the point of interconnection with the transmission system.

Data for loads

$$\sum_k i_{k,n} = i_n$$

$$P_k + Q_k j = 3\, u_k i_k^*$$

$$u_k - u_n = (R + jX_L) i_{k,n}$$

4 equations/bus

2 equation/line

2 variables/line

2 variables/bus

2 variables/bus with DRES

2 variablea/bus without DRES

2 control variables for each DRES

- n buses, $\hat{n}$ buses with DRESs
- m lines

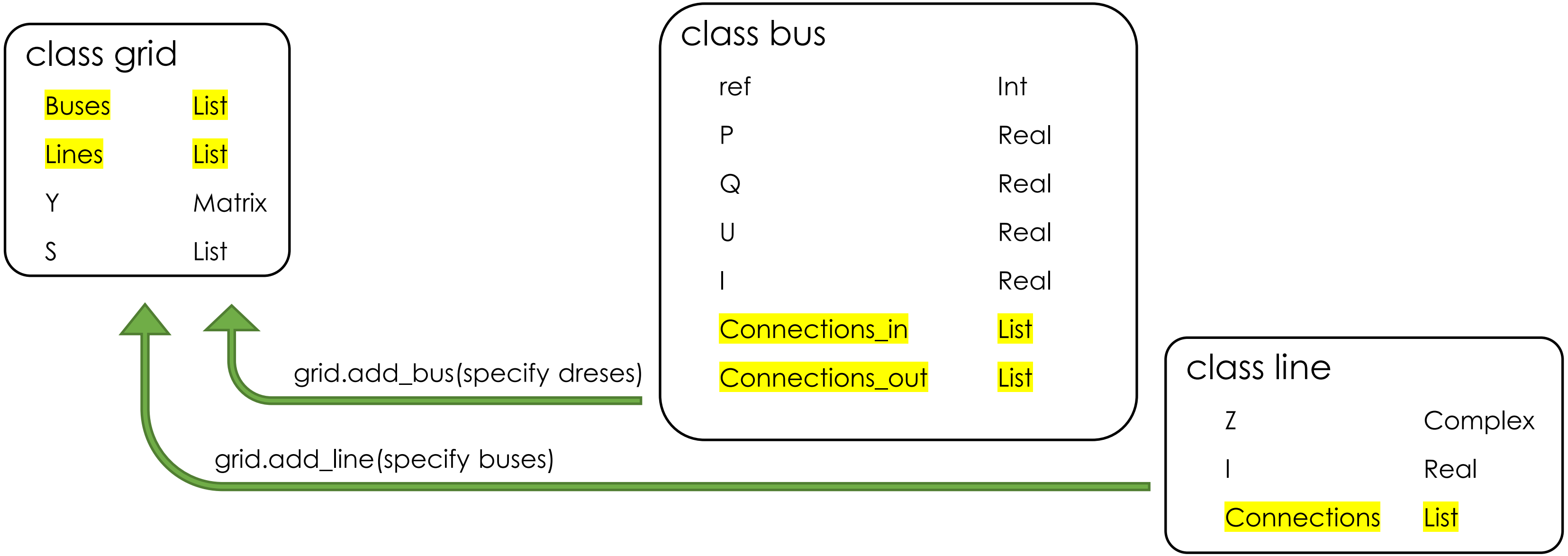
System of non-linear equations with

- $4n + 2m$ equations
- $4n + 2m + 2\hat{n}$ variables

$$\min P_{loss} = min \sum_k P_k$$

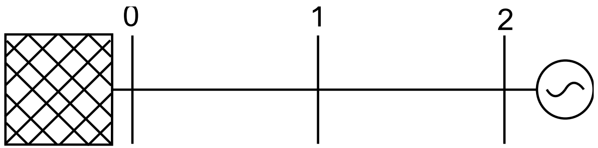
ADN model impementation guidelines

Object-based program structure



Object-based program structure

- An explanatory example



Line added to bus.connections
Bus added to line.connections

```
net = grid()
net.add_bus(ref = 0, slack = True)
net.add_bus(ref = 1)
net.add_bus(ref = 2, P = 1e6, Q = 0)
net.add_line(bus_0 = 0, bus_1 = 1, Z = Z01)
net.add_line(bus_0 = 1, bus_1 = 2, Z = Z12)
```

- All the elements are accesible from the net object.
- Every bus has access to all the lines connected to itself.
- Every line has access to all the buses connected to itself.
- Every element is accesible from every other element.

```
net.buses[1].connections
➢ [line_0_1, line_1_2]
net.buses[1].connections[0]
➢ line_0_1
net.buses[1].connections[0].connections
➢ [bus0, bus1]
```

Solving a power flow problem

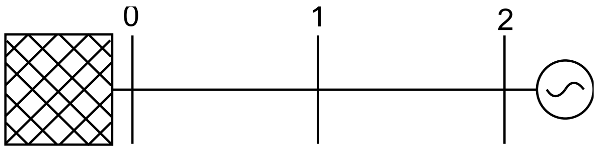
- Once the objects that describe the network has been created, the power flow problem can be solved.
- Use a nonlinear solver that from a staring estimate of a searched vector x, returns the residuals of the nonlinear system of equations to be solved.
    - The searched vector can be the voltage values at each bus (nonlinear solvers do not admit complex numbers).
- Easy solution (but computationally inefficient): generate a method in the class grid that perform the following steps to evaluate x:
    - Takes the values from x and assign them to buses attributes.
    - Each line object assign the current from the voltage Kirchhoff law.
    - Each bus object computes the residual of current Kirchhoff law.
    - A vector with all the residuals is returned to the solver.

ADN model impementation guidelines

https://github.com/alvarordn/AC-Power-Flow-PQ-.git

Solving a power flow problem

- An explanatory example



1

$$x = [u_0^{real}, u_0^{imag}, u_1^{real}, u_1^{imag}, u_2^{real}, u_2^{imag}]$$

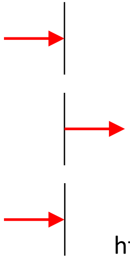
```
net.buses[0].Uc = x[0] + x[1]j
net.buses[1].Uc = x[2] + x[3]j
net.buses[2].Uc = x[4] + x[5]j
```

2

```
Bus0_u = net.lines[0].connections[0].Uc
Bus1_u = net.lines[0].connections[1].Uc
net.lines[0].I  = (Bus1_u – Bus0_u)/net.lines[0].Z
```

3

```
Residual = list()
for bus in net.buses:
    I = bus.I
    for line in bus.connections:
        if bus == line.connections[0]:
            I -= line.I
        else:
            I += line.I
Residual.append(I)
```



4

```
return Residual
```

```
from scipy import fsolve
fsolve(func, x0)
```

https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.fsolve.html

Solving an optimal power flow problem

- A similar approach can be followed:
  - A function/method that envelopes the equations of the network is needed. Returns the residual.
  - Introducing a function/method with the inequalities that must be fulfilled.
  - Defining a function/method with the objective function.
- All the new functions compute the corresponding requirements from the searched vector x.

```
from scipy.optimize import minimize
minimize(func, x0, bounds = None, constraints = ())
```

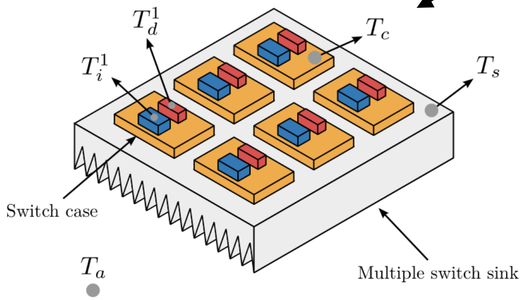
Scipy implements several methods to solve optimization problems depending on the constraints and bounds of the problem

https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.minimize.html

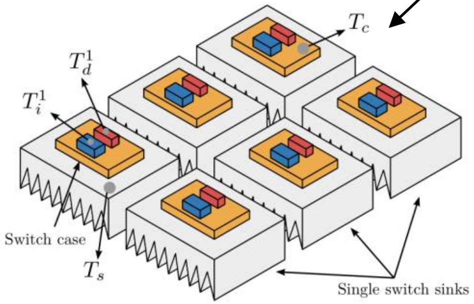
VSC electrothermal model

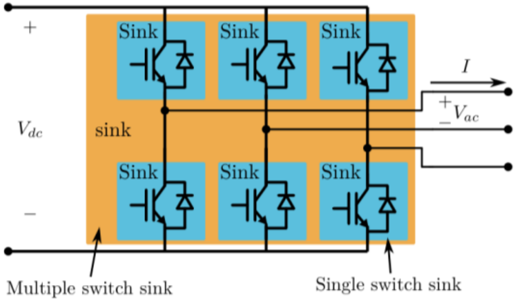
- Two-level VSC model based on switches with at least one IGBT and its corresponding antiparallel diode.
- The model estimates the temperature of IGBTs and diodes considering the injected current, power factor and environmental conditions.

Under 1 MW



Large converters



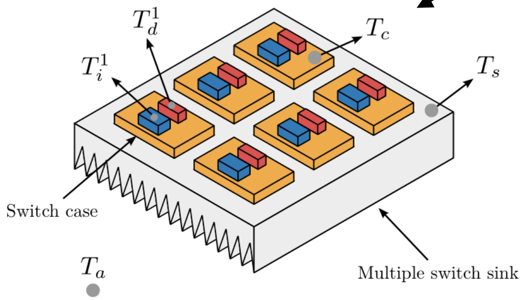


VSC electrothermal model

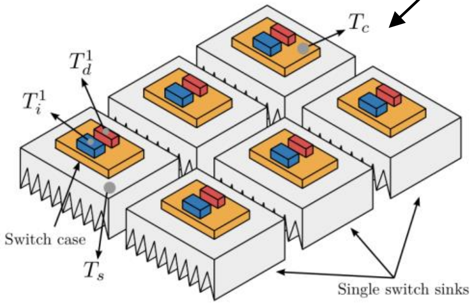
Advantages compraed to manufacturers‘ tools

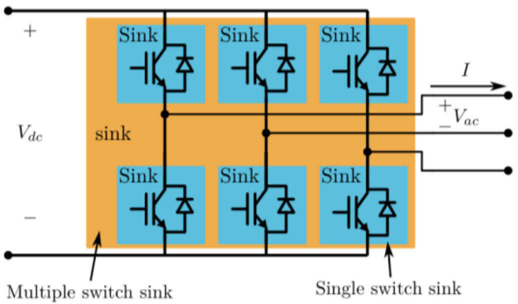
- There are several manufacturers with on-line applications or stand-alone programs to compute the losses and thermal behaviors of their respective devices.
- The proposed model allows to integrate power losses as well as temperature signals of the power converters inside the same tool and making them accessible to simulate protections or controllers.

Under 1 MW



Large converters





VSC electrothermal model

Power losses model

- Power losses in the switches and diodes are the main heat sources.
- Power losses depend on AC, current, DC and AC voltages, switching frequency, ….
  - Fixed switching frequency control strategies are the most common.
  - DC voltage is quasi-constant.
  - Ambient temperature can be considered constant.
- Under these hypothesis, power losses can be computed as a function of the RMS AC current.

$$P_i(I) = \Theta\left[a_i + (b_i + c_i\alpha)\, I + (d_i + e_i\alpha)\, I^2\right]$$
$$P_d(I) = \Theta\left[a_d + (b_d + c_d\alpha)\, I + (d_d + e_d\alpha)\, I^2\right]$$

$$\Theta = \frac{T_s}{T_s^{ss}}$$

Transient computation:
Actual/steady-state heatsink temperature

$$\alpha = m \cdot \cos(\phi)$$

Power factor: $\cos(\phi)$

Modulation index: $m = \dfrac{2\sqrt{2}\frac{400}{\sqrt{3}}}{730}$

VSC electrothermal model

Computation of the parameters

$$P_i(I) = \Theta\left[a_i + (b_i + c_i\alpha) I + (d_i + e_i\alpha) I^2\right]$$
$$P_d(I) = \Theta\left[a_d + (b_d + c_d\alpha) I + (d_d + e_d\alpha) I^2\right]$$

$$\begin{bmatrix} 1 & I_1 & I_1\alpha_1 & I_1^2 & I_1^2\alpha_1 \\ 1 & I_2 & I_2\alpha_1 & I_2^2 & I_2^2\alpha_1 \\ 1 & I_3 & I_3\alpha_1 & I_3^2 & I_3^2\alpha_1 \\ 1 & I_4 & I_4\alpha_2 & I_4^2 & I_4^2\alpha_2 \\ 1 & I_5 & I_5\alpha_2 & I_5^2 & I_5^2\alpha_2 \end{bmatrix}\begin{bmatrix} a_i \\ b_i \\ c_i \\ d_i \\ e_i \end{bmatrix} = \begin{bmatrix} P_i(I_1) \\ P_i(I_2) \\ P_i(I_3) \\ P_i(I_4) \\ P_i(I_5) \end{bmatrix}$$

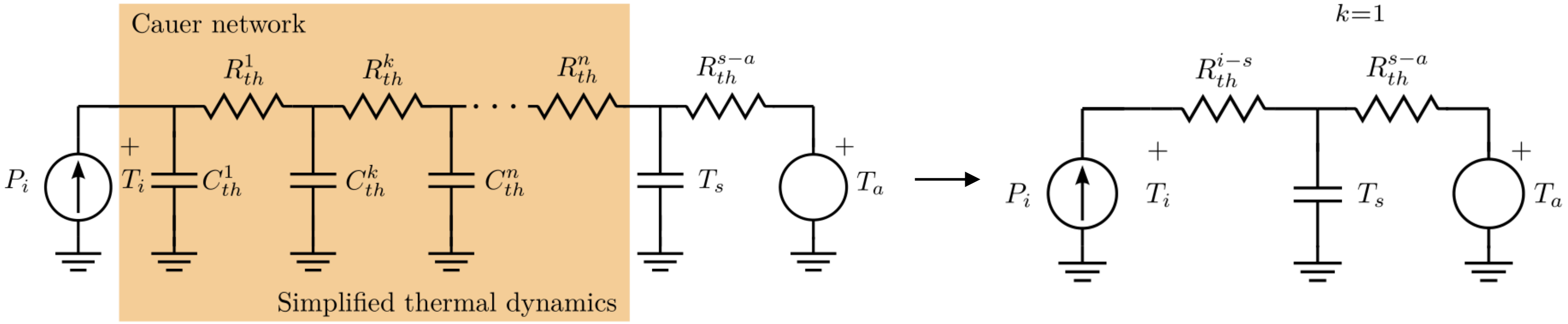
$$\begin{bmatrix} 1 & I_1 & I_1\alpha_1 & I_1^2 & I_1^2\alpha_1 \\ 1 & I_2 & I_2\alpha_1 & I_2^2 & I_2^2\alpha_1 \\ 1 & I_3 & I_3\alpha_1 & I_3^2 & I_3^2\alpha_1 \\ 1 & I_4 & I_4\alpha_2 & I_4^2 & I_4^2\alpha_2 \\ 1 & I_5 & I_5\alpha_2 & I_5^2 & I_5^2\alpha_2 \end{bmatrix}\begin{bmatrix} a_d \\ b_d \\ c_d \\ d_d \\ e_d \end{bmatrix} = \begin{bmatrix} P_d(I_1) \\ P_d(I_2) \\ P_d(I_3) \\ P_d(I_4) \\ P_d(I_5) \end{bmatrix}$$

VSC electrothermal model

Thermal model

- Once the power losses of IGBTs and diodes are known, it can be computed the thermal behavior of the VSC.
- Differential equations with fast dynamics (the capacitors at the shaded areas) are transformed into algebraic equations.
  - The fast dynamics are not neglected but considered in steady-state.
  - This simplification leads to conservative results.

$$R_{th}^{i-s} = \sum_{k=1}^{n} R_{th}^{k}$$



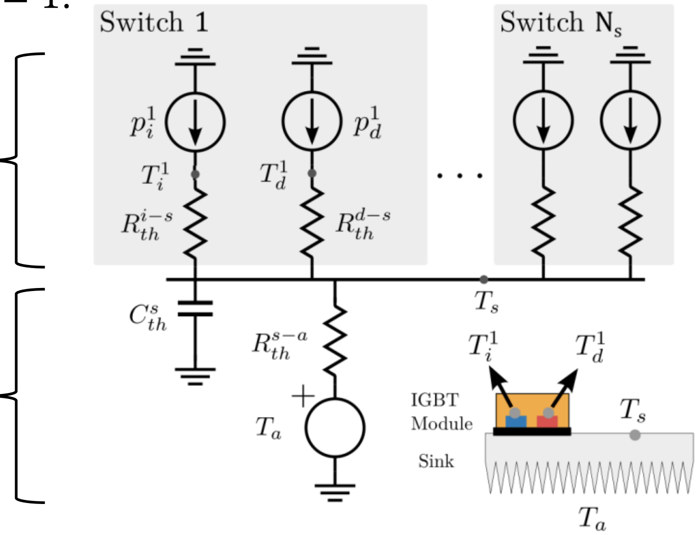
VSC electrothermal model

Thermal model

- In the case that all the switches share the same heatsink the model can be reduced to the following equations.
- In case of a single heatsink for each switch $N_s = 1$.

$$T_i = T_s + P_i R_{th}^{i-s}$$
$$T_d = T_s + P_d R_{th}^{d-s}$$

$$\frac{dT_s}{dt} = \frac{1}{C_{th}^s}\left(N_s P_s - \frac{T_s - T_a}{R_{th}^{s-a}}\right)$$
$$P_s = P_i + P_d$$
$$T_s^{ss} = T_a + N_s P_s R_{th}^{s-a}$$

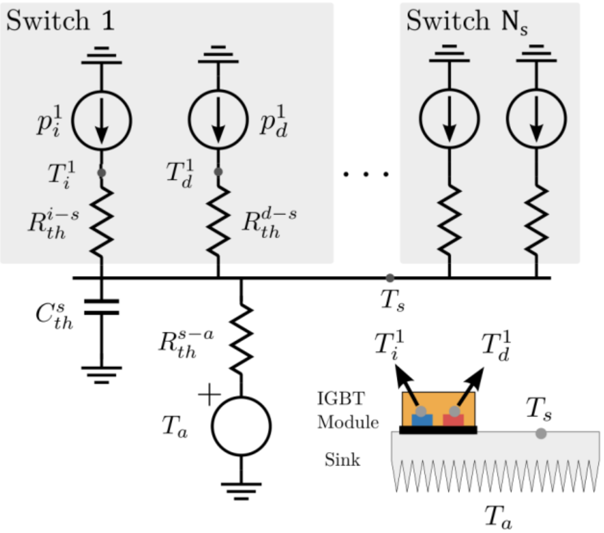


VSC electrothermal model

Computation of the parameters

- Junctures to heatsink thermal resistors: $R_{th}^{i-s} = \frac{T_i - T_s}{P_i}$ $\quad R_{th}^{d-s} = \frac{T_d - T_s}{P_d}$

- Heatsink to ambient thermal model parameters: $R_{th}^{s-a} = \frac{T_s^{ss} - T_a}{P_i + P_d}$

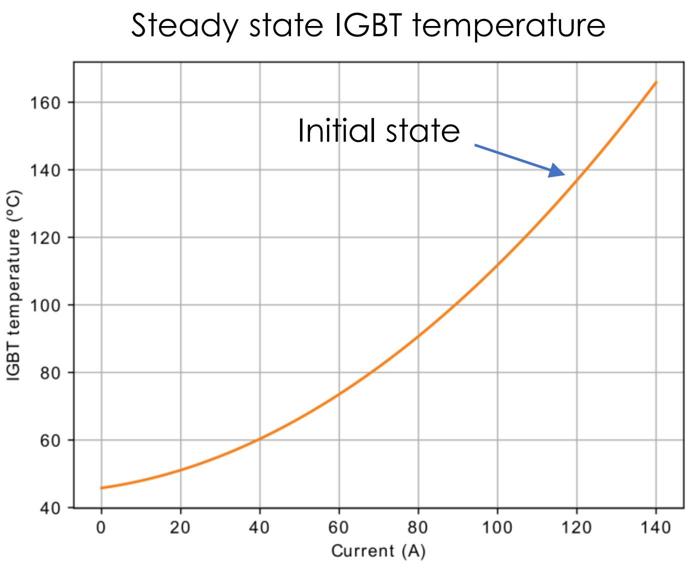


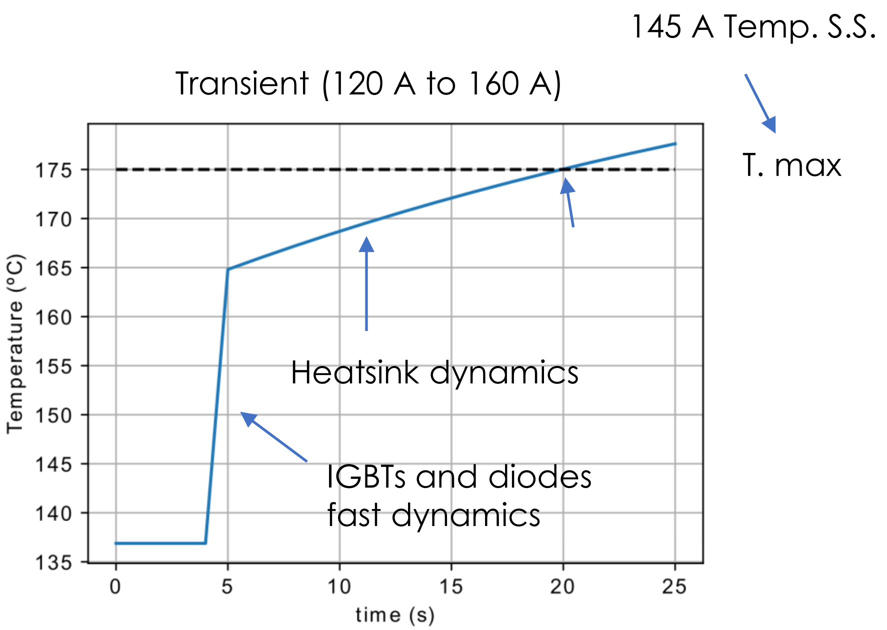
$$\tau_s = R_{th}^{s-a} C_{th}^{s} \Rightarrow C_{th}^{s} = \frac{\tau_s}{R_{th}^{s-a}}$$

VSC electrothermal model

Some simulations results

- Application: provision of inertial response.





References

Publications

- More information is accessible in Deliverable 2.2 of the EASY-RES project:
    - Mauricio, J., Kontis, Barragán-Villarejo M., Casanova Delgado, L., et al. First report on the electric and thermal dynamic modelling of DRES/BESS. Deliverable D2,2 of EU H2020 project EASY-RES – Enable Ancillary Services by Renewable Energy Sources, 2020.

Item 4: The EASY-RES Consortium

The Consortium

ARISTOTLE UNIVERSITY OF THESSALONIKI

UNIVERSIDAD DE SEVILLA

UNIVERSITÄT PASSAU

Lancaster University

TU Delft

ΑΔΜΗΕ
ΑΝΕΞΑΡΤΗΤΟΣ ΔΙΑΧΕΙΡΙΣΤΗΣ ΜΕΤΑΦΟΡΑΣ ΗΛΕΚΤΡΙΚΗΣ ΕΝΕΡΓΕΙΑΣ

elektro gorenjska

Stadtwerke LANDAU a.d.ISAR

stadtwerk haßfurt

fenecon ENERGY ENGINEERING

ZD.B ZENTRUM DIGITALISIERUNG. BAYERN

This project has received funding from the European Union´s Horizon 2020 Programme for research and innovation under Grant Agreement no 764090.

Thank you!

Álvaro Rodríguez del Nozal

Affiliation: EASY-RES
Phone: +39
E-Mail: arnozal@us.es
EASY-RES website: http://www.easyres-project.eu/