SINTEF

SINTEF Energi AS
SINTEF Energy Research
Address:
Postboks 4761 Torgarden
NO-7465 Trondheim
NORWAY
Switchboard: +47 45456000

energy.research@sintef.no

Enterprise /VAT No:
NO 939 350 675 MVA

Project memo

Battery Module Liquid-cooled Heat Sink

Modelling and Design

VERSION 2	**DATE** 2023-04-12

AUTHOR(S)
René Alexander Barrera-Cárdenas

CLIENT(S) SEABAT	**CLIENTS REF.** SEABAT
PROJECT NO. 502002730	**NO. OF PAGES AND APPENDICES:** 29

ABSTRACT
This memo describes the model, design methodology and implementation (Matlab Scripts) of the battery module water-cooled heatsink to be used within an optimization algorithm for optimal design of modular battery systems.

PREPARED BY René Alexander Barrera-Cárdenas	SIGNATURE RENE A. BARRERA
APPROVED BY Olve Mo	SIGNATURE Olve Mo (Apr 13, 2023 12:59 GMT+2)
PROJECT MEMO NO. AN 21.12.46	**CLASSIFICATION** Unrestricted

Document history

VERSION	DATE	VERSION DESCRIPTION
1.0	2021-11-04	First version for comments
1.1	2022-10-20	Updated list of variables
2	2023-04-12	Modification to the model/algorithm to included/consider a thermal pad between fin-base and cold plates.

Table of contents

1 Introduction

This memo describes the model, design methodology and implementation (Matlab Scripts) of the battery module water-cooled heatsink to be used within an optimization algorithm for optimal design of modular battery systems.

2 List of inputs, variables, and parameters.

2.1 Design Inputs

n_{cell}	Number of cells per module
$R_{thHSMax}$	Required (maximum/limit) heatsink thermal resistance per cell.

2.2 Free design parameters

t_{CP}	Cold plate thickness.
t_{Fin}	Fin thickness
V_{FR}	Inlet water flow rate

2.3 Design Variables

A_{BCP}	Cold plate base area
$Cost_{HS}$	Heatsink total cost
$Cost_{TP}$	Total cost for the thermal pads
$Cost_{Fin}$	Total cost for the fin/base conductors
$Cost_{CP}$	Total cost for the cold plates
L_{CP}	Total length of the top and bottom cold plates
$Mass_{HS}$	Heatsink total weight
$Mass_{Fin}$	Total weight of the thermal conductive fin and base
$Mass_{TP}$	Total weight of the thermal pad material between cell and fin/base
$Mass_{CP}$	Total cold plates weight
R_{thB}	Base thermal resistance
R_{thBi}	Fin-base thermal resistance (equivalent for the fin thermal path)
R_{thCP}	Cold plate thermal resistance
R_{thFin}	Fin thermal resistance
$R_{thHS.FIN}$	Average thermal resistance per cell of the fin-cooling system
R_{thTPx}	Thermal pad thermal resistance between cell and fin
R_{thTPy}	Thermal pad thermal resistance between cell and /base
R_{thTPcp}	Thermal pad thermal resistance between base and cold plate
$R_{thTPCPi}$	Cold plate and thermal pad thermal resistance per cell assuming uniform heat distribution along the cold plate surface
t_{base}	Fin base thickness
Vol_{BM}	Battery module volume (heatsink including the battery cells)
Vol_{CP}	Cold plate volume
W_{CP}	Total width of the top and bottom cold plates
$V_{FR.Min}$	Minimum water flow rate
$V_{FR.Max}$	Maximum water flow rate

2.4 Design constants, parameters

Battery cell properties

L_{cell} Cell length

t_{cell} Cell thickness

W_{cell} Cell width

$R_{thCellx}$ In-plane cell thermal resistance (from centre of the cell to the cell surface $W_{cell}xL_{cell}$)

$R_{thCelly}$ Trough-plane cell thermal resistance (from centre of the cell to the cell surface $W_{cell}xt_{cell}$)

κ_{Cellx} Equivalent/average in-plane cell thermal conductivity

κ_{Celly} Equivalent/average trough-plane cell thermal conductivity

Fin thermal conductor

$Cost_{Fin0}$ Unit cost per fin

$Cost_{Fin1}$ Cost per kg of the fin material

κ_{Fin} Thermal conductivity of the fin thermal conductor material

α_{Fin} Thermal spreading angle for the fin thermal conductor material

ρ_{Fin} Density of the thermal conductive fin material

Thermal pad

$Cost_{TP1}$ Cost per kg for the thermal pad material

t_{TP} Thermal pad thickness

κ_{TP} Thermal conductivity of the thermal pad material

ρ_{TP} Density of the thermal pad material

Cold plate

A_{BCPmx} Maximum cold plate area for the validity of the cold plate meta-model

$Cost_{CPC0}$ Cold plate offset meta-parameter (constant regression coefficient) for cold plate cost estimation

k_{CPC0} Cold plate proportionality meta-parameter (proportionality regression coefficient) for cold plate cost estimation

k_{CPCV} Cold plate meta-parameter (proportionality volume exponent regression coefficient) for cold plate cost estimation

k_{CPCR} Cold plate meta-parameter (proportionality thermal resistance exponent regression coefficient) for cold plate cost estimation

k_{CPCFR} Cold plate meta-parameter (proportionality water flow rate exponent regression coefficient) for cold plate cost estimation

$k_{CPFR0MN}$ Cold plate proportionality meta-parameter (proportionality regression coefficient) for cold plate minimum water flow rate

$k_{CPFR0MX}$ Cold plate proportionality meta-parameter (proportionality regression coefficient) for cold plate maximum water flow rate

k_{CPFRt} Cold plate meta-parameter (proportionality thickness exponent regression coefficient) for cold plate minimum/maximum water flow rate

k_{CPM0} Cold plate proportionality meta-parameter (proportionality regression coefficient) for cold plate weight estimation

k_{CPMV} Cold plate meta-parameter (proportionality volume exponent regression coefficient) for cold plate weight estimation

k_{CPR0}	Cold plate proportionality meta-parameter (proportionality regression coefficient) for cold plate thermal resistance estimation
k_{CPRA}	Cold plate meta-parameter (proportionality base area exponent regression coefficient) for cold plate thermal resistance estimation
k_{CPRt}	Cold plate meta-parameter (proportionality thickness exponent regression coefficient) for cold plate thermal resistance estimation
k_{CPRFR}	Cold plate meta-parameter (proportionality water flow rate exponent regression coefficient) for cold plate thermal resistance estimation
α_{ACP}	Scale factor for meta-model linear extrapolation

Others

$\mathrm{N_x}$	Number of values per free design parameter to be evaluated within the implement design algorithm (design space resolution)
ΔX_{HS}	Delta space for module/heatsink volume evaluation

2.5 Other indirect design inputs, variables, and parameters

$I_{cellMX.C}$	Maximum cell charge current
$I_{cellMX.D}$	Maximum cell discharge current
k_{ocf}	Overcurrent factor (safety margin on maximum module current)
kR_{EOL}	Expected increment ratio of cell resistance at EOL criteria respect to BOL
$\mathrm{Q_{Cell}}$	Cell heat
Q_{cellMX}	Maximum cell heat for worst operating conditions at EOL nominal operation
$R_{cellMX.C}$	Maximum cell charge resistance
$R_{cellMX.D}$	Maximum cell discharge resistance
T_{cell}	Average cell temperature
T_{cellOp}	Maximum cell operating temperature
T_{cellMX}	Maximum allowed cell temperature under worst operating conditions
T_{wMX}	Maximum inlet/outlet water temperature

3 Water cooling system based on thermal conductive fins and cold plates.

3.1 Thermal management

Battery cell thermal management is based on water cooling. The considered concept of battery module with cooling system based on thermal conductive fins and cold plates is illustrated in Figure 3-1. This concept is applicable for battery cells in prismatic and pouch formats/shapes.

The water-cooling system is designed to dissipate the cell heat ($\mathrm{Q_{cell}}$) while keeping the average cell temperature (T_{cell}) bellow a given maximum operating cell temperature (T_{cellOp}) (always minor or equal to the maximum allowed cell temperature (T_{cellMX})) under worst operating conditions).
The average cell heat is assumed to be driven by the cell power loss, so the maximum cell heat for worst operating conditions will be given at EOL nominal operation:

$$Q_{cellMX} = kR_{EOL} \cdot {k_{ocf}}^2 \cdot \max\{R_{cellMX.C} \cdot (I_{cellMX.C})^2;\ R_{cellMX.D} \cdot (I_{cellMX.D})^2\}$$

where $R_{cellMX.C}$ and $R_{cellMX.D}$ are the maximum charge and discharge cell resistance at BOL, respectively, kR_{EOL} is the expected increment ratio of cell resistance at EOL criteria respect to BOL (typically 1.3~2), $I_{cellMX.C}$ and $I_{cellMX.D}$ are the maximum charge and discharge cell current, and k_{ocf} is the overcurrent factor (safety margin on maximum module current, which also can be used as safety margin accounting for dynamic thermal cycling associated to short term power pulses). Then, the required (limit) heatsink thermal resistance per cell ($R_{thHSMax}$) can be estimated for the desired maximum operating cell temperature and the maximum inlet/outlet water temperature (T_{wMX}), as follows:

$$R_{thHS} \leq R_{thHSMax} = \frac{T_{cellOp} - T_{wMX}}{Q_{cellMX}}$$

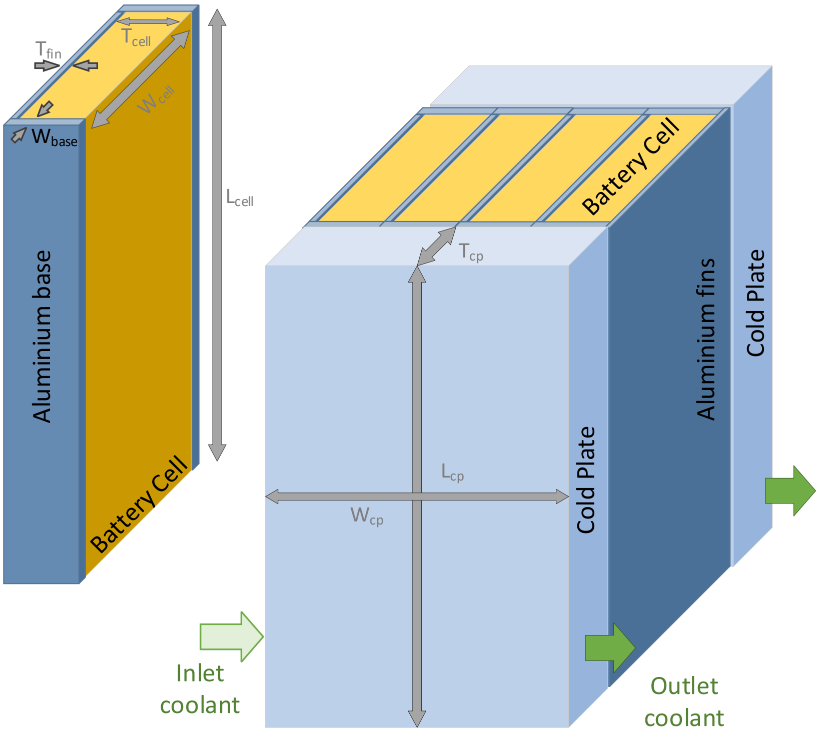

Figure 3-1 Battery module with cooling system based on thermal conductive fins and cold plates.

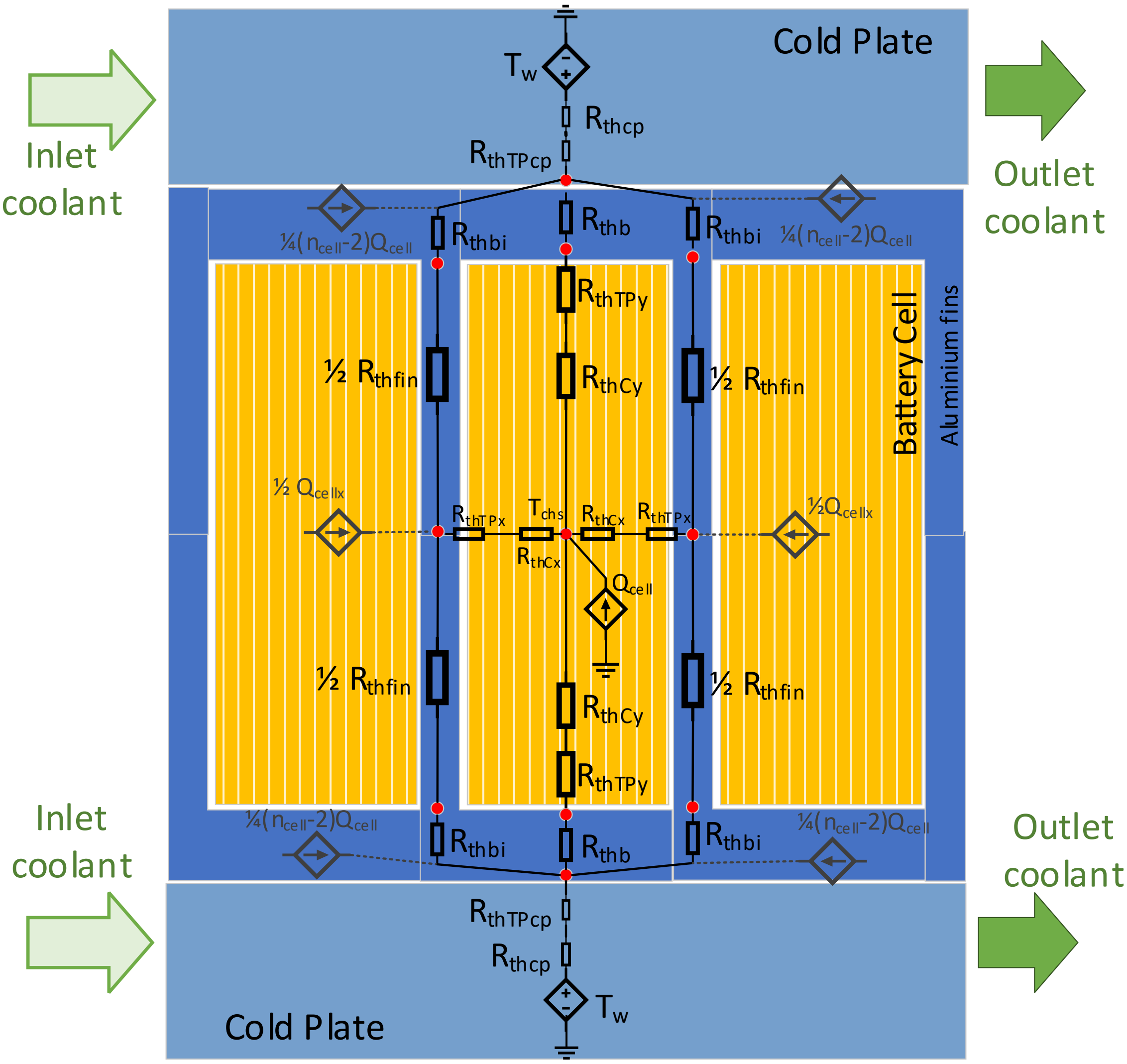

Figure 3-2 Simplified thermal model for water cooling system based on conductive thermal fins and cold plates.

3.2 Thermal resistance

This cooling system is composed by three main components: the thermal Conductive fins (to conduct the cell heat to the cold plates), thermal pads (interfacing the battery cells and thermal fins and cold plates), and the cold plates. Figure 3-2 shows a simplified thermal model for the water-cooling system based on conductive thermal fins and cold plates. The average thermal resistance per cell of the fin-cooling system ($R_{thHS.FIN}$) can be estimated as follows:

$$R_{thHS.FIN} = \frac{\left(R_{thCellx} + R_{thTPx} + \frac{R_{thFin}}{2} + R_{thBi}\right) \cdot \left(R_{thCelly} + R_{thTPy} + R_{thB}\right)}{R_{thCellx} + R_{thTPx} + \frac{R_{thFin}}{2} + R_{thBi} + 2 \cdot \left(R_{thCelly} + R_{thTPy} + R_{thB}\right)} + \frac{R_{thTPCPi}}{2}$$

where $R_{thCellx}$ and $R_{thCelly}$ are the in-plane and trough-plane cell thermal resistance (from centre of the cell to the cell surface), respectively, R_{thFin} is the fin thermal resistance, R_{thB} is the base thermal resistance, R_{thBi} is the fin-base thermal resistance (equivalent for the fin path), $R_{thTPx/y}$ are the thermal pad thermal resistances (between cell and fin/base) and $R_{thTPCPi}$ is the portion of cold plate thermal resistance (R_{thCP}) and thermal pad (between base and cold plate) thermal resistance (R_{thTPcp}) per cell assuming uniform heat distribution along the cold plate surface ($R_{thTPCPi} = n_{cell} \cdot (R_{thCP} + R_{thTPcp})$).

The cell thermal resistances are estimated based on the cell dimensions W_{cell} (width), L_{cell} (length), t_{cell} (thickness) and the equivalent/average in-plane and trough-plane cell thermal conductivities (κ_{Cellx} and κ_{Celly}), by

$$R_{thCellx} = \frac{t_{cell}}{2 \cdot \kappa_{Cellx} \cdot L_{cell} \cdot W_{cell}}$$

$$R_{thCelly} = \frac{W_{cell}}{2 \cdot \kappa_{Celly} \cdot L_{cell} \cdot t_{cell}}$$

The fin thermal resistance (R_{thFin}) is calculated based on the fin thickness (t_{Fin}) and the thermal conductivity of the fin material (typically Al but Cu could be considered to get better thermal resistance to volume trade-off):

$$R_{thFin} = \frac{L_{cell}}{\kappa_{Fin} \cdot W_{cell} \cdot t_{Fin}}$$

The base thermal resistances are calculated as follows:

$$R_{thBi} = \frac{t_{base}}{\kappa_{Fin} \cdot L_{cell} \cdot (t_{Fin} + 2 \cdot t_{base} \cdot \tan(\alpha_{Fin}))}$$

$$R_{thB} = \frac{t_{base}}{\kappa_{Fin} \cdot L_{cell} \cdot (t_{Fin} + t_{cell})}$$

where t_{base} is the fin base thickness and α_{Fin} is the thermal spreading angle for the thermal conductor material [1].

The pad thermal resistances (R_{thTPx} , R_{thTPy} and R_{thTPcp}) are calculated based on the thermal pad thickness (t_{TP}) and its thermal conductivity (κ_{TP}):

$$R_{thTPx} = \frac{t_{TP}}{\kappa_{TP} \cdot W_{cell} \cdot L_{cell}}$$

$$R_{thTPy} = \frac{t_{TP}}{\kappa_{TP} \cdot t_{cell} \cdot L_{cell}}$$

$$R_{thTPcp} = \frac{t_{TP}}{\kappa_{TP} \cdot W_{CP} \cdot L_{CP}}$$

The thermal resistance of a cold plate can be estimated by:

$$R_{thCP}(A_{BCP}, t_{CP}, V_{FR}) = \frac{K_{CPR0}}{A_{BCP}^{k_{CPRA}} \cdot t_{CP}^{k_{CPRt}} \cdot V_{FR}^{k_{CPRFR}}}$$

where A_{BCP} is the cold plate base surface, t_{CP} is the cold plate thickness, V_{FR} is the inlet water flow rate (dm3/min), and K_{CPR0}, K_{CPRA}, K_{CPRt} and K_{CPRFR} are proportionality regression coefficients (meta-parameters) found by taking data from the different cold plate technologies (see Table 2). Then, the cold plate thermal resistance per cell ($R_{thTPCPi}$) can be calculated as follows:

$$R_{thTPCPi} = n_{cell} \cdot \left(R_{thCP}(W_{CP} \cdot L_{CP}, t_{CP}, V_{FR}) + R_{thTPcp}\right)$$

$$W_{CP} = n_{Cell} \cdot (t_{cell} + 2 \cdot t_{TP}) + (n_{Cell} + 1) \cdot t_{Fin}$$

$$L_{CP} = L_{Cell}$$

where $(W_{CP} \cdot L_{CP})$ is the total base area of the top and bottom cold plates.

3.3 Battery module volume

The battery module volume (heatsink including the battery cells) can be estimated as follows:

$$Vol_{BM} = (W_{CP} + 2 \cdot \Delta X_{HS}) \cdot (L_{CP} + 2 \cdot \Delta X_{HS}) \cdot (W_{Cell} + 2 \cdot t_{Base} + 2 \cdot t_{TP} + 2 \cdot t_{CP} + 2 \cdot \Delta X_{HS})$$

$$n_{Cell} = n_{sCell} \cdot n_{pCell}$$

where ΔX_{HS} is a delta space for module/heatsink volume evaluation accounting cold plate supports, insulation, terminals along others. A module configuration with one row of n_{Cell} battery cells is assumed to simplify the heatsink design evaluation, however, alternatively the number of rows can be vary placing double side cold plates between cell rows in case maximum module dimension constraint need to be considered.

3.4 Heatsink weight

The heatsink total weight can be calculated as the sum of the main weight components:

$$Mass_{HS} = Mass_{Fin} + Mass_{TP} + Mass_{CP}$$

$$Mass_{Fin} = \rho_{Fin} \cdot (W_{Cell} \cdot t_{Fin} \cdot L_{Cell} \cdot (n_{Cell} + 1) + 2 \cdot W_{CP} \cdot L_{CP} \cdot t_{Base})$$

$$Mass_{TP} = 2 \cdot \rho_{TP} \cdot t_{TP} \cdot (n_{Cell} \cdot L_{Cell} \cdot (W_{Cell} + t_{Cell}) + W_{CP} \cdot L_{CP})$$

$$Mass_{CP} = 2 \cdot k_{CPM0} \cdot (W_{CP} \cdot L_{CP} \cdot t_{CP})^{k_{CPMV}}$$

where $Mass_{Fin}$ is the total weight of the thermal conductive fin and base, $Mass_{TP}$ is the total weight of the thermal pad material between cell and fin/base, $Mass_{CP}$ is the cold plates weight, ρ_{Fin}, ρ_{TP}, are the densities of the thermal conductive fin material (Al or Cu) and thermal pad material, respectively, and K_{CPM0}, K_{CPMV} are proportionality regression coefficients found by taking data from the different cold plate technologies (see Table 2).

3.5 Heatsink cost

The heatsink total cost can be estimated as the sum of cost for the main components:

$$Cost_{HS} = Cost_{TP} + Cost_{Fin} + Cost_{CP}$$

$$Cost_{TP} = Cost_{TP1} \cdot Mass_{TP}$$

$$Cost_{Fin} = Cost_{Fin0} \cdot (n_{Cell} + 1) + Cost_{Fin1} \cdot Mass_{Fin}$$

$$Cost_{CP} = 2 \cdot \left(Cost_{CPC0} + k_{CPC0} \cdot (W_{CP} \cdot L_{CP} \cdot t_{CP})^{k_{CPCV}} \cdot R_{thCP}^{k_{CPCR}} \cdot V_{FR}^{k_{CPCFR}}\right)$$

where $Cost_{TP1}$ is the cost per kg for the thermal pad material, $Cost_{Fin1}$ is the cost per kg of the fin material (Al or Cu), $Cost_{Fin0}$ is the unit cost per fin, and $Cost_{CPC0}, k_{CPC0}, k_{CPCV}, k_{CPCR}, K_{CPCFR}$ are proportionality regression coefficients found by taking data from the different cold plate technologies (see Table 2).

4 Heatsink Design approach and implementation

4.1 Design approach

The design approach is based on brute-force pareto-frontier optimization, where a predefined design space (with a given resolution) is explored to bring up a set of design alternatives which represent the best trade-off between the different performance indices (cost, thermal resistance, volume, weight).

The considered parameters for the fin cooling design are shown in Table 1. Two fin materials are considered, aluminium and copper. To simplify the heat sink design, the thermal pad thickness is considered as design constant associated with the selected thermal pad material.

The cold plate performance is modelled based on the meta-model shown in Table 2, which has been fitted considering 60 commercial cold plates from different manufactures. The estimated costs are based on minimum 10.000 units quantity purchase. For large cold plates areas (beyond the maximum cold plate area (A_{BCPmx}) for the validity of the cold plate meta-model), the cold plate meta-model is linearly scaled according to the factor $\alpha_{ACP} = A_{BCP}/A_{BCPmx}$, so the thermal resistance, volume, weight and cost are calculated for a cold plate with base area $\alpha_{ACP} \cdot A_{BCPmx}$ and the calculation scaled according α_{ACP} value.

It can be noted that for a given set of thermal element technologies (the thermal conductive fin, and the cold plate), there are mainly four free design parameters for this heatsink configuration to achieve the desired/required thermal resistance: t_{Fin}, t_{base}, t_{CP} and V_{FR}.

The base thickness can be estimated so the heat is evenly spread on the cold plate surface:

$$t_{base} = \frac{t_{cell}}{4 \cdot \tan(\alpha_{Fin})}$$

Therefore, the design space can be reduced to three free design parameters $\{t_{Fin}, t_{CP}, V_{FR}\}$. The fin thickness is swept considering a minimum fin thickness of 0.5 mm and a maximum fin thickness equal to

Table 1 Fin cooling parameters

	Parameter	Value
Thermal Pad Reference material: H48-6 / TG-AH486 @ T-Global Technology	Thermal Conductivity (κ_{TP})	3.4 [W/mK]
	Thickness (t_{TP})	0.3 [mm]
	Density (ρ_{TP})	2420 [kg/m3]
	Cost density ($Cost_{TP1}$)	60 [EUR/kg]
Thermal conductive fin Reference material: • 6063 aluminium alloy • Cooper	Thermal Conductivity (κ_{Fin})	Al: 210 [W/mK] Cu: 386 [W/mK]
	Thermal spreading angle (α_{Fin}):	Al: 40° Cu: 45°
	Density (ρ_{Fin})	Al: 2690 [kg/m3] Cu: 8940 [kg/m3]
	Cost density ($Cost_{FIN1}$)	Al: 3.2 [EUR/kg] [2] Cu: 6 [EUR/kg]
	Cost per unit ($Cost_{FIN0}$)	0.1 [EUR] [2]
	Minimum Thickness	0.5 [mm]
	Maximum Thickness	0.5*t_{Cell}

Table 2 Cold Plate Meta-model and meta-parameters

Reference Technologies	
Manufacturer	***Series***
Solid State Cooling Systems	LC, LCW, LFLC and HFLC
Wakefield-vette	1204xx, 1209xx, and 180-xx
Aavid, Thermal Division of Boyd Corporation	Hi-Contact
Advanced Thermal Solutions Inc.	ATS-CP and ATS-TCP
Ohmite	CP

Meta-Models			
	Input [units]	***Meta-parameter***	***value***
Thermal Resistance [°C/W]: $R_{thCP} = \frac{K_{CPR0}}{A_{BCP}^{k_{CPRA}} \cdot t_{CP}^{k_{CPRt}} \cdot V_{FR}^{k_{CPRFR}}}$	A_{BCP}: Base Area [m2] t_{CP}:Plate thickness [m] V_{FR}: Flow Rate [dm3/min]	k_{CPR0} k_{CPRA} k_{CPRt} k_{CPRFR}	1.9683e-6 0.5364 1.8718 0.4336
Cold plate Volume [m3] $Vol_{CP} = A_{BCP} \cdot t_{CP}$	A_{BCP}: Base Area [m2] t_{CP}:Plate thickness [m]	--	--
Cold plate Weight [kg] $Mass_{CP} = k_{CPM0} \cdot Vol_{CP}^{k_{CPMV}}$	Vol_{CP}: Volume [m3]	k_{CPM0} k_{CPMV}	15.0475 0.3263
Cold Plate Price [EUR] $Cost_{CP} = Cost_{CPC0} + \frac{k_{CPC0} \cdot Vol_{CP}^{k_{CPCV}} \cdot V_{FR}^{k_{CPCFR}}}{R_{thCP}^{k_{CPCR}}}$	Vol_{CP}: Volume [m3] R_{thCP}[°C/W] V_{FR}: Flow Rate [dm3/min]	$Cost_{CPC0}$ k_{CPC0} k_{CPCV} k_{CPCR} k_{CPCFR}	5 8.0635 0.1766 0.4836 0.4722
Min.-Max. Water Flow Rate $V_{FR.Min} = k_{CPFR0MN} \cdot t_{CP}^{k_{CPFRt}}$ $V_{FR.Max} = k_{CPFR0MX} \cdot t_{CP}^{k_{CPFRt}}$	t_{CP}:Plate thickness [m]	$k_{CPFR0MN}$ $k_{CPFR0MX}$ k_{CPFRt}	3.7208e4 4.4649e5 2.4086

Constraints		
Input variable	***Minimum***	***Maximum***
t_{CP}: Cold plate thickness	10 mm	35 mm
A_{BCP}:Cold plate base area	10 cm2	3000 cm2
V_{FR}: Water flow rate	$V_{FR.Min}$	$V_{FR.Max}$

half the cell thickness. On the other hand, t_{CP}, and V_{FR} are swept within the constrained range reported in Table 2.

The battery cell properties are considered as design constants from the point of view of the heatsink design perspective. Also, it should be noted that the number of cells per module and the required heatsink thermal resistance per cell are the main inputs for the heatsink design, as for a given number of cells per module, the different combinations of maximum cell heat, maximum desired operating cell temperature and maximum inlet/outlet water temperature that result in the same required heatsink thermal resistance per cell will have the same heatsink design.

Figure 4-1 shows the proposed heatsink design algorithm, which can be considered within the battery module design algorithm. So, a core cell is preselected within the general battery module design approach, then the number of cells and required thermal resistance per cell is estimated based on battery module specifications and operating conditions, which are given as inputs for the heatsink design algorithm, together

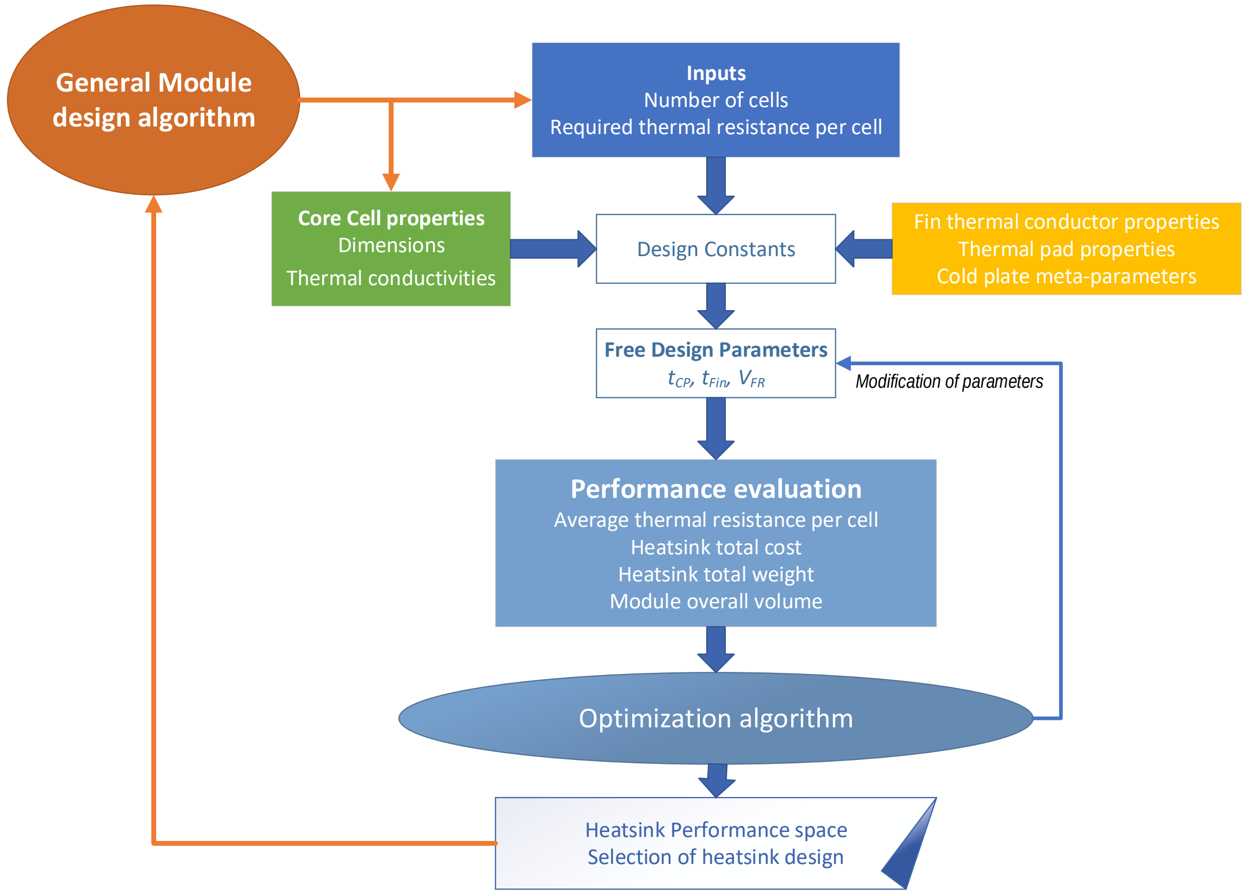
General Module design algorithm
Inputs
Number of cells
Required thermal resistance per cell
Core Cell properties
Dimensions
Thermal conductivities
Design Constants
Fin thermal conductor properties
Thermal pad properties
Cold plate meta-parameters
Free Design Parameters
t_{CP}, t_{Fin}, V_{FR}
Modification of parameters
Performance evaluation
Average thermal resistance per cell
Heatsink total cost
Heatsink total weight
Module overall volume
Optimization algorithm
Heatsink Performance space
Selection of heatsink design

Figure 4-1 Battery module heatsink design algorithm

with the relevant core cell properties. The heatsink design space is explored for a given set of design constants (fin thermal conductor, thermal pad, and cold plate properties/technologies). The optimization algorithm is simply a brute-force approach, where the heatsink performance space is mapped from a design subspace defined by a given grid resolution. Then, the trade-offs between the different performance indices can be analysed and a heatsink design is selected.

Table 3 shows the relevant cell properties for heatsink design for the considered core cell within WP3. Figure 4-2 shows a comparison of the required thermal resistance for maximum continuous operation (charge or discharge) at 50% SOC for the core cell in Table 3.

Table 3 Core Cell Properties for heatsink design

Cell reference*:		NMC 94Ah @Samsung SDI	LTO 23Ah @Toshiba SCiB	LiCap 2300F @Ultimo	LFP 302Ah @ CATL
Dimensions	Width (W_{cell})	173 [mm]	115 [mm]	150 [mm]	173.9 [mm]
	Length (L_{cell})	125 [mm]	103 [mm]	93 [mm]	204.6 [mm]
	Thickness (t_{cell})	45 [mm]	22 [mm]	15.5 [mm]	71.6 [mm]
Thermal	In-plane thermal conductivity (κ_{Cellx})	30 [W/mK]	31 [W/mK]	-- [W/mK]	-- [W/mK]
	Trough-plane thermal conductivity (κ_{Celly})	1.7 [W/mK]	0.8 [W/mK]	-- [W/mK]	-- [W/mK]
	In-plane thermal resistance ($R_{thCellx}$)	0.612 [°C/W]	1.1608 [°C/W]	--[°C/W]	--[°C/W]
	Trough-plane thermal resistance ($R_{thCelly}$)	0.5126 [°C/W]	0.8186 [°C/W]	-- [°C/W]	-- [°C/W]
Electrical	Maximum Continuous charge/discharge current	72 / 150 [A]	92 / 92 [A]	350 / 350 [A]	604 / 906 [A]
	Charge/discharge cell resistance at 50%SOC	0.79 [mΩ]	1.17 [mΩ]	0.72 [mΩ]	0.45 [mΩ]
	Charge/discharge cell resistance at 20%SOC	0.89 [mΩ]	1.25 [mΩ]	0.8 [mΩ]	0.5 [mΩ]
	Expected increment ratio of cell resistance (kR_{EOL})	1.276	1.94	1	--

*All cells have prismatic shape.

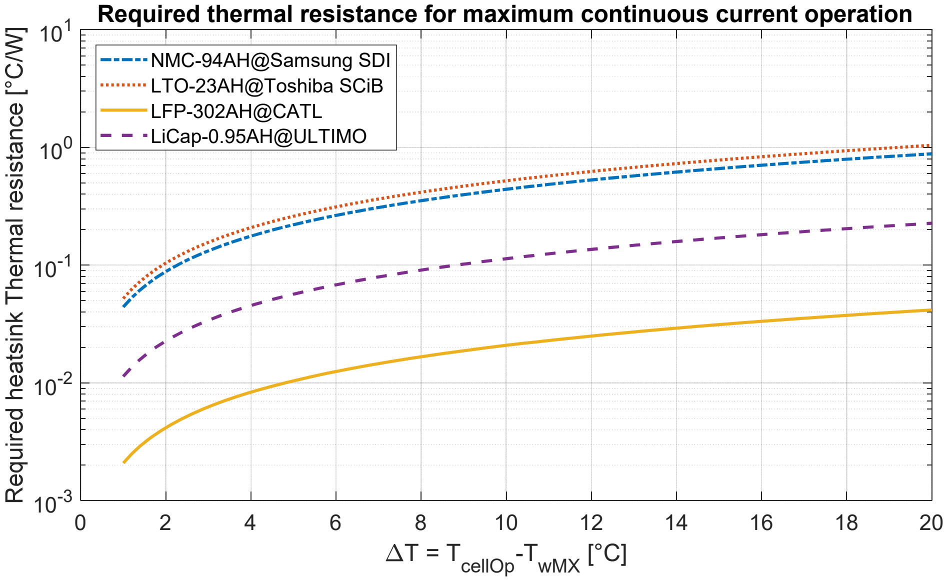

Figure 4-2 Comparison of the required thermal resistance for maximum continuous operation at 50% SOC.

4.2 Heatsink design implementation

The heatsink design algorithm described in previous section has been implemented in Matlab with the function Eval_DesignHSFin, which is presented in appendix A.1 and the inputs/outputs definitions are as follows:

[**HeatSink, DesignSpace**] = *Eval_DesignHSFin* (**CoreCell, RthReq, ncell, ParaHSfin**)

Inputs:

CoreCell: Struct with the main cell properties. Fields:

CoreCell.W → W_{cell} - Cell width in [m]

CoreCell.L → L_{cell} - Cell length in [m]

CoreCell.T → t_{cell} - Cell thickness in [m]

CoreCell.Rcx →$R_{thCellx}$ - In-plane cell thermal resistance (from centre of the cell to the cell surface $W_{cell} x L_{cell}$) in [°C/W]

CoreCell.Rcy →$R_{thCelly}$ - Trough-plane cell thermal resistance (from centre of the cell to the cell surface $W_{cell} x t_{cell}$) in [°C/W]

RthReq: Required heatsink thermal resistance per cell in [°C/W]

ncell: Number of cells

ParaHSfin: Struct with the design constants (fin thermal conductor, thermal pad, and cold plate properties/technologies) and general parameters. Fields:

ParaHSfin.k_TP → κ_{TP} - Thermal conductivity of the thermal pad material in [W/mK]

ParaHSfin.T_TP → t_{TP} - Thermal pad thickness in [m]

ParaHSfin.p_TP → ρ_{TP} - Density of the thermal pad material in [kg/m3]

ParaHSfin. CostW_TP → $Cost_{TP1}$ - Cost per kg for the thermal pad material in [EUR/kg]

ParaHSfin.kAL → κ_{Fin}- Thermal conductivity of the aluminium fin in [W/mK]

ParaHSfin.pAL → ρ_{Fin} - Density of the aluminium conductive fin in [kg/m3]

ParaHSfin.alphaAL → α_{Fin} - Thermal spreading angle for the aluminium fin in [rad]

ParaHSfin. CostW_AL → $Cost_{Fin1}$ - Cost per kg of the aluminium fin in [EUR/kg]

ParaHSfin. CostAL0 → $Cost_{Fin0}$ - Unit cost per aluminium fin in [EUR]

ParaHSfin.kCu → κ_{Fin}- Thermal conductivity of the copper fin in [W/mK]

ParaHSfin.pCu → ρ_{Fin} - Density of the copper conductive fin in [kg/m3]

ParaHSfin.alphaCu → α_{Fin} - Thermal spreading angle for the copper fin in [rad]

ParaHSfin. CostW_Cu → $Cost_{Fin1}$ - Cost per kg of the copper fin in [EUR/kg]

ParaHSfin. CostCu0 → $Cost_{Fin0}$ - Unit cost per copper fin in [EUR]

ParaHSfin. TfinMIN → Minimum fin thickness in [m]

ParaHSfin.DX → ΔX_{HS} - Delta space for module/heatsink volume evaluation in [m]

ParaHSfin. Nx → Number of values per free design parameter to be evaluated (design space resolution)

ParaHSfin.CPModel →Struct with cold plate meta-parameters. Fields (definitions in section 2.4):

kRth0 → k_{CPR0}

kRthAb → k_{CPRA}

kRthTcp →k_{CPRt}

kRthWFR → k_{CPRFR}

kWFR0→k_{CPFR0}

kWFRTcp→k_{CPFRt}

kWFR0mx→$k_{CPFR0MX}$

kWFR0mn→$k_{CPFR0MN}$

Price0 → $Cost_{CPC0}$

kP0 → k_{CPC0}
kPVol → k_{CPCV}
kPRth → k_{CPCR}
kPWFR → k_{CPCFR}
kM0 → k_{CPM0}
kMV → k_{CPMV}
kMWFR → k_{CPMFR}
AbMN → A_{BCPmn}
AbMX → A_{BCPmx}
TcpMN → t_{CPmn}
TcpMX → t_{CPmx}

Outputs:
HeatSink: struct with the selected heatsink design based on minimum cost criterion to fulfil the required thermal resistance constraint. Fields:

HeatSink.Type:	'FinCooling-Al' or 'FinCooling-Cu'
HeatSink.Volume.Total:	Heatsink total volume (including cells) in m3
HeatSink.Volume.ColdPlate:	Cold plates (x2) volume in m3
HeatSink.Volume.ThermalPad:	Thermal pad total volume in m3
HeatSink.Volume.Fin	Fin material total volume in m3
HeatSink.Mass.Total	Heatsink total weight in kg
HeatSink.Mass.ColdPlate	Cold plates(x2) total weight in kg
HeatSink.Mass.ThermalPad	Thermal pad total weight in kg
HeatSink.Mass.Fin	Fin material total weight in kg
HeatSink.Dimension.H	Heatsink/module total height in m
HeatSink.Dimension.W	Heatsink/module total width in m
HeatSink.Dimension.L	Heatsink/module total length in m
HeatSink.Cost.Total	Heatsink total cost in EUR
HeatSink.Cost.ColdPlate	Cost of the cold plates (x2) in EUR
HeatSink.Cost.ThermalPad	Thermal pad total cost in EUR
HeatSink.Cost.Fin	Fin units total cost in EUR
HeatSink.Design.Tfin	Estimated Fin thickness in m
HeatSink.Design.Tcp	Estimated cold plate thickness in m
HeatSink.Design.WFRcp	Estimated cold plate water flow rate in dm3/min
HeatSink.Design.Tbase	Estimated base thickness in m
HeatSink.Design.RthFinHS	Estimated heatsink thermal resistance per cell in [°C/W]
HeatSink.Design.RthCP	Estimated thermal resistance per cold plate in [°C/W]

DesignSpace: 1x2 struct array with the evaluated design subspace, performance subspace and main design variables. **DesignSpace**(1) contains results for heatsink design based on Aluminium fin material, and **DesignSpace**(2) contains the results for heatsink design based on Copper fin material. **DesignSpace** struct is same struct type as **HeatSink** struct.

The file HeatsinkParameters.mat contains the struct **ParaHSfin** with the prefilled fields according to Table 1 and Table 2.
The file BatteryCellSpecs.mat contains the 1x4 struct array **CoreCell** with the battery cell specifications presented in Table 3.

The matlab script Example_BatteryHeatsinkDesign.m has been created to show how to use the function Eval_DesignHSFin for the heatsink design of the battery cells reported in Table 3. The script can be found in appendix A.2, and the main results are presented in the next sections.

4.3 Design Examples

The first part of the script Example_BatteryHeatsinkDesign.m shows an example of the heatsink design space for each core cell when 20 cells per module are used. Figure 4-3, Figure 4-4, Figure 4-5 and Figure 4-6 show the heatsink design results with 20 core cell per module for NMC, LTO, LFP and Li-Cap as core cells, respectively. The thermal resistance per cell, heatsink cost, module volume and heatsink weight relationships are plotted for each case and for two fin materials (Al and Cu). In general, Fin cooling based on Al is the cheapest alternative, but fin cooling solutions based on Cu can achieve compact designs.

4.4 Design Trends

The second part of the script Example_BatteryHeatsinkDesign.m shows an example of the obtained heatsink design (aiming minimum cost) as function of the number of cells for each reference core cell. The ratio of heatsink cost to total cell cost is presented in Figure 4-7. In general, for all the considered core cells, the relative cost quickly decreases with the number of cells but beyond a given number of cells (different for each cell technology) the relative heatsink cost becomes almost constant, meaning that for the considered battery heatsink technology, the heatsink cost increases with the same ratio as the total cost of the cells increases.

Finally, Figure 4-8 shows the trend for the module (core cells cost plus heatsink cost) cost, weight and volume per unit energy and power versus number of cells for the four reference core cells and considering minimum cost as selected criteria for the heatsink design for each number of cells and core cell combinations.

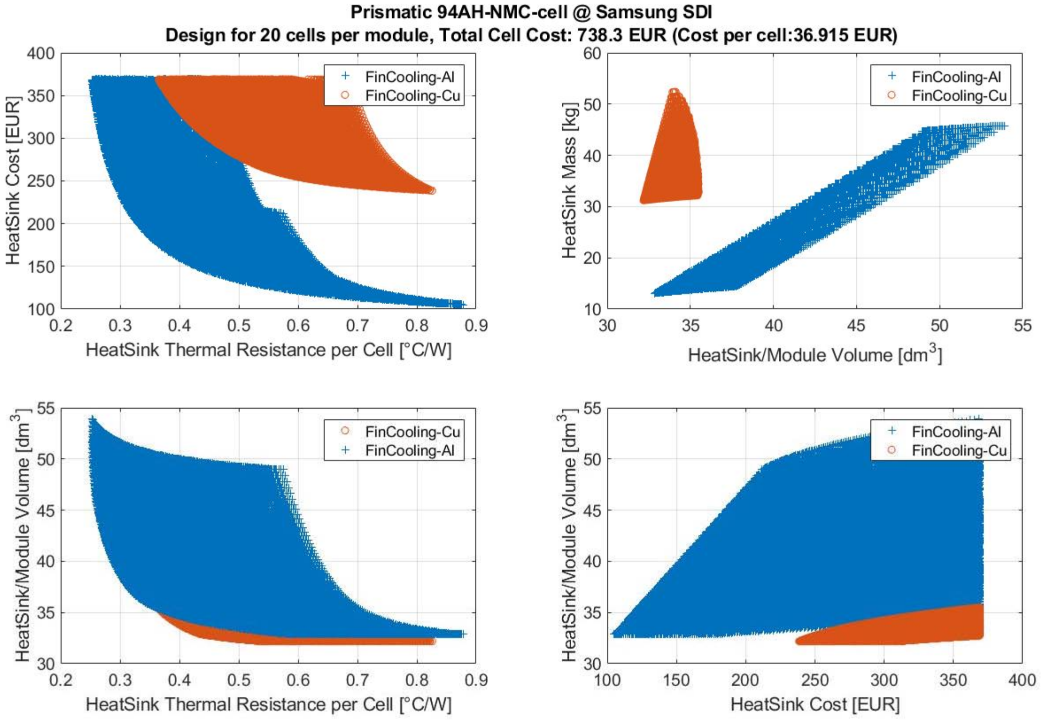

Figure 4-3 HeatSink design results for 20 reference NMC cells per module.

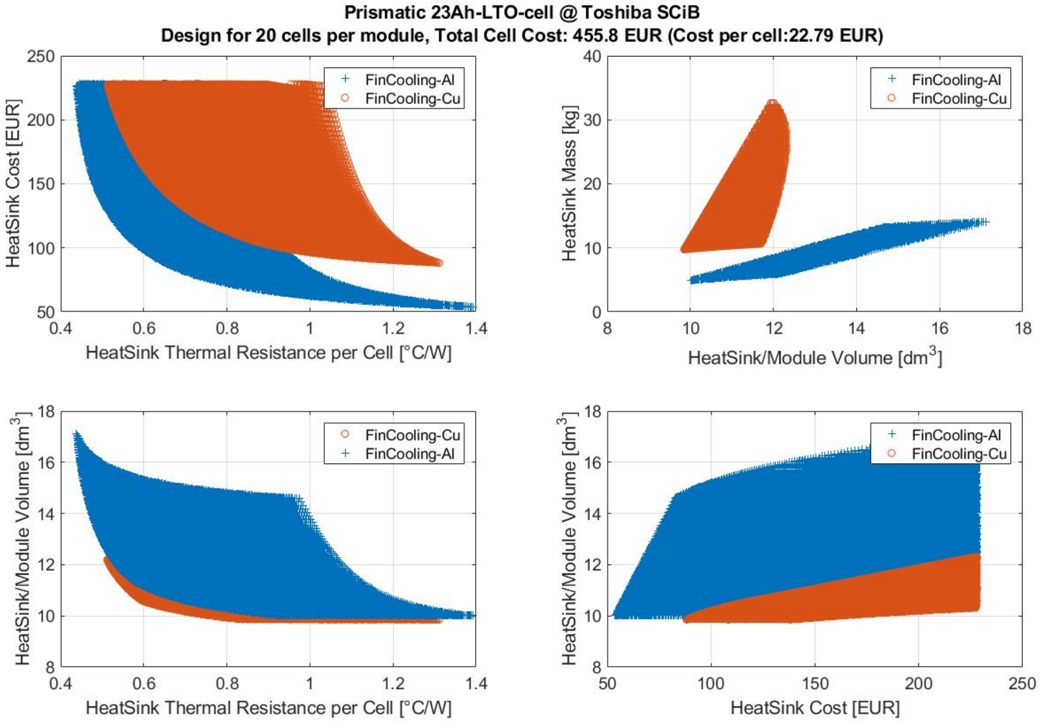

Figure 4-4 HeatSink design results for 20 reference LTO cells per module

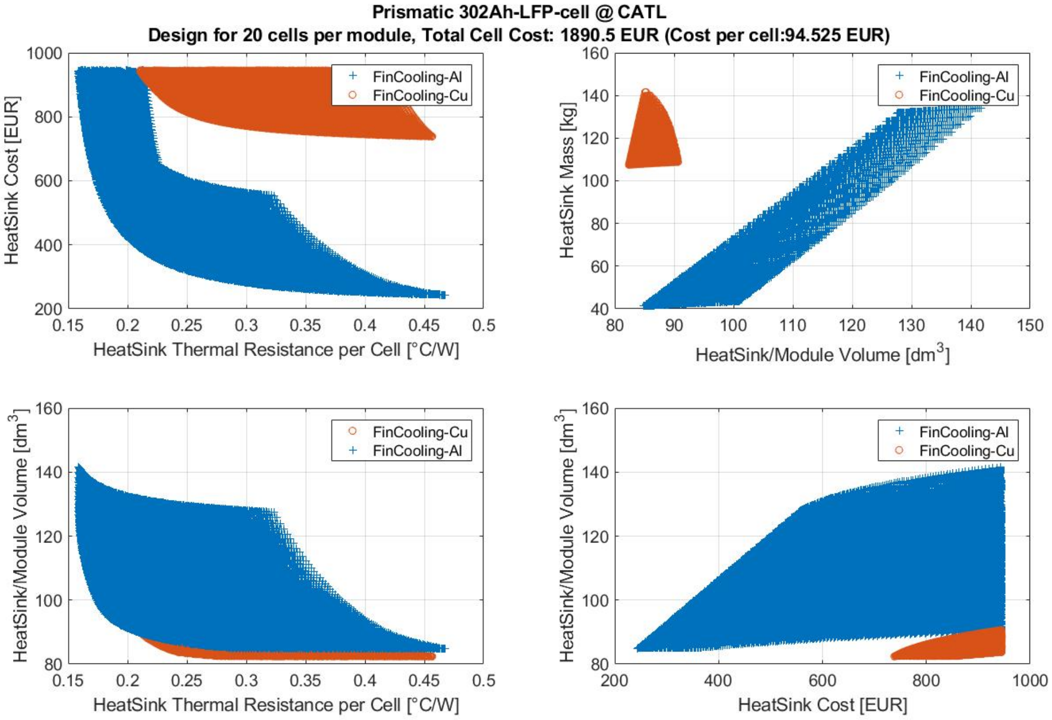

Figure 4-5 HeatSink design results for 20 reference LFP cells per module

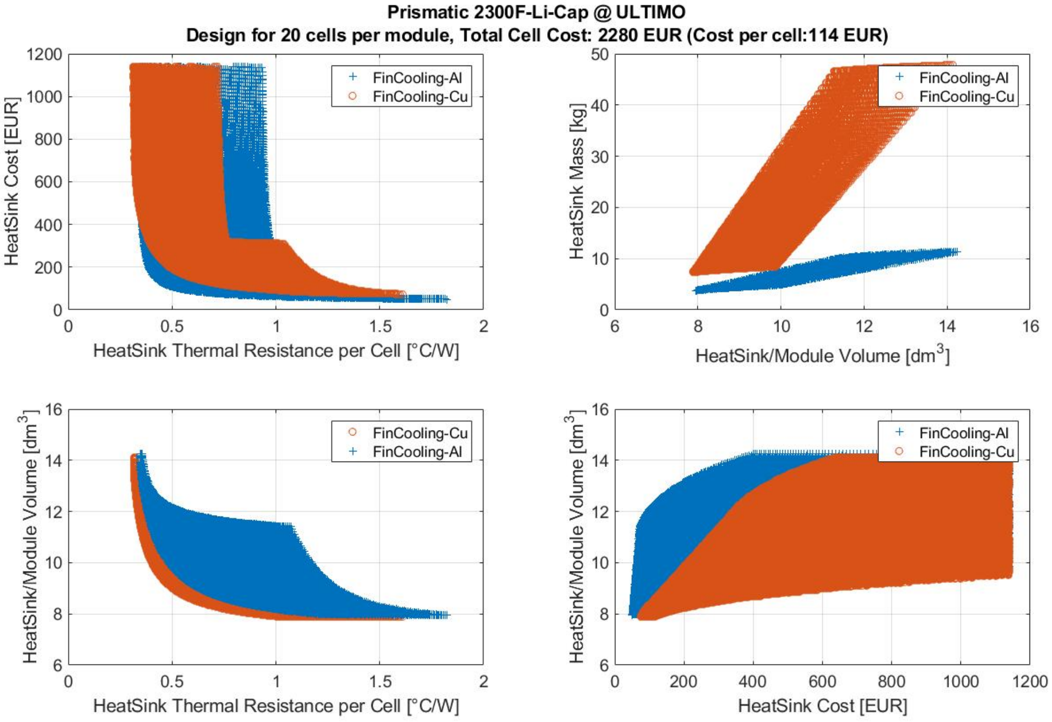

Figure 4-6 HeatSink design results for 20 reference Li-Cap cell per module

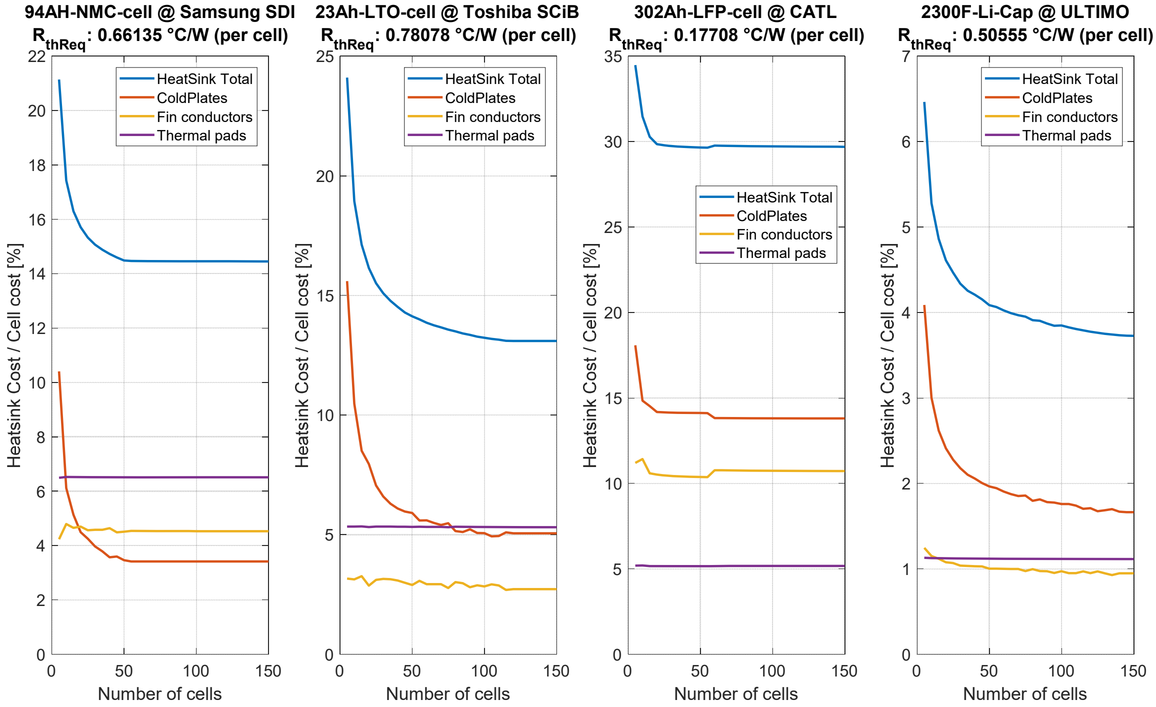

Figure 4-7 Heatsink cost versus number of cells for the reference core cells.

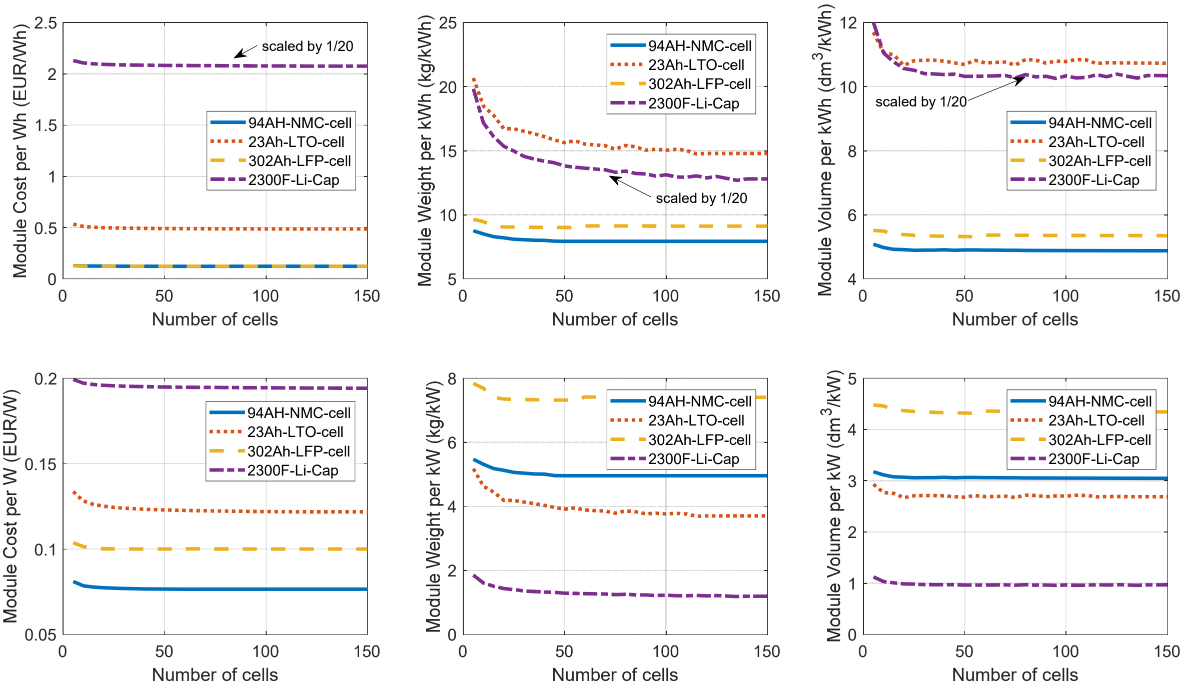

Figure 4-8 Module (core cells cost plus heatsink cost) cost, weight and volume per unit energy and power versus number of cells. Comparison of the trend for the four reference core cells.

A Matlab Scripts

A.1 Eval_DesignHSFin.m

```
function [HeatSink,DesignSpace]=Eval_DesignHSFin(CoreCell,RthReq,ncell,ParaHSfin)
CPModel=ParaHSfin.CPModel;
Nx=ParaHSfin.Nx; %number of values per variable to evaluate
Tfinx=logspace(log10(ParaHSfin.TfinMIN),log10(CoreCell.T/2),Nx);
Tcpx=logspace(log10(CPModel.TcpMN),log10(CPModel.TcpMX),Nx);
[~,~,WFRrange,~,~,~]=eval_CPmodel(CPModel,NaN,CPModel.TcpMX,NaN);
WFRcpx=linspace(WFRrange(3),WFRrange(1),Nx);
[~,WFRcpxx]=ndgrid(Tcpx,WFRcpx);
for kk=1:length(Tcpx)
    [~,~,WFRrange,~,~,~]=eval_CPmodel(CPModel,NaN,Tcpx(kk),NaN);
    WFRcpxx(kk,:)=logspace(log10(WFRrange(3)),log10(WFRrange(1)),Nx); %get the right
range for WFR as function of Tcp
end
[Tcpx,~,Tfinx]=ndgrid(Tcpx,WFRcpx,Tfinx);
WFRcpx=repmat(WFRcpxx,[1,1,Nx]);
Tcpx=Tcpx(:);
Tfinx=Tfinx(:);
WFRcpx=WFRcpx(:);

%% Evalute designs HS-Fin
W_CP=ncell*(CoreCell.T+2*ParaHSfin.T_TP)+(ncell+1)*Tfinx;
L_CP=CoreCell.L;
AbaseCP=W_CP.*L_CP;
[VolCP,RthCP,~,~,CostCP,MassCP]=eval_CPmodel(CPModel,AbaseCP,Tcpx,WFRcpx);
RthTP_BCP=ParaHSfin.T_TP./(ParaHSfin.k_TP.*AbaseCP);
MassTP_BCP=AbaseCP.*ParaHSfin.T_TP.*ParaHSfin.p_TP;

[RthFinHS_Al,Tbase_Al,MassFIN_Al,CostFIN_Al]=Eval_RthFin(CoreCell,Tfinx,ncell*(RthCP+RthTP_BCP),
ncell,ParaHSfin,1);
[RthFinHS_Cu,Tbase_Cu,MassFIN_Cu,CostFIN_Cu]=Eval_RthFin(CoreCell,Tfinx,ncell*(RthCP+RthTP_BCP),
ncell,ParaHSfin,2);

MassTP=(CoreCell.W+CoreCell.T).*CoreCell.L.*ParaHSfin.T_TP.*(2*ncell).*ParaHSfin.p_TP+2*MassTP_BCP;
CostTP=ParaHSfin.CostW_TP.*MassTP;
W_HSFin=(W_CP+2*ParaHSfin.DX);
L_HSFin=(L_CP+2*ParaHSfin.DX);

H_HSFin_Al=(CoreCell.W+2*Tbase_Al+2*ParaHSfin.T_TP+2*Tcpx+2*ParaHSfin.DX);
VolumeTotal_Al=W_HSFin.*L_HSFin.*H_HSFin_Al;
CostTotal_Al=2*CostCP+CostTP+CostFIN_Al;
MassTotal_Al=2*MassCP+MassFIN_Al+MassTP;

H_HSFin_Cu=(CoreCell.W+2*Tbase_Cu+2*Tcpx+2*ParaHSfin.DX);
VolumeTotal_Cu=W_HSFin.*L_HSFin.*H_HSFin_Cu;
CostTotal_Cu=2*CostCP+CostTP+CostFIN_Cu;
MassTotal_Cu=2*MassCP+MassFIN_Cu+MassTP;

op=1;
DesignSpace(op).Type='FinCooling-Al';
DesignSpace(op).Volume.Total=VolumeTotal_Al;
DesignSpace(op).Volume.ColdPlate=2*VolCP;
DesignSpace(op).Volume.ThermalPad=MassTP./ParaHSfin.p_TP;
DesignSpace(op).Volume.Fin=MassFIN_Al./ParaHSfin.pAL;
DesignSpace(op).Mass.Total=MassTotal_Al;
DesignSpace(op).Mass.ColdPlate=2*MassCP;
DesignSpace(op).Mass.ThermalPad=MassTP;
DesignSpace(op).Mass.Fin=MassFIN_Al;
DesignSpace(op).Dimension.H=H_HSFin_Al;
DesignSpace(op).Dimension.W=W_HSFin;
DesignSpace(op).Dimension.L=L_HSFin;
DesignSpace(op).Cost.Total=CostTotal_Al;
DesignSpace(op).Cost.ColdPlate=2*CostCP;
DesignSpace(op).Cost.ThermalPad=CostTP;
DesignSpace(op).Cost.Fin=CostFIN_Al;
DesignSpace(op).Design.Tfin=Tfinx;
DesignSpace(op).Design.Tcp=Tcpx;
DesignSpace(op).Design.WFRcp=WFRcpx;
DesignSpace(op).Design.Tbase=Tbase_Al;
DesignSpace(op).Design.RthFinHS=RthFinHS_Al;
DesignSpace(op).Design.RthCP=RthCP;

op=2;
DesignSpace(op).Type='FinCooling-Cu';
DesignSpace(op).Volume.Total=VolumeTotal_Cu;
DesignSpace(op).Volume.ColdPlate=2*VolCP;
DesignSpace(op).Volume.ThermalPad=MassTP./ParaHSfin.p_TP;
DesignSpace(op).Volume.Fin=MassFIN_Cu./ParaHSfin.pCu;
DesignSpace(op).Mass.Total=MassTotal_Cu;
DesignSpace(op).Mass.ColdPlate=2*MassCP;
DesignSpace(op).Mass.ThermalPad=MassTP;
DesignSpace(op).Mass.Fin=MassFIN_Cu;
DesignSpace(op).Dimension.H=H_HSFin_Cu;
DesignSpace(op).Dimension.W=W_HSFin;
DesignSpace(op).Dimension.L=L_HSFin;
DesignSpace(op).Cost.Total=CostTotal_Cu;
DesignSpace(op).Cost.ColdPlate=2*CostCP;
DesignSpace(op).Cost.ThermalPad=CostTP;
DesignSpace(op).Cost.Fin=CostFIN_Cu;
DesignSpace(op).Design.Tfin=Tfinx;
DesignSpace(op).Design.Tcp=Tcpx;
DesignSpace(op).Design.WFRcp=WFRcpx;
DesignSpace(op).Design.Tbase=Tbase_Cu;
DesignSpace(op).Design.RthFinHS=RthFinHS_Cu;
DesignSpace(op).Design.RthCP=RthCP;

%% Select design
indexAl=find(RthFinHS_Al<=RthReq);
[MinCostAl,idMinCostAl]=min(CostTotal_Al(indexAl));
indexCu=find(RthFinHS_Cu<=RthReq);
[MinCostCu,idMinCostCu]=min(CostTotal_Cu(indexCu));

if ~isempty(MinCostAl) && ~isempty(MinCostCu)
    [~,idOP]=min([MinCostAl,MinCostCu]);
elseif ~isempty(MinCostAl)
    idOP=1;
elseif ~isempty(MinCostCu)
    idOP=2;
else
    idOP=3;
end
switch idOP
    case 1 %Al Fin
        HeatSink.Type='FinCooling-Al';
        HeatSink.Volume.Total=VolumeTotal_Al(indexAl(idMinCostAl));
        HeatSink.Volume.ColdPlate=2*VolCP(indexAl(idMinCostAl));
        HeatSink.Volume.ThermalPad=MassTP(indexAl(idMinCostAl))./ParaHSfin.p_TP;
        HeatSink.Volume.Fin=MassFIN_Al(indexAl(idMinCostAl))./ParaHSfin.pAL;
        HeatSink.Mass.Total=MassTotal_Al(indexAl(idMinCostAl));
        HeatSink.Mass.ColdPlate=2*MassCP(indexAl(idMinCostAl));
        HeatSink.Mass.ThermalPad=MassTP(indexAl(idMinCostAl));
        HeatSink.Mass.Fin=MassFIN_Al(indexAl(idMinCostAl));
        HeatSink.Dimension.H=H_HSFin_Al(indexAl(idMinCostAl));
        HeatSink.Dimension.W=W_HSFin(indexAl(idMinCostAl));
        HeatSink.Dimension.L=L_HSFin;
        HeatSink.Cost.Total=CostTotal_Al(indexAl(idMinCostAl));
        HeatSink.Cost.ColdPlate=2*CostCP(indexAl(idMinCostAl));
        HeatSink.Cost.ThermalPad=CostTP(indexAl(idMinCostAl));
        HeatSink.Cost.Fin=CostFIN_Al(indexAl(idMinCostAl));
        HeatSink.Design.Tfin=Tfinx(indexAl(idMinCostAl));
        HeatSink.Design.Tcp=Tcpx(indexAl(idMinCostAl));
        HeatSink.Design.WFRcp=WFRcpx(indexAl(idMinCostAl));
        HeatSink.Design.Tbase=Tbase_Al;
        HeatSink.Design.RthFinHS=RthFinHS_Al(indexAl(idMinCostAl));
        HeatSink.Design.RthCP=RthCP(indexAl(idMinCostAl));
    case 2 %Cu Fin
        HeatSink.Type='FinCooling-Cu';
        HeatSink.Volume.Total=VolumeTotal_Cu(indexCu(idMinCostCu));
        HeatSink.Volume.ColdPlate=2*VolCP(indexCu(idMinCostCu));
        HeatSink.Volume.ThermalPad=MassTP(indexAl(idMinCostAl))./ParaHSfin.p_TP;
        HeatSink.Volume.Fin=MassFIN_Cu(indexCu(idMinCostCu))./ParaHSfin.pCu;
        HeatSink.Mass.Total=MassTotal_Cu(indexCu(idMinCostCu));
        HeatSink.Mass.ColdPlate=2*MassCP(indexCu(idMinCostCu));
        HeatSink.Mass.ThermalPad=MassTP(indexAl(idMinCostAl));
        HeatSink.Mass.Fin=MassFIN_Cu(indexCu(idMinCostCu));
        HeatSink.Dimension.H=H_HSFin_Cu(indexCu(idMinCostCu));
        HeatSink.Dimension.W=W_HSFin(indexCu(idMinCostCu));
        HeatSink.Dimension.L=L_HSFin;
        HeatSink.Cost.Total=CostTotal_Cu(indexCu(idMinCostCu));
        HeatSink.Cost.ColdPlate=2*CostCP(indexCu(idMinCostCu));
        HeatSink.Cost.ThermalPad=CostTP(indexAl(idMinCostAl));
        HeatSink.Cost.Fin=CostFIN_Cu(indexCu(idMinCostCu));
        HeatSink.Design.Tfin=Tfinx(indexCu(idMinCostCu));
        HeatSink.Design.Tcp=Tcpx(indexCu(idMinCostCu));
        HeatSink.Design.WFRcp=WFRcpx(indexCu(idMinCostCu));
        HeatSink.Design.Tbase=Tbase_Cu;
        HeatSink.Design.RthFinHS=RthFinHS_Cu(indexCu(idMinCostCu));
        HeatSink.Design.RthCP=RthCP(indexCu(idMinCostCu));
    case 3
        HeatSink.Type='No Design Found';
        HeatSink.Volume=DesignSpace(1).Volume([]);
        HeatSink.Mass=DesignSpace(1).Mass([]);
        HeatSink.Dimension=DesignSpace(1).Dimension([]);
        HeatSink.Cost=DesignSpace(1).Cost([]);
        HeatSink.Design=DesignSpace(1).Design([]);

end
```

A.2 Example_BatteryHeatsinkDesign.m

```
% Battery heatsink design examples
clc
clear
close all
%% Load parameters and core cell properties
load HeatsinkParameters ParaHSfin
load BatteryCellSpecs CoreCell
%% 1. Exploring design space for each core cell
ncell=20; %example heatsink design for 20 core cells per module
Rthreq=nan; % No target an especific design but exploring the Rth-Cost-Volume-Mass
performance space
for kk=1:length(CoreCell)
    disp(['Evaluating designs for ',CoreCell(kk).Shape,' ',num2str(CoreCell(kk).Ref),' @
',CoreCell(kk).Manufacturer])
    [HeatSink,DesignSpace]=Eval_DesignHSFin(CoreCell(kk),Rthreq,ncell,ParaHSfin);
    MaxCost=0.5*ncell*CoreCell(kk).Cost; % Set a maximum cost for plotting (Max. 50% of
total cell cost)
    figure(200+kk)
    subplot(2,2,1)

plot(DesignSpace(1).Design.RthFinHS(DesignSpace(1).Cost.Total<=MaxCost),DesignSpace(1).Co
st.Total(DesignSpace(1).Cost.Total<=MaxCost),'+')
    hold on

plot(DesignSpace(2).Design.RthFinHS(DesignSpace(2).Cost.Total<=MaxCost),DesignSpace(2).Co
st.Total(DesignSpace(2).Cost.Total<=MaxCost),'o')
    xlabel('HeatSink Thermal Resistance per Cell [°C/W]')
    ylabel('HeatSink Cost [EUR]')
    grid on
    legend(DesignSpace(1).Type,DesignSpace(2).Type)
    title({[CoreCell(kk).Shape,' ',CoreCell(kk).Ref,' @ ',CoreCell(kk).Manufacturer];
['Design for ',num2str(ncell),' cells per module, Total Cell Cost:
',num2str(ncell*CoreCell(kk).Cost),' EUR (Cost per cell:',num2str(CoreCell(kk).Cost),'
EUR)']})
    subplot(2,2,3)

plot(DesignSpace(2).Design.RthFinHS(DesignSpace(2).Cost.Total<=MaxCost),DesignSpace(2).Vo
lume.Total(DesignSpace(2).Cost.Total<=MaxCost)*1e3,'o')
    hold on

plot(DesignSpace(1).Design.RthFinHS(DesignSpace(1).Cost.Total<=MaxCost),DesignSpace(1).Vo
lume.Total(DesignSpace(1).Cost.Total<=MaxCost)*1e3,'+')
    ylabel('HeatSink/Module Volume [dm^3]')
    xlabel('HeatSink Thermal Resistance per Cell [°C/W]')
    grid on
    legend(DesignSpace(2).Type,DesignSpace(1).Type)
    title({[CoreCell(kk).Shape,' ',num2str(CoreCell(kk).Ref),' @
',CoreCell(kk).Manufacturer]; ['Design for ',num2str(ncell),' cells per module, Total
Cell Cost: ',num2str(ncell*CoreCell(kk).Cost),' EUR (Cost per
cell:',num2str(CoreCell(kk).Cost),' EUR)']})
    subplot(2,2,2)

plot(DesignSpace(1).Volume.Total(DesignSpace(1).Cost.Total<=MaxCost)*1e3,DesignSpace(1).M
ass.Total(DesignSpace(1).Cost.Total<=MaxCost),'+')
    hold on

plot(DesignSpace(2).Volume.Total(DesignSpace(2).Cost.Total<=MaxCost)*1e3,DesignSpace(2).M
ass.Total(DesignSpace(2).Cost.Total<=MaxCost),'o')
    ylabel('HeatSink Mass [kg]')
    xlabel('HeatSink/Module Volume [dm^3]')
    grid on
    legend(DesignSpace(1).Type,DesignSpace(2).Type)
    title({[CoreCell(kk).Shape,' ',CoreCell(kk).Ref,' @ ',CoreCell(kk).Manufacturer];
['Design for ',num2str(ncell),' cells per module, Total Cell Cost:
',num2str(ncell*CoreCell(kk).Cost),' EUR (Cost per cell:',num2str(CoreCell(kk).Cost),'
EUR)']})
    subplot(2,2,4)

plot(DesignSpace(1).Cost.Total(DesignSpace(1).Cost.Total<=MaxCost),DesignSpace(1).Volume.
Total(DesignSpace(1).Cost.Total<=MaxCost)*1e3,'+')
    hold on

plot(DesignSpace(2).Cost.Total(DesignSpace(2).Cost.Total<=MaxCost),DesignSpace(2).Volume.
Total(DesignSpace(2).Cost.Total<=MaxCost)*1e3,'o')
    ylabel('HeatSink/Module Volume [dm^3]')
    xlabel('HeatSink Cost [EUR]')
    grid on
    legend(DesignSpace(1).Type,DesignSpace(2).Type)
    title({[CoreCell(kk).Shape,' ',CoreCell(kk).Ref,' @ ',CoreCell(kk).Manufacturer];
['Design for ',num2str(ncell),' cells per module, Total Cell Cost:
',num2str(ncell*CoreCell(kk).Cost),' EUR (Cost per cell:',num2str(CoreCell(kk).Cost),'
EUR)']})
end
%% 2. HeatSink desing trends
% example of minimum cost heatsink design trend as number of cell increases
TCellmx=35; %Target Maximum cell operating temperature
Twmx=20;    %maximum water temperature
kocf=[1 1 0.42 0.58]; %Overcurrent factor, for Li-cap and LFP cell other kocf used
because too low RthReq for full current design
disp('2. HeatSink desing trends:')
RthReq=zeros(size(CoreCell));
for kk=1:length(CoreCell) %comparison for the core cells (NMC vs LTO vs LFP vs LiCap)
    disp(['Running for ',CoreCell(kk).Shape,' ',num2str(CoreCell(kk).Ref),' @
',CoreCell(kk).Manufacturer])
    QcellD=CoreCell(kk).kR_EOL*CoreCell(kk).Rd.*(kocf(kk)*CoreCell(kk).ICDmx).^2; %max.
Discahrge losses
    QcellC=CoreCell(kk).kR_EOL*CoreCell(kk).Rc.*(kocf(kk)*CoreCell(kk).ICCmx).^2; %max.
cahrge losses

    if TCellmx>CoreCell(kk).TcMxD || TCellmx>CoreCell(kk).TcMxC
        disp('Maximum cell operating temperature major than maximum allowed for the
selected cell!!')
        disp('Proceeding with maximum allowed temperature for the selected cell')
        RthReq(kk)=min((min(CoreCell(kk).TcMxD,TCellmx)-
Twmx)./QcellD,(min(CoreCell(kk).TcMxC,TCellmx)-Twmx)./QcellC);
    else
        RthReq(kk)=min((TCellmx-Twmx)./QcellD,(TCellmx-Twmx)./QcellC);
    end
    disp(['Required RthHS per cell: ',num2str(RthReq(kk)),' °C/W
(',CoreCell(kk).Ref,')'])

    ncell=5:5:150;
    CostHSTotal=zeros(size(ncell));
    CostCP=zeros(size(ncell));
    CostFin=zeros(size(ncell));
    CostTP=zeros(size(ncell));
    MassHSTotal=zeros(size(ncell));
    VolumeHSTotal=zeros(size(ncell));

    for kk2=1:length(ncell)
        [HeatSink,~]=Eval_DesignHSFin(CoreCell(kk),RthReq(kk),ncell(kk2),ParaHSfin);
        if isempty(HeatSink.Volume)
            disp(HeatSink.Type)
        else
            CostHSTotal(kk2)=HeatSink.Cost.Total;
            CostCP(kk2)=HeatSink.Cost.ColdPlate;
            CostFin(kk2)=HeatSink.Cost.Fin;
            CostTP(kk2)=HeatSink.Cost.ThermalPad;
            MassHSTotal(kk2)=HeatSink.Mass.Total;
            VolumeHSTotal(kk2)=HeatSink.Volume.Total;
        end
    end

    figure(2)
    subplot(1,4,kk)
    plot(ncell,CostHSTotal./(ncell*CoreCell(kk).Cost)*100)
    hold on
    plot(ncell,CostCP./(ncell*CoreCell(kk).Cost)*100)
    plot(ncell,CostFin./(ncell*CoreCell(kk).Cost)*100)
    plot(ncell,CostTP./(ncell*CoreCell(kk).Cost)*100)
    ylabel('Heatsink Cost / Cell cost [%]')
    xlabel('Number of cells')
    legend('HeatSink Total','ColdPlates','Fin conductors','Thermal pad material')
    title({[CoreCell(kk).Shape,' ',CoreCell(kk).Ref,' @ ',CoreCell(kk).Manufacturer];
['RthReq: ',num2str(RthReq(kk)),' °C/W (per cell)']})
    grid on

    figure(3)
    plot(ncell.*CoreCell(kk).En*1e-
3,kocf(kk)*ncell.*CoreCell(kk).Vn.*CoreCell(kk).ICDmx*1e-
3,'*','DisplayName',CoreCell(kk).Ref)
    hold on
    xlabel('Module Energy [kWh]')
    ylabel('Module Nominal Cont. Discharge Power [kW]')
    legend('show')


    figure(4)
    subplot(2,3,1)

plot(ncell,(CostHSTotal+ncell*CoreCell(kk).Cost)./(ncell.*CoreCell(kk).En),'DisplayName',
CoreCell(kk).Ref)
    hold on
    xlabel('Number of cells')
    ylabel('Module Cost per Energy EUR/Wh')
    grid on
    legend('show')
    subplot(2,3,2)
    plot(ncell,(MassHSTotal+ncell*CoreCell(kk).Weight)./(ncell.*CoreCell(kk).En*1e-
3),'DisplayName',CoreCell(kk).Ref)
    hold on
    xlabel('Number of cells')
    ylabel('Module Weight per Energy kg/kWh')
    grid on
    subplot(2,3,3)
    plot(ncell,(VolumeHSTotal*1e3)./(ncell.*CoreCell(kk).En*1e-
3),'DisplayName',CoreCell(kk).Ref)
    hold on
    xlabel('Number of cells')
    ylabel('Module Volume per Energy dm^3/kWh')
    grid on
    subplot(2,3,4)

plot(ncell,(CostHSTotal+ncell*CoreCell(kk).Cost)./(kocf(kk)*ncell.*CoreCell(kk).Vn.*CoreC
ell(kk).ICDmx),'DisplayName',CoreCell(kk).Ref)
    hold on
    xlabel('Number of cells')
    ylabel('Module Cost per Power EUR/W')
    grid on
    legend('show')
    subplot(2,3,5)

plot(ncell,(MassHSTotal+ncell*CoreCell(kk).Weight)./(kocf(kk)*ncell.*CoreCell(kk).Vn.*CoreCell(kk).ICDmx*1e-3),'DisplayName',CoreCell(kk).Ref)
    hold on
    xlabel('Number of cells')
    ylabel('Module Weight per Power kg/kW')
    grid on
    subplot(2,3,6)

plot(ncell,(VolumeHSTotal*1e3)./(kocf(kk)*ncell.*CoreCell(kk).Vn.*CoreCell(kk).ICDmx*1e-3),'DisplayName',CoreCell(kk).Ref)
    hold on
    xlabel('Number of cells')
    ylabel('Module Volume per Power dm^3/kW')
    grid on
end
```

SINTEF

Technology for a better society

www.sintef.no