The Python Battery Optimisation and Parameterisation Package (PyBOP)

Brady Planden, Nicola Courtier, David Howey
Department of Engineering Science, University of Oxford, UK

Battery Intelligence Lab

UNIVERSITY OF OXFORD

Abstract

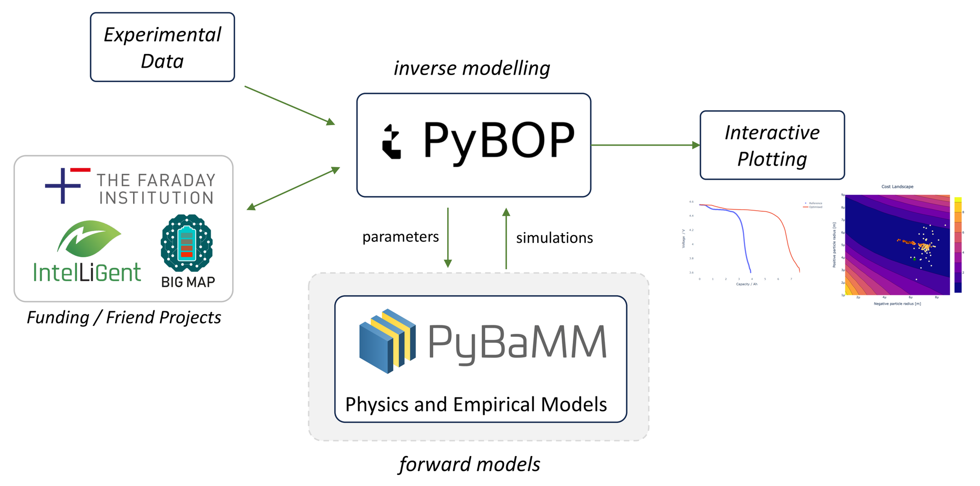
Battery models are complex and require many parameters. Identification of these parameters is challenging due to:

- **Parameter uniqueness is unknown** (in principle or in practice).
- Users may not be experts in modelling, but still want to know if their models are fit for purpose.
- **Excitation signals are not obvious** in gathered test data.
- Battery models are used over a range of temperatures, pressures and C-rates, so model parameters need to be determined accordingly.

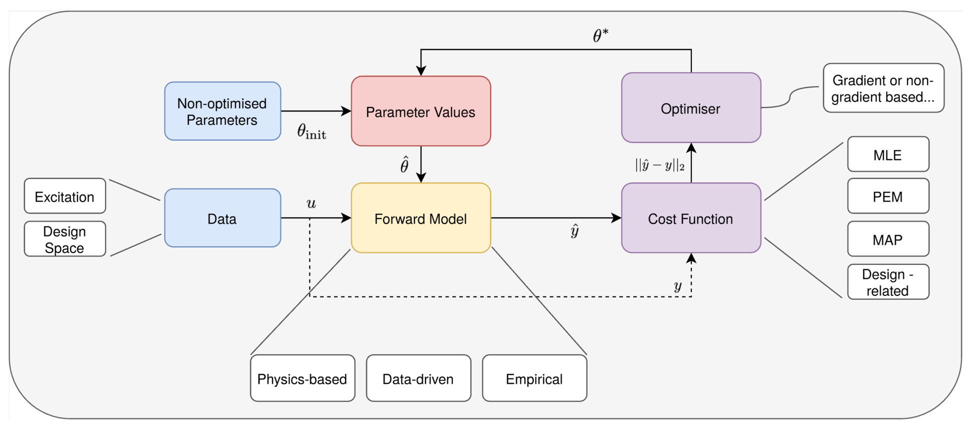
Many ad hoc methods have been developed to address to these challenges. However, a standardised framework for identification and optimisation is required to ensure that parameters are correctly identified.

Design Philosophy

PyBOP[1] is a modular battery model parameterisation and optimisation package, with a focus on classes built upon PyBaMM[2] models. This is achieved through a variety of **probabilistic methods**[3] for time series parameter identification and optimisation with physical and empirical type models.



PyBOP's hierarchical design with interface to PyBaMM



PyBOP's internal design philosophy includes modular cost functions, optimisers, and forward models.

```python
# Generate problem, cost function, and optimisation class
problem = pybop.FittingProblem(model, parameters, dataset)
cost = pybop.SumSquaredError(problem)
optim = pybop.Optimisation(cost, optimiser=pybop.CMAES)
x, final_cost = optim.run()
```

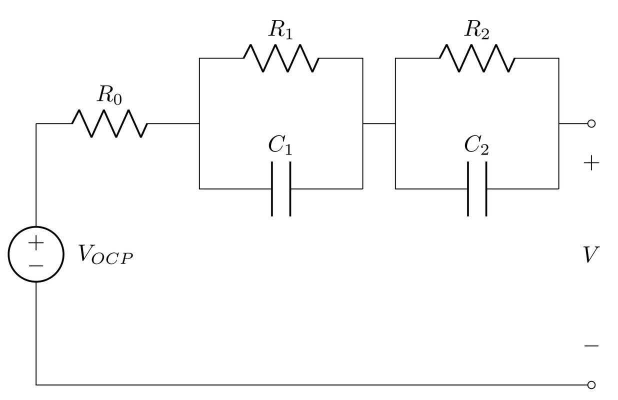
Cost functions: RMSE, SSE, MLE, MAP

Optimisers: Gradient Descent, Adam, IRPropMin, CMAES, XNES, SNES, NelderMead, PSO

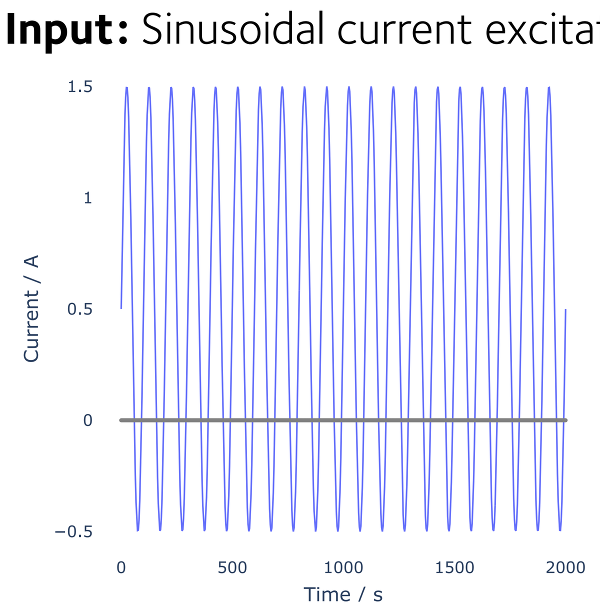
Parameterisation and Optimisation

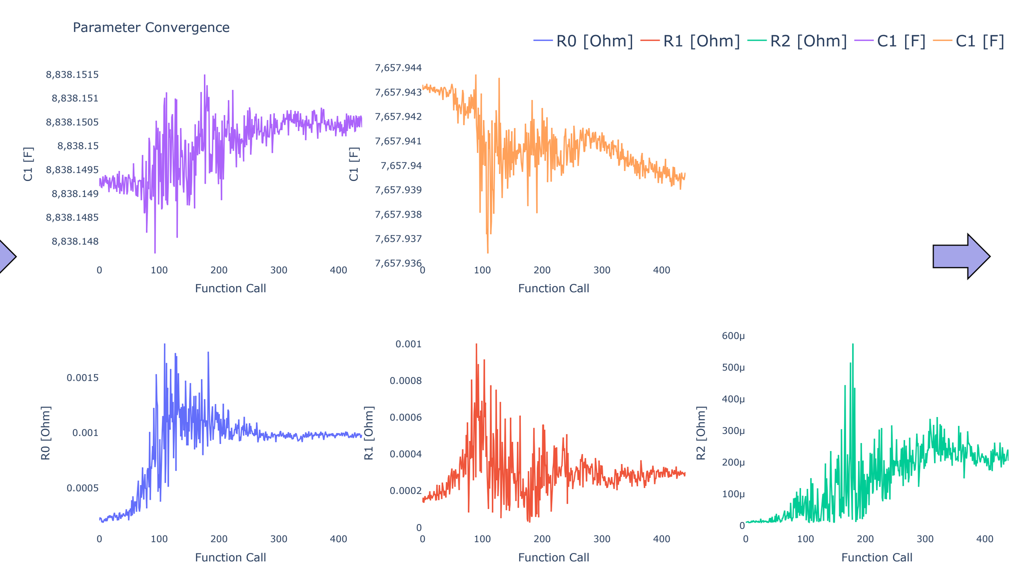
Parameter Identification

Target: Identify R0, R1, R2, C1, C2



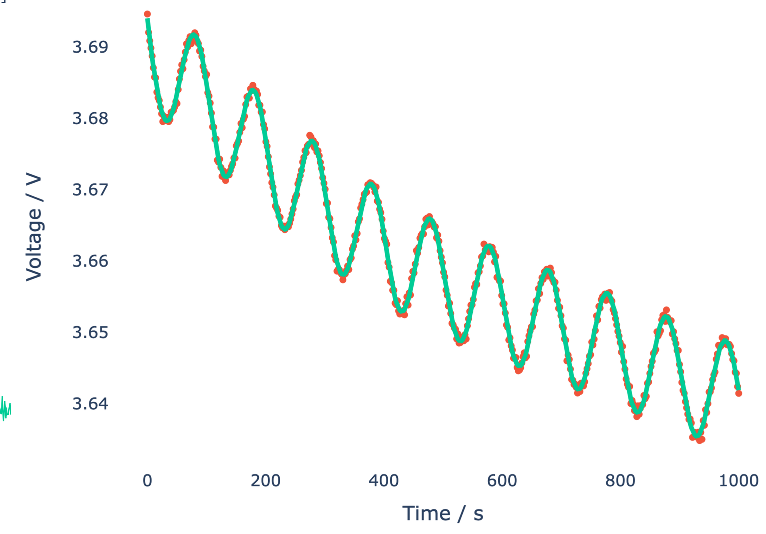
Input: Sinusoidal current excitation





Parameter convergence plot, note observability challenges with C1, C2

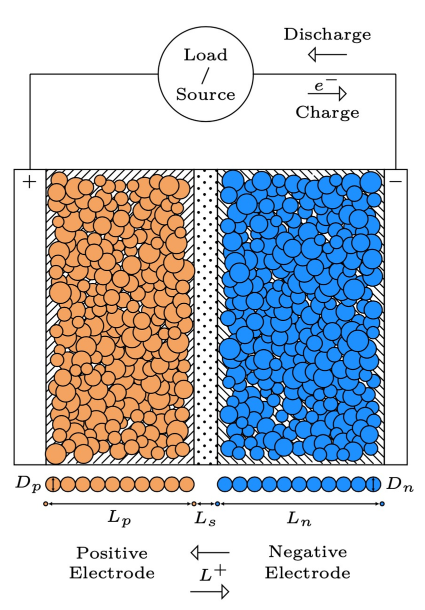
Parameter identification via terminal voltage, V

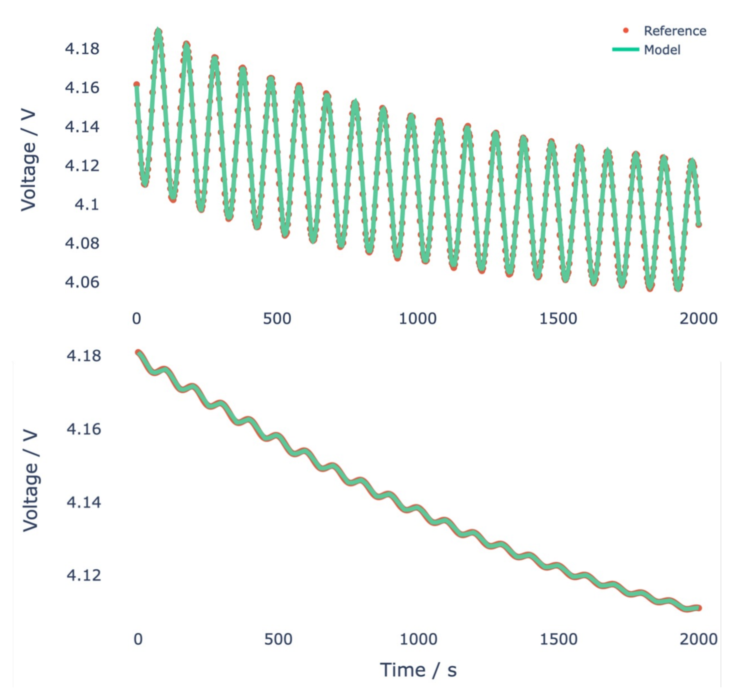


(Top) Empirical parameter identification showing the challenges of parameter observability with general system excitation.

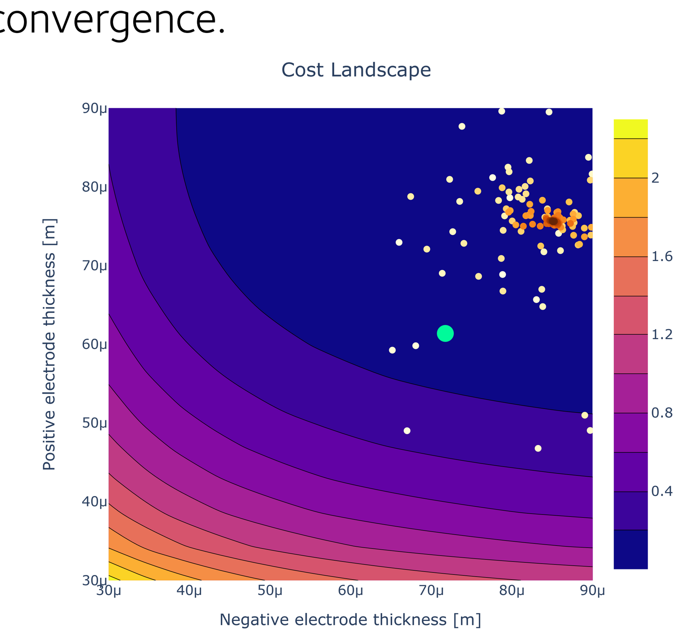
(Bottom) Physics-based electrode thickness parameter identification with two observed signals and corresponding cost landscape with optimiser convergence.

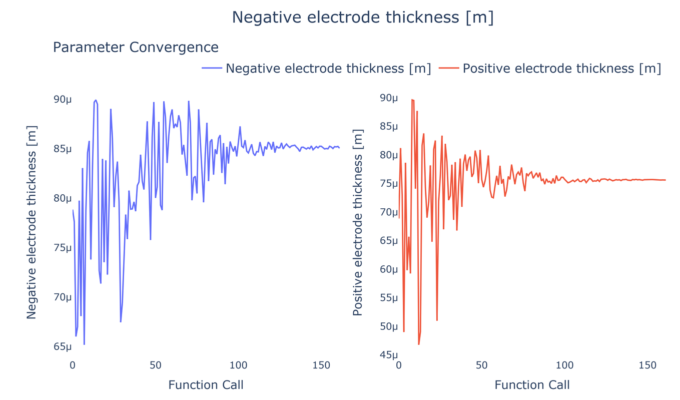
Target: Identify electrode thicknesses [L_p, L_n]





Parameter identification via terminal voltage, V and OCV contribution, V_{ocv}. The cost function is integrated over both signals with equal weighting.

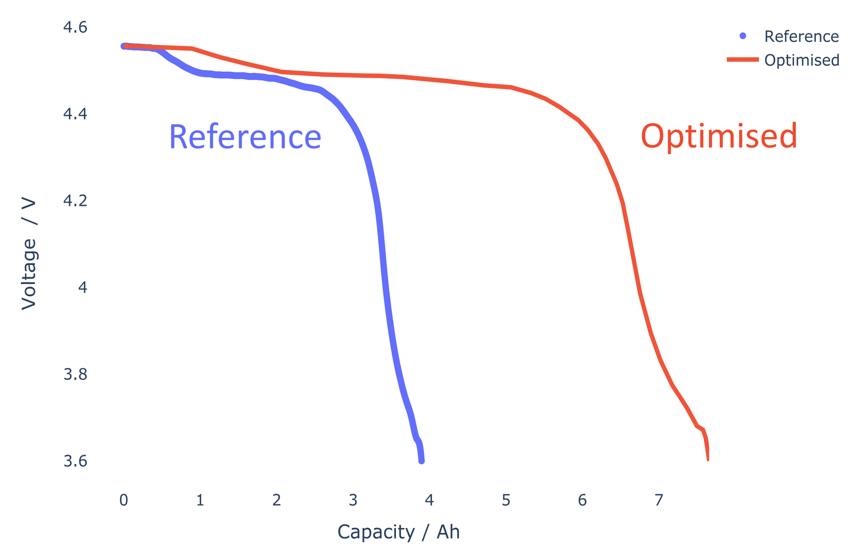


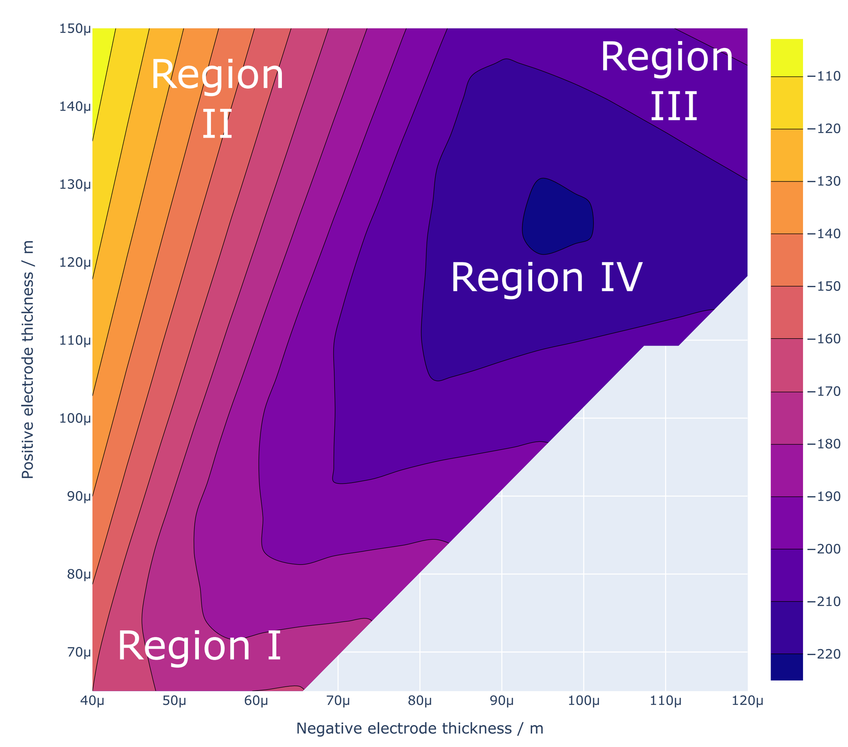


Parameter convergence plots providing information on the performance of the optimiser and cost function.

Design Optimisation

Target: Find optimal electrode coating thicknesses for maximum specific energy. The chemistry is LNMO-Gr/SiOx, with a Single Particle Model (SPM) cycled at 1C optimised with a Particle Swam Algorithm (PSO).





Region I: Electrodes too thin compared to inactive material
Region II: Poor N:P balance
Region III: Electrodes too thick causing transport limitations
Region IV: Ideal N:P balance and optimised wrt active material and transport

Conclusions

- PyBOP presents methods for parameterisation and optimisation of a variety of battery models, enabling rapid parameter verification and comparison.
- Workflows for parameter identification of equivalent circuit and physics-based models have been presented, discussing parameter uniqueness and identifiability.
- A design optimisation workflow for LNMO – Gr/SiOx chemistry was presented with physical insights.

GitHub Repository: pybop-team/pybop

Installation: `pip install pybop`

References

[1] **Planden, B., Courtier, N., & Howey, D.** Python Battery Optimisation and Parameterisation (PyBOP) (v24.3) [Computer software].

[2] **Sulzer, V., et al.** "Python Battery Mathematical Modelling (PyBaMM)." Journal of Open Research Software, vol. 9, no. 1, Ubiquity Press, 2021.

[3] **Clerx, M., Robinson, M., Lambert, B., Lei, C. L., Ghosh, S., Mirams, G. R., & Gavaghan, D. J.** (2019). Probabilistic Inference on Noisy Time Series (PINTS). Journal of Open Research Software, 7(1), 23.

Acknowledgements

This work was supported by the EU IntelLiGent and Faraday Institution Multi-scale Modelling Projects.

UK Research and Innovation | THE FARADAY INSTITUTION | IntelLiGent

Examples | Docs | Benchmarks