Implementing and validating MOX fuel property models in FINIX

Valtteri Turkki

Introduction

Motivation

- Utilizing the data from Halden MOX experiments
  - The Halden data lake was recently opened
- Developing the MOX capabilities of the KRAKEN framework [1]
  - FINIX is especially designed for coupled calculations
- Improve VTT's competence in MOX fuels and high Pu content fuels
- Get more validation for FINIX
  - Validation against the state-of-the-art FRAPCON-4.0 code [2]

Basics of MOX fuels

- Mixed oxide (MOX) fuels
  - Composed of uranium (~ 95 wt% with enrichment ~1 wt%) and plutonium oxides (~ 5 wt%)
  - Introduced already in the 60s
- Benefits of MOX fuels
  - The fuel pellets can be manufactured from recycled fuel
    - U-238 transmutes to fissile Pu-239
    - Can be manufactured from weapons-grade plutonium
  - The fuel can reach higher burnups, which makes the fuel cycle more efficient

Methods

MOX fuel details

- The mixture nature causes MOX fuels to have specific properties [2]
  - Any mixing process will leave Pu-rich spots of size > 10 µm
  - Fuel homogeneity affects the fuel power distribution
  - The Pu-rich spots evolve through diffusion during irradiation
  - Many models do not capture these microstructural changes
- This leads to differences in thermal, mechanical and fission gas release performance compared to UO_2
  - Especially high burnup behaviour has been studied (also in this work)

The models

- The changes were made for three thermal and one mechanical model
  - Thermal: Fuel thermal conductance λ_{th}, fuel heat capacity c_p and fuel melting point T_{mp}
  - Mechanical: Fuel thermal strain ε_{th}
- The most significant effect is given by the thermal conductance λ_{th}

$$\lambda_{th} = 1.0789\lambda_{th,95}\frac{\rho_{\%}}{1+\frac{1}{2}(1-\rho_{\%})}$$

$$\lambda_{th,95} = \frac{1}{A(x)+B(x)T+h(Bu,T)} + \frac{1.5\cdot 10^{-9}e^{-13520/T}}{T^2}$$ [3]

$$c_p = \frac{K_1\theta^2 e^{\theta/T}}{T^2(e^{\theta/T}-1)^2} + K_2 T + \frac{yK_3E_D}{2RT^2}e^{-E_D/(RT)}$$

$$c_{p\mathrm{MOX}} = \gamma c_{p\mathrm{PuO_2}} + (1-\gamma)c_{p\mathrm{UO_2}}$$ [4]

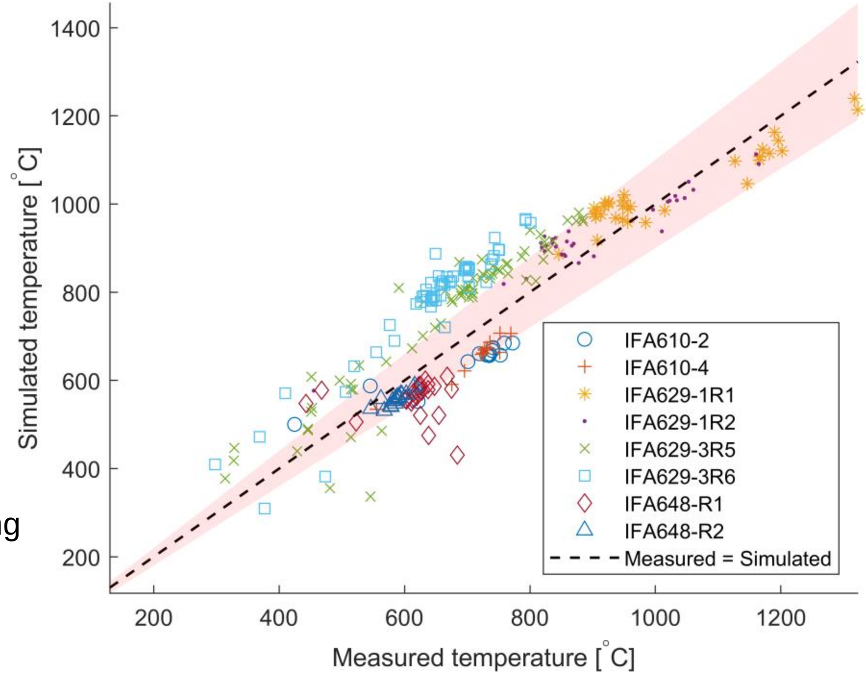
Implementation

- The models were implemented to FINIX source code along with new input options
  - Fuel type, Pu-wt%
- The validation input-files were obtained by converting FRAPCON inputs to FINIX inputs with a custom made Python tool
  - FRAPCON inputs from the integral assessment report [5]
- For further research purposes a version of FINIX that allows inputting model parameters was also implemented

MOX model results

Validating FINIX against Halden data

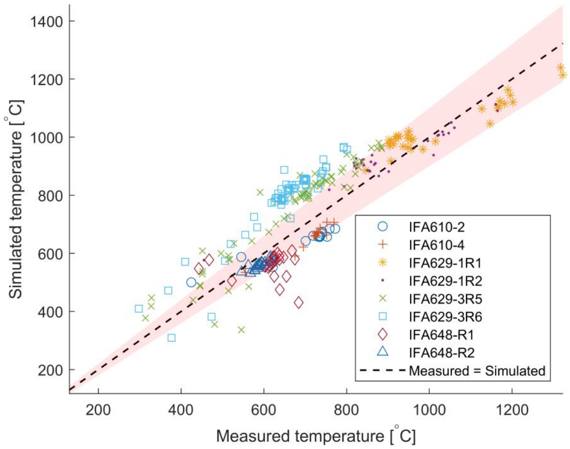
- The MOX models were validated using 8 Halden cases
  - Rods had Pu content of around 5%
  - Starting burnups ranging between 23 and 57 MWd/kg
  - All had been irradiated in a power reactor prior to refabrication
  - Prediction accuracy improvement from old FINIX version around 30%

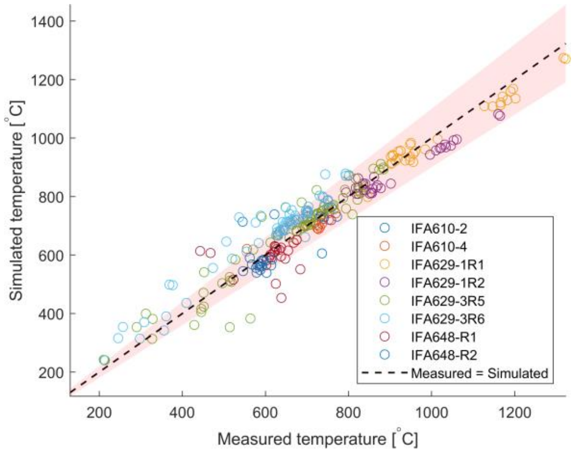


Right: Masterplot showing the FINIX predicted fuel centreline temperatures against the measured ones.

Validation against FRAPCON-4.0

- Comparing FINIX and FRAPCON results showed that FRAPCON performed slightly better
  - FINIX total error was 7.5% and FRAPCON was 5.9%
  - Both struggled in the high burnup IFA629-3R6 case





Far left: The FINIX masterplot from previous slide.
Left: Similar masterplot for FRAPCON-4.0 predictions.

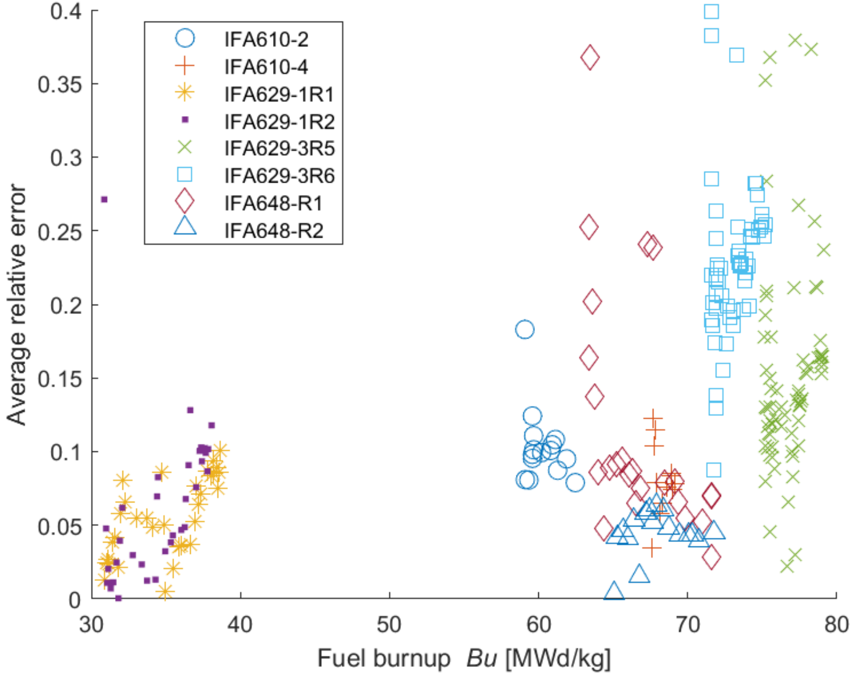
Inspecting the errors

- The error was measured as relative error pointwise

$$\Delta T_{rel} = \frac{|T_s - T_r|}{T_r}$$

  - Interpolation was used to get the same timestep for both simulation and reference data

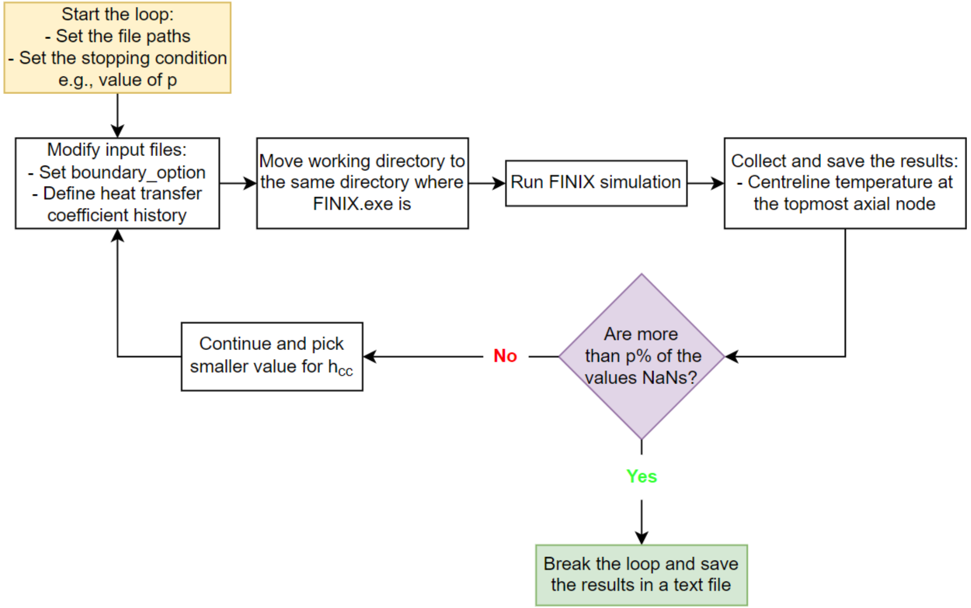
Right: The relative error of the simulations plotted as a function of fuel burnup. The figure shows well that the error increases as burnup increases.



Further research
- sensitivity analysis

Sensitivity analysis for the thermal boundary condition

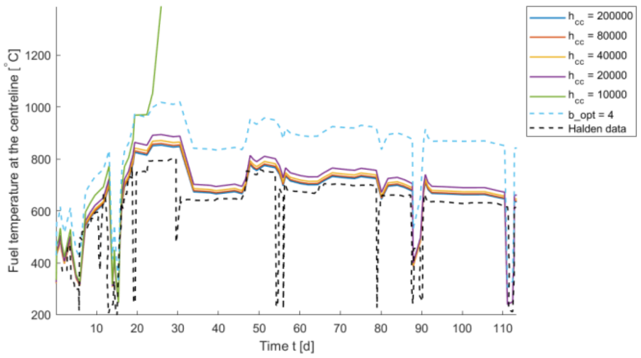
- The idea was to see where the solution breaks and what values give the best fit
  - Breaking was caused by infinite temperature values resulting NaN output
- The implementation was done with a MATLAB script

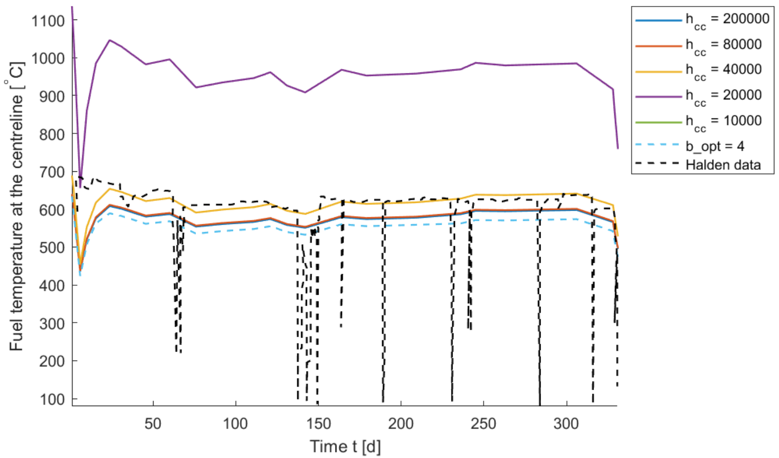


Above: Diagram explaining the logic of the boundary option testing script.

Results from the sensitivity analysis

- The best results were obtained with option that uses user-given heat transfer coefficient h_{cc} and coolant bulk temperature
- Heat transfer coefficient h_{cc} values around 1e4 broken the simulation





Above: Results for fuel centreline temperature in IFA648-R1, when the boundary options were modified.
Left: Results from IFA629-3R6 with the boundary option modifications. The breaking of the simulation is clearly visible with the green line.

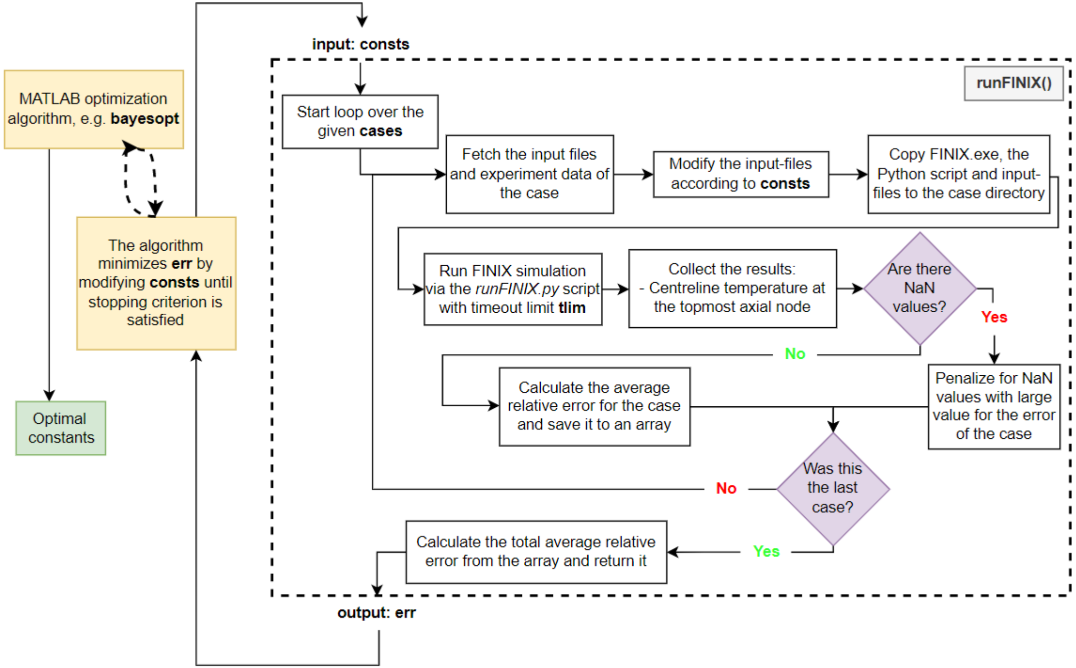
Taking the idea further…

- Implementing an optimization script
  - Allows to find optimal input parameters
  - Useful for model development and validation
- The script was based around the optimization tools available in MATLAB
  - `fminsearch`, `fmincon`, `bayesopt`
- The script was first tested for finding optimal heat transfer coefficient h_{cc} value
  - Later testing performed with MOX fuel thermal conductance λ_{th} model

The iteration logic of the optimization script

- Make use of parallel processing
- Implement weighting procedure to guide the optimization algorithm
- Mathematically:

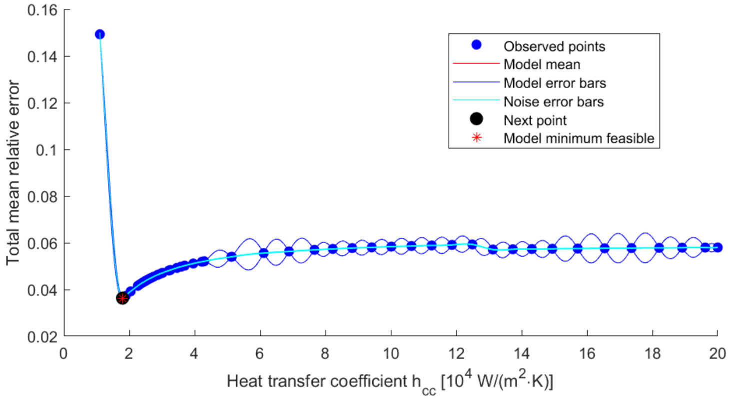
$$
\begin{aligned}
\text{min.} \quad & \frac{1}{n}\sum_{j=1}^{n}\omega_j \frac{1}{m_j}\sum_{i=1}^{m_j}\frac{|f(\mathbf{x})_i - y_{ij}|}{y_{ij}} \\
\text{s.t.} \quad & |\mathbf{x}_k| \le b_k, \quad \forall k \in [1,l] \\
& f : \mathbb{R}^l \to \mathbb{R}^m \\
& j \in [1,n] \quad \text{cases} \\
& i \in [1,m_j] \quad \text{time steps} \\
& k \in [1,l] \quad \text{varied parameters}
\end{aligned}
$$

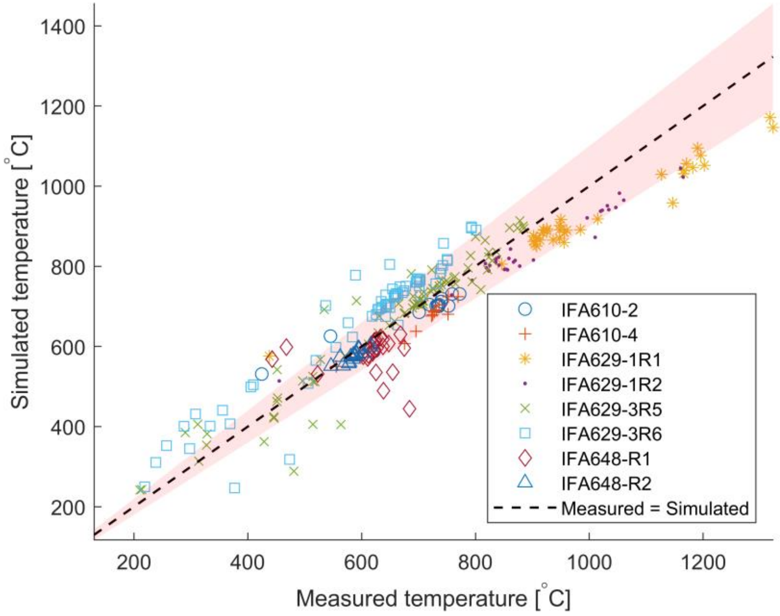


Above: Diagram showing the iteration logic of the optimization script.

Results from the optimization of heat transfer coefficient

- Optimal value for heat transfer coefficient h_{cc} was found
  - This decreased the total error to the FRAPCON-4.0 level

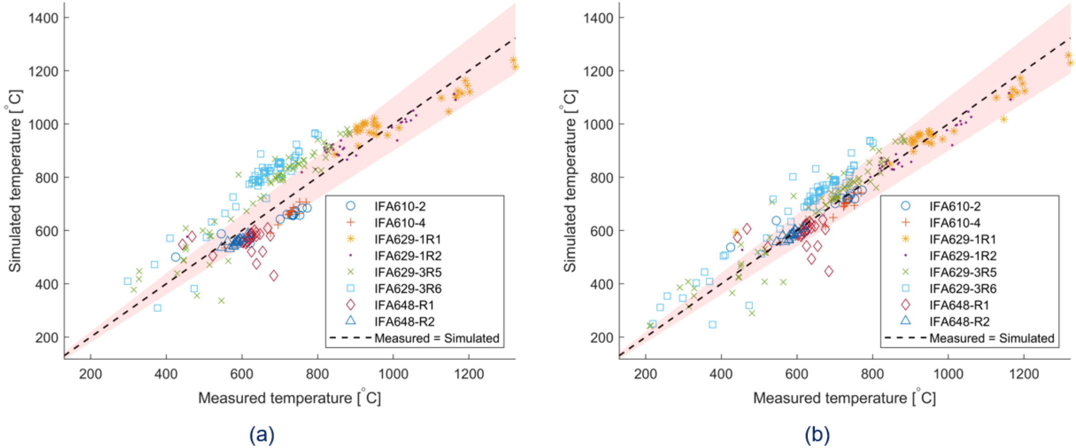


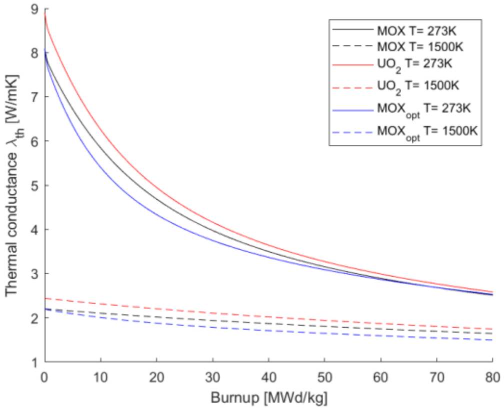


Above: Masterplot showing the measured vs. predicted fuel centreline temperatures with optimized h_{cc}.
Left: The model for the dependency of the total error and heat transfer coefficient.

Optimizing the fuel thermal conductance model for MOX fuels

- Optimizing the burnup dependency in the model improved the results quite similarly as the h_{cc} value





Above: (a) The masterplot from the MOX cases prior to any optimization. (b) Masterplot showing the performance of FINIX with the optimized fuel thermal conductance model λ_{th}.
Left: Fuel thermal conductance as a function of burnup in different temperatures. The optimized curves (blue) show the high burnup behaviour changes.

Issues with the optimization approach

- The approach assumes that the error comes from the models/inputs and thus cannot adapt to for instance instrumentation errors
  - One solution would be to filter the input and reference data beforehand
- The algorithm needs some kind of heuristics in order to escape from local minimums and keep the model physical
  - Bayesian inference with informative priors for model parameters could help
- For higher number of cases a more efficient tool would be needed
  - Python implementation for a computer cluster is under work
    - It utilizes the *Bayesian optimization* library [6]

Summary

- FINIX fuel performance code can now be used to model MOX fuels
- MOX implementation was validated against experimental data and the state-of-the-art FRAPCON-4.0 code
  - The total relative error was less than 10%
  - Differences between FRAPCON and FINIX were small
    - Both codes shared the same difficulties

- The boundary options of FINIX were studied extensively
- A new kind of optimization approach was demonstrated and its potential for model development was shown

Thank you

Questions?

References:

[1] J. Leppänen, V. Valtavirta, A. Rintala, et al., *Current Status and On-Going Development of VTT's KRAKEN Core Physics Computational Framework.* Energies, vol. 15, no. 3, p. 876, 2022.

[2] D. D. Lanning, C. E. Beyer, and K. J. Geelhood, *FRAPCON-3 Updates, Including Mixed-Oxide Fuel Properties* . Technical Report PNNL-19418, Pacific Northwest National Laboratory, 2015.

[3] Ali R Massih. *Models for MOX fuel behaviour A selective review*. Technical Report SKI-R–06-10, Sweden, 2006.

[4] K. J. Geelhood, W. G. Luscher, P. A. Raynaud, and P. I. E, *FRAPCON-4.0: A Computer Code for the Calculation of Steady-State, Thermal-Mechanical Behavior of Oxide Fuel Rods for High Burnup.* Tech. Rep. PNNL-19418, 2015.

[5] K. J. Geelhood and W. G. Luscher, *FRAPCON-4.0: Integral Assessment.* Tech. Rep. PNNL-19418, 2015.

[6] Nogueira F., *Bayesian optimization: Open source constrained global optimization tool for Python*., 2014, https://github.com/fmfn/BayesianOptimization

Acknowledgements:

The work for this study was performed during summer 2022 and was funded from VTT project GG Compa 2022.