AI-Orchestrated Physics-Informed Modelling and Predictive Control with Applications in Energy-Intensive Manufacturing and Battery Management

By

XIAOFEI HAN



A thesis submitted for the degree of
Doctor of Philosophy

School of Electronic & Electrical Engineering
Faculty of Engineering
UNIVERSITY OF LEEDS

SEPTEMBER 2026

Abstract

The rapid advancement of artificial intelligence (AI) in engineering has created both opportunities and challenges for complex system modelling and control. Data-driven approaches can approximate complex behaviour but may lack physical interpretability and generalisation, whereas physics-based methods can be too computationally demanding for real-time use. This thesis develops a unified AI-orchestrated physics-informed modelling and predictive control framework integrating data-driven learning, physical knowledge, and optimisation-based decision-making. By embedding engineering knowledge directly into learning and control processes, the framework seeks to improve reliability while retaining the computational advantages of AI. It is developed progressively across energy-intensive manufacturing and battery management, covering steel reheating, single-cell charging, and multi-cell pack balancing.

Grounded in this framework, one primary contribution in this thesis is the real-time prediction of transient heat transfer in steel reheating furnaces, where nonlinear partial differential equations, limited measurements, scarce industrial data, and changing operating conditions challenge both black-box and numerical approaches. A physics-informed deep learning framework, termed EngGeneNet, is developed for 2D temperature-field prediction in large-scale intermediate steel stock. It combines a convolutional neural network (CNN) with a Crank–Nicolson-based physics-informed loss, an Engineering-Gene (Eng-Gene) module embedding the temperature-dependent Fourier number, and knowledge distillation with transfer learning. These components allow the model to use physical structure both during training and when generalising to previously unseen operating conditions. Validation on a pilot-scale walking-beam bloom reheating furnace shows accuracy comparable to the conventional alternating direction implicit (ADI) method, while reducing per-frame computation time by **96**%. EngGeneNet therefore enables real-time prediction suitable for closed-loop furnace control.

Beyond modelling, a further contribution in this thesis is the battery charging control process optimisation, where charging speed and internal-temperature safety are competing objectives and historical trial data are often underused. A historical data-driven self-learning model predictive control (MPC) framework combines the trial-to-trial learning capability of iterative learning control (ILC) with MPC constraint handling. A convex mapping constraint, constructed from the convex hull of historical state trajectories, narrows the feasible solution space, while a memory function-enhanced cost function uses previous charging iterations to refine the current trajectory. This enables the controller to improve performance progressively without relying on unsafe exploratory interactions

with the battery. In a lithium iron phosphate ($LiFePO_4$) case study, the proposed method reduces charging duration by 70–100 s relative to an existing constrained generalised predictive control benchmark, strictly satisfies internal temperature constraints, and reduces solve-time variability by approximately **80%** relative to standard MPC.

Finally, the framework is extended from single-cell control to multi-cell battery-pack balancing, where state-of-charge (SOC) equalisation, thermal regulation, and output-voltage stabilisation must be achieved simultaneously under coupled electrical and thermal dynamics. A 2D Cost Map-Guided Parallel-Inference Neural Modelling and Predictive Control Framework is proposed. A unified multi-task neural model predicts coupled states under candidate balancing actions, while an inertia-aware 2D cost map represents the different time constants of electrical and thermal states. This formulation permits fast comparison of multiple candidate actions while accounting for the distinct responses of electrical and thermal variables. Parallel Gaussian sampling and softmax-weighted aggregation determine the control action without gradient-based optimisation. Simulations under three real-world driving-cycle profiles achieve an inter-cell temperature difference below 5 °C, terminal SOC non-uniformity below 1%, voltage regulation within a ±10% error band, and a per-command decision time of 0.121 s.

Taken together, the proposed frameworks demonstrate that integrating physical laws, engineering knowledge, and data-driven learning can balance model fidelity, computational efficiency, interpretability, and adaptability. The progression from physics-informed modelling to historical-data-driven control and parallel multi-objective decision-making provides generalisable solutions for complex engineering systems characterised by coupled multi-physics dynamics, limited data, repetitive operation, and stringent real-time constraints.

NOMENCLATURE

Abbreviations

ADI Alternating Direction Implicit

BBS Battery Balancing System

BJDST Beijing Dynamic Stress Test

BMS Battery Management System

CC-CV Constant Current-Constant Voltage

CFD Computational Fluid Dynamics

CNNs Convolution Neural Networks

CZ Control Zone

DASC DAgger-Based Stochastic Control

DDC Data-Driven Control

DDPG Deep Deterministic Policy Gradient

DNNs Deep Neural Networks

DOD Depth of Discharge

EngGeneNet Engineering-genes Network

EOD End of Discharge

ESS Energy Storage System

EVs Electric Vehicles

FCN Fully Connected Network

FDEs Finite Difference Equations

FDM Finite Difference Method

FFT Fast Fourier Transform

FLC Fuzzy Logic Control

FPN Feature Pyramid Network

FUDS Federal Urban Driving Schedule

GA Genetic algorithm

GPC Generalised Predictive Control

GPR Gaussian Process Regression

HEVs Hybrid Electric Vehicles

ILC Iterative Learning Control

LLM Large Language Model

LSTM Long Short Term Memory

MB Modular Battery

MBRL Model-Based Reinforcement Learning

ML Machine Learning

MMC Multi-level Converter

MPC Model Predictive Control

NMPC Neural Model Predictive Control

OC Operating Condition

P2D Pseudo-two-Dimension

PDEs Partial Differential Equations

PDF Probability Density Function

PICNN Physics-Informed Convolution Neural Network

PINN Physics-Informed Neural Network

PPO Proximal Policy Optimization

PSO Particle Swarm Optimization

RL Reinforcement Learning

RNN Recurrent Neural Network

SAC Soft Actor-Critic

SB Steel Bloom

SD Standard Deviation

SMPC Stochastic Model Predictive Control

SOC State of Charge

SOH State of Health

SPM Single Particle Model

TD3 Twin Delayed DDPG

US06 Supplemental Federal Test Procedure (SFTP) US06

Parameters and Variables

$\mathbf{H}_T$ Timesblock output (-)

$\mathbf{u}_k^*(t)$ Solved optimal control input sequence (-)

$\alpha_Q, \alpha_T, \alpha_V$ Inertia adjustment coefficients (-)

β Loss function weighting factor [-]

Δt Sampling time interval (s)

ϵ Crank-Nicolson method coefficient[-]

$\hat{Q}$ Predicted SOC (-)

$\hat{T}$ Predicted temperature ($^\circ C$)

$\hat{V}$ Predicted voltage (V)

$\mathbf{R}_k^d$ Final data representation (-)

$\mathbf{x}_{\text{elec}}$ Electrical state vector(-)

$\mathbf{x}_{\text{therm}}$ Thermal state vector (-)

$\mathbf{X}_t$ 2D state input matrix (-)

$\mathscr{B}(\cdot)$ Bounding function (-)

ω Weighted gate for action samples (-)

∂D Boundary domain [-]

$\Phi_x[\cdot]$ Backbone of the PINN[-]

ρ Density (kgm^{-3})

σ Stefan-Boltzmann constant ($Wm^{-2}K^{-4}$)

τ Total sampling time (s)

$\mathbf{M}$ Memory function set (-)

$\mathbf{X}$ Historical model states set (-)

$\widetilde{\mathscr{L}}$ Loss function of the proposed EngGeneNet[-]

$\widetilde{T}$ Temperature distribution predicted by the EngGeneNet ($^\circ C$)

B The batch in a candidate action pool (-)

C_1 Capacitance in the RC sub-circuit (F)

C_n Nominal capacity (As)

$C_p(T)$ Specific heat ($Jkg^{-1}K^{-1}$)

C_{p1} Heat capacity of the battery internal ($Jkg^{-1}K^{-1}$)

C_{p2} Heat capacity of the battery surface ($Jkg^{-1}K^{-1}$)

d Space interval (m)

D_I Interior computation domain [-]

$diag$ Diagonal matrix (-)

e Emissivity [-]

e_{soc} SOC inconsistency error (-)

e_T Temperature inconsistency error (-)

e_V Output voltage error (-)

Fo Fourier number[-]

H Input weighting factors in cost function (-)

h_c Convection heat transfer coefficient ($Wm^{-2}K^{-1}$)

h_k^d Value embedding (-)

I Load current (A)

$i(t)$ Charging current (A)

$J(t)$ Cost function (-)

K Iteration number (-)

K Length of historical state sequence (-)

$k(T)$ Thermal conductivity ($Wm^{-1}K^{-1}$)

k_1 Heat transfer coefficient between the battery internal and surface ($Wm^{-1}K^{-1}$))

k_2 Heat transfer coefficient between the battery surface and ambient environment ($Wm^{-1}K^{-1}$))

L Length of the stock(m)

m Number of nodes in x dimension[-]

MB The modular battery (-)

MLP_{SOC} SOC prediction subnetwork (-)

MLP_T Temperature prediction subnetwork (-)

MLP_V Terminal voltage prediction subnetwork (-)

N Number of cells in a pack (-)

n Number of nodes in y dimension[-]

N_p Prediction horizon by MPC (-)

P Prediction horizon (-)

p_k^d Position embedding (-)

Q State of charge (-)

Q, H Weighting factors in the cost function (-)

Q_J Internal resistance ohmic heat (J)

R_0 Internal resistance (Ω)

R_1 Resistance in the RC sub-circuit (Ω)

r_Q, r_T, r_V State weighting factors in cost function (-)

SH The high-side switch (-)

SL The low-side switch (-)

T Temperature ($^\circ C$)

t Time step (-)

T_A Ambient temperature ($^\circ C$)

T_m Mean value of the temperature distribution

T_a Ambient temperature ($^\circ C$)

T_{in} Battery internal temperature ($^\circ C$)

T_{sf} Battery surface temperature ($^\circ C$)

U Candidate action pool (-)

$u'(t)$ Battery electric sub-model input (-)

$u(t)$ Battery thermal sub-model input (-)

$u(x,t)$ Solutions of the PINN

u_i Control action (-)

U_{ocv} Open circuit voltage (V)

V Pack output voltage (V)

V_1 Voltage of the RC sub-circuit (V)

$w, \widetilde{W}$ Input and output of the Eng-Gene module[-]

ACKNOWLEDGEMENTS

A PhD is rarely the straight line you imagine it to be when you start. Looking back at my four years in Leeds, I realise how much I have grown, not just as a researcher, but as a person, and how much of that growth I owe to the people around me. I feel genuinely lucky.

None of this would have happened without my principal supervisor, Professor Kang Li. I am deeply grateful to Professor Li for giving me the opportunity to pursue my PhD in the UK in the first place, and for the patience, encouragement, and generous support he has shown me throughout. He built a research environment that is efficient, open, and genuinely inspiring to work in. Whenever I ran into difficulties, whether in the research itself or in everyday life, he was always there to help. His guidance has shaped not only the way I approach research problems, but the way I think and work more broadly.

I am also very grateful to my second supervisor, Dr. Benjamin Chong, who patiently helped me work through many technical challenges I encountered along the way. His thoughtful advice and practical suggestions have made a real difference to this thesis. I would also like to thank my third supervisor, Dr. Li Zhang, whose conscientious and dedicated approach to everything she does has been a constant source of inspiration. Watching her work made me want to do better.

My thanks also go to the other staff in ICaPNet, Dr. Sadegh Azizi, Dr. Amir Abiri Jahromi, and Dr. Tong Zhang, for their support and encouragement. I have also been fortunate to learn from some wonderful senior group members: Dr. Chen Xing, Dr. Yang Zhang, Dr. Jialei Su, Dr. Jiayang Zhang, and Dr. Wei Zuo. Their willingness to share their knowledge and experience made my time in the group much richer. And to the rest of the group, thank you for the stimulating discussions, the help you offered whenever I needed it, and for making the lab such an enjoyable place to be. The memories we made together are ones I will carry for a long time.

Research life can be intense, and I am glad I found a way to switch off. A big thank you to my badminton friends, who gave me something to look forward to every week and were genuinely great company through some of the tougher stretches. Those evenings on the court were more important than they probably knew.

I am especially grateful to my girlfriend, Yayu Chen, who has been by my side throughout this entire journey. Four years apart, across countries, is not easy, but here we are. Thank you for being there through all of it, for the patience and warmth you have always given me, and for being the first person I wanted to call whenever things went wrong (or right). I am really glad we made it to this point together. Meeting you is,

without a doubt, the best thing that has happened to me. I look forward to everything that is still ahead of us.

And to my parents, thank you. You have been my foundation throughout this whole journey. Your trust in me, your quiet and unconditional support. I am sorry I could not be with you as much as I would have liked over these four years. None of what I have achieved would have been possible without you.

Finally, I would like to thank my homeland China and the China Scholarship Council (CSC) for sponsoring my PhD studies and giving me the chance to experience the world beyond. It has been an opportunity I will always be grateful for.

DECLARATION

I declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements.

Details of my publications as first author which have been used in this thesis are as follows:

- In Chapter 2: X. Han and K. Li, "AI-orchestrated data-driven modelling and control in battery management systems: A review," in *2024 6th International Conference on Industrial Artificial Intelligence (IAI)*. IEEE, 2024, pp. 1–7.
  As the lead author, the candidate conceived the review scope, surveyed the relevant literature, and wrote the paper.

  Prof. Kang Li, my main supervisor, supervised the work, provided excellent technical advice, and modified and proof-read the draft of the paper.

- In Chapter 2: X. Han, Y. Hu, and K. Li, "Dynamic modelling for improved furnace operation and control," in *Encyclopedia of Systems and Control Engineering*. Elsevier, 2026, pp. 245–261.
  As the lead author, the candidate developed the technical content and wrote the chapter.

  Prof. Kang Li, my main supervisor, supervised the work and provided excellent technical advice.

  Dr. Yukun Hu provided technical advice and contributed to the revision of the manuscript.

- In Chapter 3: X. Han, K. Li, C. Wang, J. Zhang, and Y. Hu, "Transient heat conduction modelling for real-time operation and control of steel-making reheating furnaces—a physics-informed knowledge distillation-assisted EngGeneNet approach," *Control Engineering Practice*, vol. 164, p. 106472, 2025.
  As the lead author, the candidate developed the physics-informed modelling framework and validation experiments, and wrote the paper.

Prof. Kang Li, my main supervisor, supervised the work, provided excellent technical advice, and modified and proof-read the draft of the paper.

Dr. Chuan Wang and Dr. Jiayang Zhang provided technical advice and contributed to the revision of the manuscript.

Dr. Yukun Hu provided domain expertise on furnace operation and contributed to the revision of the manuscript.

- In Chapter 4: X. Han, K. Li, F. Gao, and Y. Zhou, "Historical data-driven self-learning control of battery charging with convex mapping constraints," *Journal of Energy Storage*, vol. 152, p. 120430, 2026.
  As the lead author, the candidate developed the self-learning control framework and validation experiments, and wrote the paper.

  Prof. Kang Li, my main supervisor, supervised the work, provided excellent technical advice, and modified and proof-read the draft of the paper.

  Prof. Furong Gao and Dr. Yuanqiang Zhou provided technical advice and contributed to the revision of the manuscript.

- In Chapter 5: X. Han, K. Li, "2D cost map-guided deep neural modelling and control for battery pack balancing", *Applied Energy* (under review).
  As the lead author, the candidate developed the neural modelling and control framework and validation experiments, and wrote the paper.

  Prof. Kang Li, my main supervisor, supervised the work, provided excellent technical advice, and modified and proof-read the draft of the paper.

  X. Han, K. Li, "Vision-Based Spatio-Temporal Convolutional Load Forecasting for Battery Pack Balancing in Electric Vehicles", in *2026 8th International Conference on Industrial Artificial Intelligence (IAI)*. IEEE, 2026, (Accepted).
  As the lead author, the candidate developed the neural modelling framework and validation experiments, and wrote the paper.

  Prof. Kang Li, my main supervisor, supervised the work, provided excellent technical advice, and modified and proof-read the draft of the paper.

TABLE OF CONTENTS

Page

List of Tables **xix**

List of Figures **xxi**

1 Introduction **1**

1.1 Background 1

1.2 Challenges 6

1.2.1 Modelling Challenges 6

1.2.2 Control Challenges 7

1.3 Research Motivations and Objectives 8

1.4 Main Contributions of the Thesis 9

1.5 Thesis Structure 12

2 Literature Review **15**

2.1 Overview of the development of Data-Driven Modelling and Control in Complex Engineering Systems 15

2.2 Physics-Informed and Grey-Box Modeling Approaches 17

2.2.1 Physics-Informed Neural Networks 17

2.2.2 Grey-Box modelling with Engineering-Gene Neural Networks . . . 21

2.3 Learning-Enhanced Predictive Control 25

2.3.1 Foundations of Predictive and Iterative Learning Control 25

2.3.2 Learning-Based and Data-Driven Formulations of Model Predictive Control . . . 27
2.3.3 Reinforcement Learning for Control . . . 28
2.4 Thermal modelling and Control in Steel Industrial Reheating Furnaces . 30
2.4.1 Heat Conduction modelling within Steel Stock . . . 31
2.4.2 Furnace Control Strategies . . . 38
2.5 Modelling and Control in Battery Energy Systems . . . 40
2.5.1 Battery Modelling Approaches . . . 41
2.5.2 Control Strategies in Battery Management Systems . . . 44
2.6 Chapter Summary . . . 46

3 Physics-Informed EngGene Neural Modelling for 2D Transient Heat Conduction 49

3.1 Introduction . . . 49
3.2 General Problem Formulation for 2D Transient Heat Conduction . . . 51
3.2.1 Governing Equations and Modelling Assumptions . . . 51
3.2.2 Boundary Conditions . . . 52
3.2.3 Numerical Discretisation and the Crank–Nicolson Scheme . . . 52
3.3 Physics-Informed Learning Framework and Constraint Embedding . . . 54
3.3.1 Formulation of the 2D Loss Function . . . 54
3.3.2 Customised Loss Convolution Kernels . . . 56
3.3.3 Constrained Boundary Condition Integration . . . 57
3.4 Engineering Gene Network Architecture and Training Strategy . . . 59
3.4.1 CNN Backbone: UNet as a Surrogate PDE Solver . . . 59
3.4.2 Engineering-Gene (Eng-Gene) Module Design . . . 60
3.4.3 Knowledge Distillation for Training Data Generation . . . 62
3.4.4 Transfer Learning for Operating Condition Adaptation . . . 63
3.5 Case Study: Transient Heat Conduction in Steel Reheating Process . . . 64
3.5.1 Furnace Operating Conditions and Product Batches . . . 64
3.5.2 Knowledge Distillation and Dataset Generation for Model Training 65

3.5.3 Ablation Studies: Effectiveness of the Eng-Gene Module 67
3.5.4 Sensitivity Analysis of Model Parameters 71
3.5.5 Validation under Various Physical Constraints 73
3.5.6 Transient Temperature Evolution Performance 76
3.5.7 Model Computational Efficiency 77
3.6 Chapter Summary 78

4 Historical Data-Driven Self-Learning Predictive Control for Repetitive Processes 81

4.1 Introduction 81
4.2 General MPC Problem Formulation 84
4.2.1 General System Model 84
4.2.2 Standard MPC Problem Formulation 84
4.3 Historical Data Representation and Convex Mapping Constraints 86
4.3.1 Motivation for Historical Data Integration 86
4.3.2 Historical State and Input Datasets 86
4.3.3 Convex Hull Formulation 87
4.3.4 Convex Mapping Constraint on Terminal Predicted State 87
4.4 Memory-Enhanced Objective Design for Self-Learning Control 89
4.4.1 Design of the Memory Function 89
4.4.2 Memory Function-Augmented Cost Function 90
4.4.3 Self-Learning Mechanism 90
4.5 Self-Learning Iterative MPC: General Framework and Algorithm 91
4.5.1 Complete Augmented Optimisation Problem 91
4.5.2 General Self-Learning Algorithm 92
4.6 Case Study: Iteratively Optimised Battery Charging under Thermal Constraints 93
4.6.1 Cascading Linearised Thermoelectric Battery Model 93
4.6.2 Instantiation of the Self-Learning MPC Framework for Battery Charging 97

4.6.3 Experimental Setup and Battery Model Parameters 100
4.6.4 Constraint Formulation for Battery Charging 101
4.6.5 Validation of Cascaded Model Constraints 102
4.6.6 Effect of Prediction Horizon 103
4.6.7 Analysis of Weighting Factors 104
4.6.8 Robustness under Parameter Uncertainty 104
4.6.9 Sensitivity to Initial Solutions 106
4.6.10 Computational Efficiency Analysis 108
4.6.11 Continuous Optimisation Performance 109
4.6.12 Charging Duration Comparison 111
4.7 Chapter Summary 112

5 2D Cost Map-Guided Parallel-Inference Neural Modelling and Predictive Control Framework 115

5.1 Introduction 115
5.2 General Formulation of Multi-State Coupled Control Problems 118
5.2.1 Multi-Subsystem Architecture and State Representation 118
5.2.2 Multi-Objective Control Problem and Key Challenges 119
5.3 Unified Multi-Task State-Action Neural Modelling 120
5.3.1 Architecture Overview 120
5.3.2 2D State-Action Input Formulation 122
5.3.3 Time-Series Embedding and Unified Representation 123
5.3.4 Decomposed Multi-Rate Prediction Subnetworks 124
5.4 Inertia-Aware 2D Cost-Map Guided Parallel-Inference Control Framework 126
5.4.1 Parallel Sampling-Based Action Generation 126
5.4.2 Inertia-Aware 2D Cost Map 127
5.4.3 Weighted Action Selection Mechanism 128
5.4.4 Algorithm Summary 129
5.5 Case Study: Multi-Objective Balancing of Converter-Based Modular Battery Packs 130

5.5.1 Battery Pack Architecture and Application-Specific State Formulation . . . 130
5.5.2 Experimental Setup and Battery Pack Configuration . . . 133
5.5.3 Dataset and Implementation Details . . . 135
5.5.4 Ablation Study: Neural Prediction Model Components . . . 135
5.5.5 Overall Balancing Control Performance . . . 136
5.5.6 Parameter Sensitivity Analysis . . . 139
5.5.7 2D Cost Map Visualisation and Analysis . . . 140
5.5.8 Computation Efficiency Analysis . . . 141
5.5.9 Output Voltage Regulation under Multiple Setpoints . . . 142
5.5.10 Vision-Based Load Prediction for EV Deployment . . . 142
5.6 Chapter Summary . . . 150

6 Conclusions and Future Work 153
6.1 Summary of the Thesis . . . 153
6.2 Conclusions . . . 154
6.2.1 Physics-Informed EngGeneNet for 2D Transient Heat Conduction Modelling . . . 154
6.2.2 Historical Data-Driven Self-Learning Predictive Control for Repetitive Processes . . . 155
6.2.3 2D Cost Map-Guided Parallel-Inference Neural Modelling and Predictive Control Framework for Battery Pack Balancing . . . 156
6.3 Recommendations for Future Work . . . 157
6.3.1 Advanced Physics-Informed Architectures for Complex Geometries and Multi-Physics Coupling . . . 157
6.3.2 Uncertainty Quantification and Robust Physics-Informed Control 158
6.3.3 Extension to Multi-Agent and Distributed Energy Systems . . . 158
6.3.4 Integration of Reinforcement Learning with Physics-Informed Structure . . . 159

Bibliography **163**

Appendix A: ADI Method Derivation and Calculation for 2D Transient Heat Conduction **183**

Appendix B: Theoretical Analysis of the Convex Mapping Constraint in the Self-Learning MPC Framework **189**

List of Tables

Table **Page**

3.1 Polynomial coefficients a_i for $C_p(T)$ and b_i for $k(T)$. 60
3.2 Material properties in the experiment. 66
3.3 Operating conditions in experiments. 66
3.4 Performance comparison of various deep learning backbones with and without 'Eng-Gene' in the physics-informed framework. 68
3.5 Comparison results of the integration of the 'Eng-Gene' module or an extra convolution layer. 70
3.6 Ablation study results on different variables in 'Eng-Gene' module. 71
3.7 Comparison results on test dataset for different components in the framework. 73
3.8 Comparison results under different operating conditions in transfer learning. 74
3.9 The influence of batch size for transfer learning under OC2 (30 epochs). . . . 76

4.1 Identified thermoelectric battery model parameters. 100
4.2 SOC–U_{ocv} look-up table. 100
4.3 T_{in}–R_0 look-up table. 101
4.4 Computation time statistics for different prediction horizons (unit: seconds). . 108
4.5 Computation time standard deviation comparison at $N_p = 10$. 108
4.6 Mean and standard deviation of charging current under different internal temperature limits across 10 iterations. 110
4.7 Charging duration comparison against the reference GPC strategy. 112

5.1 Battery cell parameters. 134

5.2 Dataset sizes for training, validation, and testing. 135
5.3 Ablation study of different model components under various test datasets (SOC and temperature errors averaged over cells). 136
5.4 Balancing performance and computation time under different batch sizes B (US06, $N_p = 20$). 142
5.5 Vehicle dynamics parameters used in the physics-based speed-to-load conversion module. 147
5.6 Single-step-ahead speed prediction performance on the test dataset. 147
5.7 Regime-level speed prediction performance on the test dataset. 148

List of Figures

Figure **Page**

1.1 (a) Trends in artificial intelligence scholarly publications (2015–2023), (b) Trends in external private investment in artificial intelligence companies (2015–2024) 2

1.2 Structural organisation of the thesis. 14

2.1 General framework of PINN model. 18

2.2 Grey box modelling 22

2.3 The Eng-Gene neural modelling framework. 23

2.4 The reheating furnace and the steel stock 32

2.5 Control zones 38

2.6 Battery equivalent circuit models 42

3.1 2D heat conduction problem. 51

3.2 The framework of the proposed physics-informed EngGeneNet. 55

3.3 (a) and (b) are the customised convolution kernels for computing $Y'(x, y)$ and $Y(x, y)$, respectively. 57

3.4 Customised loss function convolution process. 57

3.5 Mask of boundary constraints. 58

3.6 'Eng-Gene' module. 61

3.7 Schematic of the reheating furnace. 64

3.8 Schematic of the intermediate steel product. 65

3.9 Metrics gap between training and test datasets of various deep learning models with and without the 'Eng-Gene' module. 68
3.10 Training and validation loss evolution of PI-EngGeneUNet and PI-UNet. . . 69
3.11 Validation loss evolution for 'Eng-Gene' module ablation study. 71
3.12 Training parameters analysis: (a) initial learning rate; (b) decay factor; (c) loss weighting factor β. 73
3.13 Several prediction results on test datasets: (a) ADI solutions; (b) PI-EngGeneNet solutions; (c) error map. 75
3.14 Predicted stock temperature history during the trial period. 77
3.15 Computation time per frame of temperature distribution prediction. 78

4.1 Illustration of the historical data convex mapping process. 88
4.2 First-order Thevenin equivalent circuit model of the lithium-ion battery. . . . 95
4.3 Lumped two-node battery thermal model. 96
4.4 Overall control framework of the proposed historical data-driven self-learning MPC for battery charging. 98
4.5 Flowchart of the proposed historical data-driven self-learning control framework applied to battery charging. 99
4.6 Comparative results: (a) without terminal voltage constraint; (b) with terminal voltage constraint. ($N_p = 15$, $H = 20$, $T_{in}^{\max} = 38\,^{\circ}\mathrm{C}$) 102
4.7 Effect of different prediction horizons: (a) $N_p = 15$; (b) $N_p = 30$; (c) $N_p = 45$. ($H = 10$, $T_{in}^{\max} = 38\,^{\circ}\mathrm{C}$) 103
4.8 Effect of weighting factor on control performance: (a) $H = 10$; (b) $H = 20$; (c) $H = 40$. ($N_p = 15$, $T_{in}^{\max} = 40\,^{\circ}\mathrm{C}$) 105
4.9 Robustness under parameter uncertainty: (a) 5% Beta uncertainty; (b) 10% Beta uncertainty. ($T_{in}^{\max} = 37\,^{\circ}\mathrm{C}$) 106
4.10 Sensitivity analysis of initial solutions: (a) CC-CV initialisation; (b) standard MPC initialisation. ($T_{in}^{\max} = 34\,^{\circ}\mathrm{C}$) 107
4.11 Violin plot of computation time distributions across different prediction horizons. 109
4.12 Memory function evolution across iterations. ($N_p = 20$, $T_{in}^{\max} = 40\,^{\circ}\mathrm{C}$) 111

5.1 Overall architecture of the proposed control-oriented multi-task neural model for multi-state coupled systems. 121
5.2 Structure of the 2D state input matrix and associated action input vector. . . 122
5.3 Proposed parallel sampling-based neural MPC framework for multi-state coupled systems. 126
5.4 Converter-based modular battery structure. 131
5.5 Battery working current and value density for the three dynamic current profiles used in the case study: (a) BJDST, (b) FUDS, (c) US06. 134
5.6 Prediction results for SOC non-uniformity, temperature non-uniformity, and pack output voltage under the three driving cycle datasets. 137
5.7 Balancing control results under three driving cycle profiles: (a) BJDST; (b) FUDS; (c) US06. Rows 1–5 show SOC trajectories, SOC non-uniformity $SD(Q)$, temperature trajectories, temperature non-uniformity $SD(T)$, and pack output voltage. 138
5.8 Balancing current distribution for individual cells: (a) BJDST; (b) FUDS; (c) US06. The step-like SOC–current profile confirms effective cell-level energy redistribution. 139
5.9 SOC and thermal non-uniformity under prediction horizons $N_p = 10$, 20, and 50 (US06 driving cycle). 140
5.10 SOC and thermal non-uniformity under different cost function weighting factor configurations. 141
5.11 Visualised 2D cost map examples for different action samples at one sampling step. 141
5.12 Output voltage regulation under setpoints of 8 V and 9 V, and the uncontrolled baseline. 143
5.13 Overview of the vision-based load prediction pipeline. 144
5.14 Representative test segment from the test dataset: (top) measured vehicle speed profile; (bottom) corresponding battery current load profile. 150

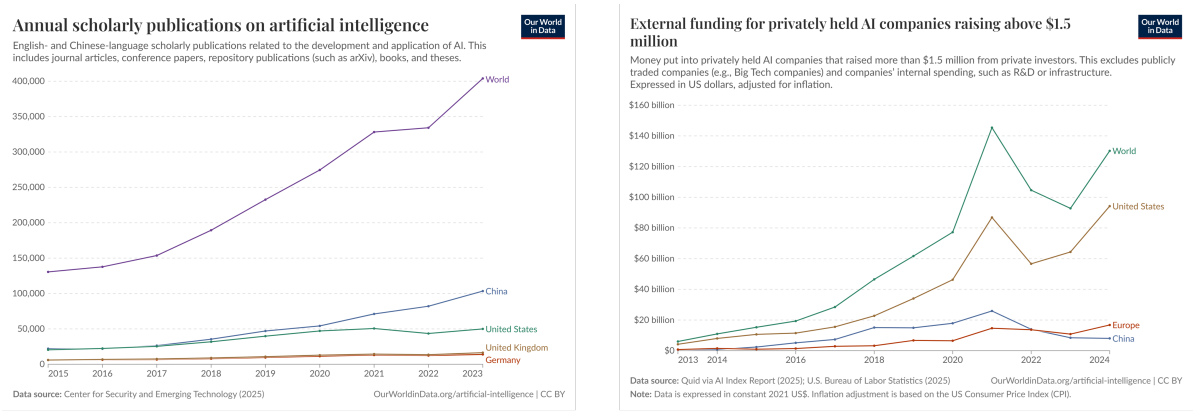
CHAPTER 1

INTRODUCTION

1.1 Background

Over the past several decades, Artificial Intelligence (AI) has experienced rapid and transformative development, evolving from a primarily theoretical discipline into a core enabling technology for real-world engineering and industrial systems. Numerous industrial reports and academic surveys consistently indicate that AI-driven technologies now play a central role in smart manufacturing [1], intelligent energy systems [2], and cyber–physical systems [3]. In this context, data-centric decision-making and AI-orchestrated automation approaches have emerged as key drivers for improving operational efficiency, system reliability, and product quality across a wide range of industrial sectors.

The rapid advancement of artificial intelligence is also clearly reflected in the growth of academic research output. As shown in Fig. 1.1 (a) [4], the annual number of scholarly publications related to artificial intelligence has increased sharply over the past decade. According to statistics from the *Center for Security and Emerging Technology (2025)* – processed by *Our World in Data*, global AI-related scholarly outputs, including journal articles, conference papers, repository publications, books, and theses, more than tripled



(a) (b)

Figure 1.1: (a) Trends in artificial intelligence scholarly publications (2015–2023), (b) Trends in external private investment in artificial intelligence companies (2015–2024)

between 2015 and 2023. In particular, the global annual volume of AI publications grew from approximately 1.3×10^5 in 2015 to over 4.0×10^5 in 2023, highlighting a sustained and accelerating research momentum. Furthermore, artificial intelligence has also experienced a substantial increase in industrial investment. As illustrated in Fig. 1.1 (b) [5], external private funding for AI companies raising more than $1.5 million has grown dramatically over the past decade. According to data in *Quid via AI Index Report (2025); U.S. Bureau of Labor Statistics (2025)*, with major processing by Our World in Data, global private investment in AI increased from less than $10 billion in the early 2010s to well over $100 billion in recent years, even after adjusting for inflation. Both the growth in academic output and the increase in industrial investment highlight the transition of AI from a research-driven field to a production-critical technology deployed at scale. As AI systems increasingly move from laboratory environments into real-world industrial applications, the emphasis has shifted from algorithmic performance alone toward robustness, reliability, interpretability, and compliance with physical and operational constraints.

Within the AI ecosystem, Deep Learning (DL) based approaches have been increasingly adopted in complex engineering systems where first-principles modelling is difficult, incomplete, or computationally prohibitive [6]. Consequently, data-driven modelling and control have gained prominence as powerful alternatives or complements to conventional

physics-based approaches. The advent of deep neural networks (DNNs) has further accelerated this transition. Deep learning extends traditional machine learning by employing multi-layer neural network architectures capable of learning highly nonlinear and hierarchical representations directly from raw data [7]. Compared with conventional machine learning (ML) techniques that rely heavily on handcrafted features, deep learning demonstrates superior capability in handling high-dimensional, unstructured, and heterogeneous data, including images, audio signals, text, and multivariate time series [8]. These strengths have led to substantial breakthroughs in computer vision [9], speech recognition [10], and natural language processing [11], with deep learning models achieving or surpassing human-level performance in specific tasks.

From an industrial perspective, deep learning has already delivered significant economic and operational benefits. For example, large-scale speech recognition systems have achieved word error rate reductions exceeding 50% compared with early-generation statistical models, enabling widespread adoption in consumer electronics and industrial human–machine interfaces [12]. In manufacturing, vision-based deep learning techniques have been reported to reduce defect detection error rates by more than 30–40% relative to traditional rule-based inspection methods, particularly in surface defect detection and quality control applications [13]. In the domain of prognostics and health management, deep recurrent neural networks, such as Long Short-Term Memory (LSTM) models, have demonstrated strong capability in remaining useful life (RUL) estimation for complex equipment, offering improved prediction accuracy and robustness under varying operating conditions [14].

Despite these achievements, the direct deployment of purely data-driven AI models in safety-critical and high-reliability engineering systems remains challenging. First, the acquisition of large-scale, high-quality, and representative training data is often costly, time-consuming, or infeasible [15]. In domains such as energy systems, transportation, and industrial manufacturing, data collection may be constrained by safety considerations, long operational cycles, or limited fault occurrences [16–18]. Second, real-world industrial datasets frequently exhibit severe class imbalance, where normal operating

conditions dominate and fault or extreme-condition data are scarce. This imbalance can lead to biased learning, overfitting, and degraded generalisation performance if not explicitly addressed. Third, deep neural networks are commonly criticised for their limited interpretability and transparency, which hinders their acceptance in engineering applications where explainability, verification, and regulatory compliance are essential [19, 20]. In contrast, traditional physics-based modelling and control approaches, while grounded in physical laws and offering strong interpretability, face inherent limitations when applied to large-scale, nonlinear, and time-varying systems. High-fidelity first-principles models often require simplifying assumptions, precise parameter identification, and significant computational resources, which can limit their accuracy and real-time applicability [21]. These challenges become particularly pronounced in industrial systems characterised by strong nonlinearities, multi-physics coupling, and evolving operating conditions.

These challenges are particularly evident in energy and thermal engineering systems, where accurate modelling and control are crucial for safety, efficiency, and longevity. In battery management systems (BMS), electrical, electrochemical, and thermal dynamics are tightly coupled, and system behaviour evolves continuously with ageing, environmental conditions, and operational history. Conventional model-based approaches often rely on simplified equivalent circuit models or lumped-parameter thermal representations [22], which may fail to capture internal state evolution under aggressive charging or high-load conditions. In contrast, purely data-driven models may achieve strong short-term prediction accuracy but often lack robustness, physical consistency, and interpretability when deployed in safety-critical environments [23]. Similar challenges are also encountered in industrial thermal processes, such as heat conduction modelling in reheating furnaces used in steel manufacturing. These systems are governed by nonlinear heat conduction equations with temperature-dependent material properties, combined radiative and convective boundary conditions, and spatially distributed states. While high-fidelity numerical models based on partial differential equations (PDEs) can provide accurate representations [24, 25], they are typically computationally intensive and unsuitable

for real-time optimisation or control. Conversely, data-driven surrogate models often struggle to generalise across varying furnace configurations, operating schedules, and product batches.

Across these application domains, historical operational data are abundant yet remain underexploited, and system dynamics exhibit strong repeatability over long operational horizons. This observation motivates the development of AI-orchestrated, physics-informed modelling and control frameworks, in which data-driven learning is guided and constrained by governing physical laws and domain knowledge. By embedding conservation principles, structural constraints, and physical consistency into learning architectures and optimisation formulations, such frameworks can substantially improve data efficiency, robustness, and interpretability while maintaining computational tractability. Moreover, AI-orchestrated approaches enable the systematic incorporation of historical state–input trajectories into decision-making processes, allowing models and controllers to adapt continuously to system degradation, parameter drift, and changing operating conditions. This capability is particularly valuable for repetitive industrial processes, such as battery charging cycles, and reheating furnace operations, where historical data provide rich information for performance enhancement and constraint shaping.

In summary, Physics-informed and data-driven modelling and control represent a unifying methodological framework for complex engineering systems. By bridging data-driven intelligence with physical insight, this framework provides a coherent foundation for addressing both industrial heat conduction modelling problems and battery management challenges, which constitute the primary application scenarios explored in this thesis.

1.2 Challenges

1.2.1 Modelling Challenges

- **Data Scarcity and Limited In-Process Measurement:** A fundamental challenge in the systems investigated in this thesis arises from the difficulty of obtaining high-quality, usable, and well-annotated data. In-process measurement of key internal states is frequently unavailable, while changes in load, ageing, material properties, or process scheduling complicate the collection of representative training data. Even when raw data are available, preprocessing and expert-driven labelling can remain time-consuming and labour-intensive.

- **Trade-off Between Black-Box Generalisation and White-Box Real-Time Feasibility:** Purely data-driven (black-box) models may generalise poorly when the furnace or battery system departs from the training distribution, while high-fidelity physics-based solvers can be too computationally expensive for the sampling intervals required by the real-time tasks considered here. This creates a practical trade-off between prediction speed, model fidelity, and reliability outside the training envelope.

- **Limited Generalisation and Extrapolation Capability:** Beyond the black-box/white-box trade-off itself, data-driven models often exhibit strong performance within the range of observed training data but struggle to extrapolate under evolving operating conditions. In many engineering systems, dynamics change over time due to ageing, degradation, environmental variation, or operating-condition shifts. Without explicit mechanisms to account for such evolution, purely data-driven models may suffer from performance degradation and reduced robustness in long-term deployment.

- **Lack of Physical Consistency and Interpretability:** Conventional deep learning models typically operate as black-box approximators, offering limited interpretability and no explicit guarantees of physical consistency. This opacity poses

significant challenges in engineering applications where compliance with physical laws, safety constraints, and regulatory requirements is mandatory. Predictions that violate physical principles can undermine trust and restrict practical adoption in safety-critical systems.

1.2.2 Control Challenges

- **Simplification–Accuracy Trade-off in Model-Based Control:** Traditional model-based control frameworks, such as model predictive control (MPC), typically require simplified system models to maintain optimisation tractability and support real-time execution. However, for the coupled battery and furnace processes considered in this thesis, such simplifications can reduce prediction accuracy, leading to a trade-off between model fidelity and controller tractability.

- **Computational Burden of High-Fidelity Optimisation:** Conversely, adopting high-fidelity models to improve prediction accuracy substantially increases the dimensionality and nonlinearity of the resulting optimisation problem, leading to computationally intensive online optimisation that is difficult to solve within the timing constraints of real-time control.

- **Reliability Limitations of Model-Free Control:** Model-free approaches, such as reinforcement learning (RL), offer an alternative that avoids explicit model construction, but they often suffer from unstable learned policies, unsafe exploration during training, and limited reliability when deployed in safety-critical engineering systems where exploratory actions can have physical consequences.

- **Optimisation Efficiency and Local Minima:** Many learning-enhanced control formulations rely on gradient-based optimisation, which is prone to becoming trapped in local minima, particularly in non-convex or multi-modal cost landscapes. This reduces optimisation efficiency and can compromise the quality and consistency of the resulting control actions.

- **Underexploited Historical Knowledge and System Evolution:** Many industrial processes exhibit strong repeatability and generate rich historical state–input trajectories, yet this information is often underexploited in conventional control approaches. Existing methods typically treat historical data as static training samples rather than as a dynamic source of knowledge, limiting the controller's ability to adapt to system evolution and leverage long-term operational experience.
- **Unified Treatment of Multi-Physics Coupling:** Complex engineering systems are inherently governed by tightly coupled multi-physics interactions, such as electro-thermal or thermo-mechanical dynamics. Capturing these interactions accurately and efficiently, while keeping the resulting control problem computationally tractable, remains challenging for both simplified physics-based models and black-box learning approaches.

Collectively, these challenges highlight the limitations of purely data-driven and purely physics-based modelling and control strategies when applied to complex engineering systems. They motivate the development of AI-orchestrated, physics-informed modelling and control frameworks that integrate data-driven learning with physical insight, historical knowledge, and optimisation-based decision-making, forming the foundation of the research presented in this thesis.

1.3 Research Motivations and Objectives

The challenges identified in the preceding section arise specifically in the battery-management and industrial thermal processes investigated in this thesis. In particular, limited access to internal temperature states, changes in material or battery operating conditions, physical consistency requirements, and real-time computational constraints motivate the development of AI-orchestrated modelling and control frameworks that integrate data-driven learning, physical knowledge, and optimisation-based decision-making. In response to these application-specific challenges, this thesis sets out the following objectives:

- **To develop a physics-informed learning approach for the investigated distributed-parameter thermal system**, by embedding governing equations, boundary conditions, and conservation laws into neural network architectures and training objectives, and evaluating its effect on data efficiency, generalisation, and physical consistency.
- **To formulate a data-driven self-learning control method for repetitive battery charging**, by exploiting historical state–input information to quantify future control cost and improve performance across charging iterations.
- **To develop a learning-enabled predictive control method for multi-cell battery balancing**, by investigating parallel-inference neural representations and structured cost formulations for efficient multi-objective control under the constraints considered.
- **To validate the proposed methods through application-specific case studies**, including pilot-scale steel reheating and simulation-based battery management studies, and to assess their effectiveness and computational feasibility under the operating conditions considered.

1.4 Main Contributions of the Thesis

The principal contributions of this thesis lie at the intersection of AI-orchestrated physics-informed modelling and predictive control for complex engineering systems, addressing two industrially motivated application domains: energy-intensive manufacturing (reheating furnace thermal modelling) and electric vehicle battery management.

1. **Physics-Informed Neural Architecture for Industrial Thermal Field Prediction.** A physics-informed 2D loss function is formulated by embedding the Crank–Nicolson discretisation of the transient Fourier heat conduction equation directly into the training objective, with hard-constrained boundary conditions that

incorporate both convective and radiative heat exchange. Building on this foundation, an *Eng-Gene* module is extracted from the governing PDE and integrated as a physically meaningful intermediate layer within a CNN backbone, encoding the temperature-dependent Fourier number as a domain-specific nonlinear activation. Applied to reheating furnace temperature field prediction, this design achieves at least 90% MAE reduction over a plain physics-informed UNet baseline and a **96% reduction in per-frame inference time** relative to the conventional ADI solver, enabling genuine real-time integration within the furnace control loop [26].

2. **Knowledge Distillation and Transfer Learning for Data-Scarce Industrial Deployment.** To overcome the chronic data scarcity encountered in industrial furnace operations, a knowledge distillation strategy is developed in which the numerically accurate ADI method acts as a teacher to generate high-fidelity synthetic training samples for the EngGeneNet student. A complementary transfer learning scheme then enables efficient adaptation to new steel product grades and shifting operating conditions, reducing retraining time by **75%** compared to training from scratch [26].

3. **Cascading Linearised Thermoelectric Modelling and Convex MPC Formulation for battery Charging.** A cascading linearised battery model is developed that decouples the coupled electrical and thermal dynamics while preserving their physically meaningful nonlinear coupling structure. By reformulating the thermal subsystem as linear in the squared current input, the model enables a fully convex MPC optimisation problem, validated experimentally with voltage RMSE ≈ 3.4 mV and internal temperature RMSE $\approx 0.82\,^{\circ}$C. To further tighten the feasible solution space, convex hulls of historical state trajectories are embedded as hard constraints in the MPC, reducing solve-time variability by approximately **80%** relative to standard MPC and providing a principled mechanism for exploiting repetitive cycle data [27].

4. **Self-Learning Iterative MPC with Memory-Augmented Cost for Optimal**

Charging. A memory function is introduced that quantifies the minimum achievable cumulative future cost inferred from historical state–input trajectories and appends it to the standard quadratic MPC cost, preserving convexity while enabling the controller to iteratively refine the charging current profile across successive cycles. The resulting self-learning iterative MPC algorithm is validated through extensive simulation, demonstrating charging duration reductions of 70–100 s against a constrained GPC benchmark, strict enforcement of internal temperature limits across multiple thresholds, and robustness to battery parameter uncertainties of up to 10%.

5. **Multi-Task Neural Prediction Model for Heterogeneous Battery Pack Dynamics.** A unified multi-task neural model is developed using a 2D image-like input representation that encodes historical multi-cell state sequences and control action vectors jointly. Decomposed electrical and thermal prediction subnetworks with sinusoidal positional embedding capture the heterogeneous temporal dynamics of fast electrical and slow cumulative thermal states, achieving maximum SOC MAE below 0.11 and temperature MAE below 0.03 per cell across three real-world EV driving cycle datasets.

6. **2D Cost Map-Guided Sampling-Based Neural MPC for Real-Time Multi-Objective Battery Pack Balancing.** A gradient-free parallel sampling MPC framework is proposed that generates a pool of Gaussian-sampled candidate action trajectories, evaluates them simultaneously through the multi-task neural model via GPU batch processing. An inertia-aware 2D cost map is further introduced as a physically grounded multi-objective evaluation metric, encoding state-specific exponential inertia coefficients to capture the heterogeneous dynamic inertias of SOC, temperature, and voltage along the prediction horizon. The final balancing action is then synthesised through softmax-weighted aggregation, entirely eliminating the need for back-propagation through the network at decision time. Comprehensive experiments under three EV driving cycles (BJDST, FUDS, US06) validate

simultaneous achievement of less than 5 °C inter-cell temperature difference, below 1% terminal SOC non-uniformity, and output voltage regulation within ±10%, with a per-command decision time of **0.121 s** confirming real-time feasibility [28].

1.5 Thesis Structure

The remainder of this thesis is organised into six chapters, as illustrated in Figure 1.2. The logical flow progresses from theoretical foundations, through methodology development, to case-study validation in corresponding engineering application domains.

Chapter 2: Literature Review. This chapter provides a comprehensive review of the methodological foundations and application-domain knowledge underpinning the research. It first surveys physics-informed neural network (PINN) approaches and grey-box modelling techniques, including the Engineering-Gene (Eng-Gene) framework, positioning them within the broader landscape of hybrid physics-data modelling. It then reviews learning-enhanced predictive control formulations, covering data-driven MPC, iterative learning control, and reinforcement learning-based strategies. The chapter subsequently addresses two specific engineering domains: thermal modelling and adaptive control of steel industrial reheating furnaces (heat conduction modelling, zone methods, and furnace-level control strategies), and modelling and control in battery energy systems (electrochemical and equivalent circuit models, battery state estimation, and charging and balancing control strategies). The chapter concludes by identifying the open research gaps that motivate the contributions of Chapters 3-5, which progress methodologically from modelling (Chapter 3), to control (Chapter 4), to an integrated modelling-and-control framework (Chapter 5)[29, 30].

Chapter 3: Physics-Informed EngGene Neural Modelling for 2D Transient Heat Conduction. This chapter presents the first technical contribution of the thesis. It formulates the 2D transient heat conduction problem in steel stock, derives the physics-informed 2D loss function based on the Crank-Nicolson discretisation scheme with hard-constrained boundary conditions, and introduces the EngGeneNet architecture

integrating the Eng-Gene module and knowledge distillation/transfer learning strategy. Comprehensive case studies on a pilot-scale walking-beam bloom reheating furnace, including ablation studies, sensitivity analyses, and computational efficiency evaluation, demonstrate a 96% reduction in per-frame prediction time relative to the conventional ADI method, alongside prediction accuracy comparable to the numerical reference [26].

Chapter 4: Historical Data-Driven Self-Learning Predictive Control for Repetitive Processes. This chapter develops the cascading linearised thermoelectric battery model, formulates the novel historical state convex mapping constraint, and introduces the memory function-enhanced cost function within an iterative self-learning MPC framework. Battery charging case studies demonstrate iterative convergence of the charging current trajectory, superior charging efficiency relative to GPC benchmarks across multiple thermal constraint thresholds, and improved computational stability through convex mapping constraint regularisation [27].

Chapter 5: 2D Cost Map-Guided Parallel-Inference Neural Modelling and Predictive Control Framework. This chapter introduces the unified multi-task state-action neural model with 2D input formulation and decomposed thermal-electrical subnetworks, the inertia-aware 2D cost map, and the parallel sampling-based NMPC framework. Simulation studies on a converter-based modular battery pack under BJDST, FUDS, and US06 driving cycle profiles validate simultaneous multi-objective SOC balancing, thermal regulation, and voltage control with real-time decision-making efficiency[28].

Chapter 6: Conclusions and Future Work. This chapter summarises the principal findings and contributions of the thesis, discusses the broader implications of the proposed physics-informed and data-driven modelling and control framework, and identifies promising directions for future research extending the presented methodologies to larger-scale, more complex, and experimentally validated engineering systems.

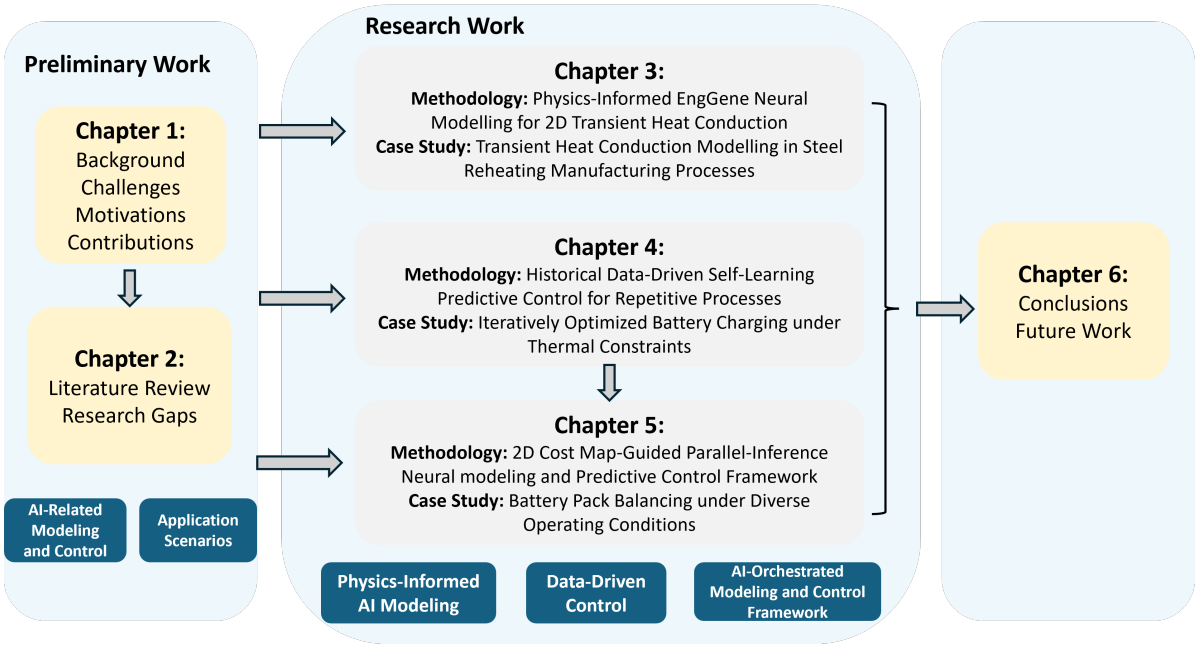


Figure 1.2: Structural organisation of the thesis.

CHAPTER 2

LITERATURE REVIEW

2.1 Overview of the development of Data-Driven Modelling and Control in Complex Engineering Systems

Over the last few decades, substantial progress has been achieved in the advancement of Artificial Intelligence (AI), and an increasing number of AI technologies have been successfully transferred into traditional industrial scenarios to improve manufacturing efficiency and product quality. Driven by rapidly growing data availability in industrial systems and the development of scalable algorithms, Data-Driven methods have been increasingly used to learn system behaviours directly from measured data, often without explicitly modelling the underlying physical mechanisms [6]. Consequently, data-driven modelling and control have become an active and influential research direction for complex engineering systems, where high-dimensional nonlinearities, uncertainties, and partial observability can make purely mechanistic modelling difficult or computationally prohibitive.

A notable feature of this evolution is that data-driven methods have progressively

expanded beyond classical regression and classification to include system identification, state estimation, anomaly detection, and closed-loop decision-making. In industrial modelling and control, data-driven approaches are typically used to (i) approximate complex input-output mappings, (ii) infer latent states that are not directly measurable, and (iii) enable predictive or adaptive control policies that respond to disturbances and operational changes. Such capabilities are especially attractive in complex engineering systems, where accurate modelling from first principles may require a large number of coupled differential equations and parameters that are difficult to identify reliably in practice. As a result, AI-driven approaches have been widely explored across diverse application domains, including manufacturing [31], agriculture [32], economics and decision science [33]. These applications collectively demonstrate that data-driven methods can discover useful structures and correlations from data, thereby enabling improved monitoring, forecasting, and control performance in real-world applications.

The rise of deep learning is closely associated with the rapid advancement of computer hardware (e.g., GPUs and specialised accelerators) and the availability of large-scale datasets. Deep learning generally follows neural-network architectures with multiple layers, and is therefore often referred to as deep neural networks (DNNs) [7]. Deep learning methods can operate effectively on unstructured data (e.g., images, text, and high-frequency signals) and can automatically extract hierarchical features, thereby reducing dependence on manual feature engineering and domain expertise [8]. This property makes deep learning particularly suitable for large-scale and high-complexity industrial tasks, where raw sensory streams (vision, acoustics, thermal images, multivariate time series) contain rich information but are challenging to represent with handcrafted features. In terms of practical impact, deep learning has become embedded in many industrial products and services. For instance, deep learning techniques have contributed to speech recognition systems that enable voice-based retrieval and interaction [12]. Similarly, deep learning has been used in large-scale image searching and retrieval pipelines [34]. In manufacturing, deep learning combined with machine vision has demonstrated strong performance for surface defect detection and quality inspection,

offering improvements in automation and consistency compared with manual inspection [13]. Beyond inspection, deep learning methods have also been applied to predictive maintenance and prognostics; for example, Long Short-Term Memory (LSTM) networks have been used for remaining useful life (RUL) estimation in complex equipment such as aircraft turbofan engines [14]. These developments collectively highlight that data-driven modelling is increasingly positioned not only as an analytics tool, but also as an enabling technology for closed-loop decision-making and control in industrial systems.

Despite the successes of purely data-driven methods in industrial applications, their limitations are particularly relevant to the battery and reheating-furnace systems considered in this thesis. Black-box models may require substantial training data, extrapolate poorly beyond the training distribution, and provide limited interpretability. They may also fail to respect physical constraints such as energy conservation or known input-output relationships. These concerns motivate the review of grey-box and physics-informed approaches that combine mechanistic insight with data-driven flexibility. The remainder of this chapter therefore focuses on methods and application domains most relevant to the modelling and control problems addressed in this thesis.

2.2 Physics-Informed and Grey-Box Modeling Approaches

2.2.1 Physics-Informed Neural Networks

In recent years, physics-informed neural networks (PINNs) have attracted substantial attention as an emerging framework for solving partial differential equations (PDEs) using deep neural networks (DNNs), particularly in scenarios where labelled data are limited or expensive to obtain. Unlike conventional supervised learning approaches that rely heavily on dense ground-truth measurements, PINNs embed governing equations, initial conditions, and boundary conditions into the training objective through physics-based residual losses. Moreover, compared to traditional analytical or numerical PDE

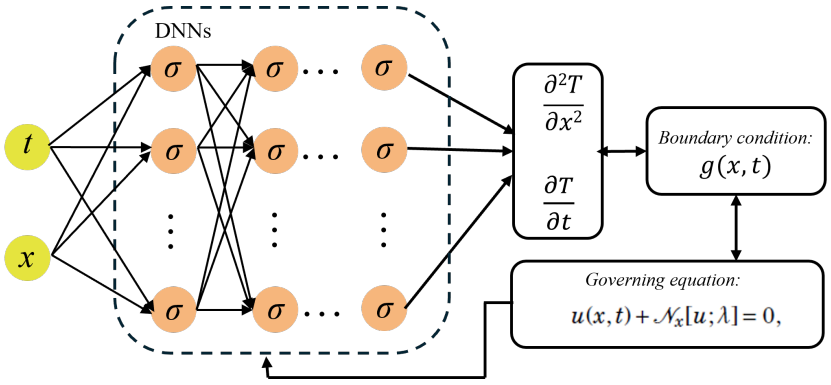


Figure 2.1: General framework of PINN model.

solvers, PINNs avoid complex discretisation schemes and iterative solution procedures, providing a unified framework that can handle irregular geometries, heterogeneous material properties, and inverse problems within the same computational pipeline. Such characteristics are highly relevant to industrial processes, where experiments are costly and full-field sensing is often impractical.

Physics-informed neural networks (PINNs) were proposed to bridge this gap by explicitly integrating governing physical laws into the learning process. Instead of relying solely on labelled data, PINNs encode the residuals of the governing PDEs, together with initial and boundary conditions, into the loss function [35]. In this framework, deep neural networks such as fully connected networks (FCNs) or convolutional neural networks (CNNs) are employed as universal function approximators, while automatic differentiation is used to efficiently compute the required spatial and temporal derivatives [36]. As a result, PINNs avoid explicit numerical discretisation and complex analytical derivations, offering a unified and flexible framework for solving both forward and inverse problems governed by PDEs.

The general formulation of a PINN can be expressed as [37]

$$u(x,t)+\mathcal{N}_x[u;\lambda]=0, \quad x\in\Omega,\ t\in[0,\tau], \tag{2.1}$$

subject to the initial condition

$$u(x,0)=h(x), \quad x\in\Omega, \tag{2.2}$$

and the boundary condition

$$u(x,t) = g(x,t), \quad x \in \partial\Omega,\ t \in [0,\tau], \tag{2.3}$$

where $\mathcal{N}_x[\cdot;\lambda]$ denotes a nonlinear operator parameterised by λ, typically represented by a deep neural network such as an FCN or CNN. Here, $u(x,t)$ denotes the target solution field, Ω and $\partial\Omega$ represent the computational domain and its boundary, respectively, $h(x)$ and $g(x,t)$ correspond to the prescribed initial and boundary conditions, and τ denotes the time horizon of interest.

Figure 2.1 illustrates the general framework of a PINN [30]. Unlike conventional neural networks trained purely on labelled datasets, the loss function of a PINN consists of multiple components, including the PDE residual loss, the initial condition loss, and the boundary condition loss. This physics-based loss formulation enables the model to learn physically admissible solutions even with sparse measurement data. For two-dimensional and higher-dimensional thermal conduction problems, convolutional neural networks can be employed as the backbone to exploit their inherent parallel computation capabilities, thereby significantly reducing computational cost. This structural flexibility further provides a foundation for extending PINN-based models towards grey-box formulations and control-oriented modelling frameworks that combine data-driven learning with known physical structures.

PINNs have emerged as a powerful mesh-free framework for solving partial differential equations by embedding physical laws into the loss function of neural networks. To address limitations of the conventional PINN approach, several variants have been developed. Bayesian PINNs (B-PINNs) incorporate Bayesian frameworks with Hamiltonian Monte Carlo or variational inference to quantify aleatoric uncertainty and improve accuracy in noisy data scenarios [38]. For problems involving interfaces and discontinuities, hard-constraint PINNs impose boundary and interface conditions exactly rather than as soft penalties, enabling rigorous satisfaction of constraints in optimal control problems [39]. Extended PINNs (XPINNs) introduce space-time domain decomposition with multiple neural networks operating on subdomains, offering enhanced parallelisation and allowing tailored network architectures for regions with varying solution complex-

ity [40]. Coupled PINNs (CPINNs) have been developed for multi-physics problems such as geothermal systems, incorporating adaptive loss weights to improve approximation capability [41]. Recent work has also explored reduced-order PINNs (ro-PINNs) that decrease derivative orders in automatic differentiation while increasing network output dimensions, demonstrating superior stability and accuracy compared to vanilla and extended-variable formulations in macroscopic flow simulations [42]. Additionally, improved PINNs (Im-PINNs) employ adaptive strategies in activation functions, loss functions, and sampling to enhance convergence for strongly nonlinear inverse problems, achieving reductions in parameter estimation errors by up to two orders of magnitude [43]. Collectively, these variants aim to mitigate optimisation pathologies, enhance accuracy and stability, and expand PINNs to more complex inverse and high-dimensional fluid mechanics problems.

A growing body of literature has demonstrated the effectiveness of PINNs in both forward and inverse problems across diverse engineering domains. For example, [44] employed PINNs to approximate feasible solutions for forward PDE problems and to identify key parameters in inverse settings across different dimensionalities, highlighting the capability of PINNs for parameter inference under physical constraints. In thermal–fluid applications, [45] applied PINNs to heat convection problems involving coupled thermal and velocity fields under unknown boundary conditions, illustrating the flexibility of PINNs in handling incomplete boundary information through physics-based regularisation. To address heterogeneous materials and discontinuous properties, [46] proposed a multi-domain PINN framework consisting of multiple sub-networks, enabling improved representation for heat conduction in multi-layer structures. Beyond classical PDE forms, [47] demonstrated PINNs for forward and inverse problems of variable-order space-fractional advection–diffusion equations, suggesting their potential in modelling anomalous transport phenomena. More recently, PINN-based modelling has been extended to complex industrial processes; for instance, [48] modelled the dehydration stage of ternary cathode material sintering using a two-sphere model combined with PINNs, improving prediction accuracy for reaction states while accounting for temperature

variation effects through simulation and experimental validation. Collectively, these studies indicate that PINNs provide a powerful framework for integrating physical laws with data-driven learning to support modelling tasks under limited measurements and uncertain process conditions.

2.2.2 Grey-Box modelling with Engineering-Gene Neural Networks

In parallel to PINNs, grey-box modelling provides a complementary route to bridge purely first-principles models and purely black-box learning. As shown in Figure 2.2 [29], grey-box approaches typically retain a mechanistic backbone (e.g., equivalent circuit models, lumped thermal networks, zone models, energy balance equations, or simplified PDE discretisations) and use data-driven components to compensate for uncertain parameters, unmodelled dynamics, or complex nonlinear mappings. This framework is attractive for industrial systems because it supports interpretability, constraint-awareness, and extrapolation robustness, while still benefiting from learning-based expressiveness. In practice, grey-box models can be realised in multiple forms: (i) *parameter learning*, where neural networks estimate time-varying or state-dependent parameters; (ii) *residual learning*, where neural networks model the mismatch between nominal physics and observed dynamics; and (iii) *structural embedding*, where engineered features or physically meaningful basis functions are injected into network architectures to guide representation learning.

A particularly noteworthy aspect of structural embedding is the eng-genes (engineering genes) architecture proposed by Li et al. [49], which explicitly incorporates domain-specific nonlinear functions extracted from first-principles laws into the neural network structure. Unlike conventional multilayer perceptrons (MLPs) that employ uniform activation functions (e.g., sigmoid, tanh, or ReLU) across all hidden nodes, the eng-genes framework allows heterogeneous activation functions derived from the fundamental physics and chemistry governing the target system.

The core philosophy of eng-genes is analogous to genetic encoding in biological sys-

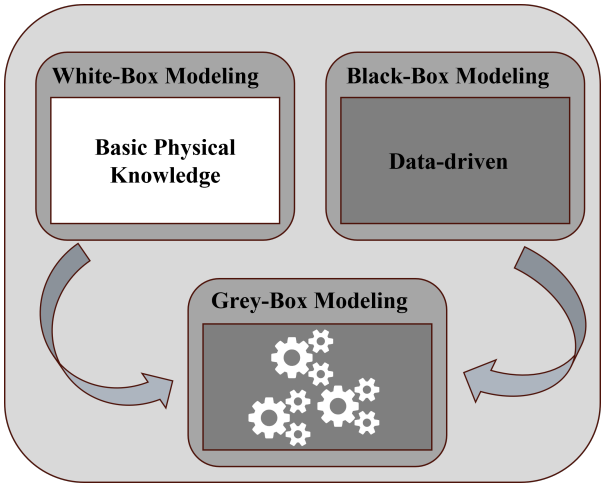


Figure 2.2: Grey box modelling

tems: just as genes determine the fundamental building blocks of organisms, engineering genes represent the essential nonlinear functional primitives that characterise industrial processes. These primitives are identified through systematic analysis of governing equations, typically expressed as systems of ordinary differential equations (ODEs) or partial differential equations (PDEs), and then embedded as activation functions within the network architecture.

Figure 2.3 [49] illustrates the complete eng-genes modelling framework, which comprises two parallel knowledge streams that converge into a unified network optimisation module. In the upper stream, grounded in *a priori* physical insight, characteristic nonlinear functions are extracted from the governing ODEs or PDEs of the target system to form the eng-gene library. In the lower stream, experimental or operational data are acquired and preprocessed to provide the training and validation sets. Both streams feed into the network optimisation module, in which an evolutionary algorithm simultaneously selects appropriate eng-genes from the library, determines the network topology, and identifies the synaptic weights and biases. A validation and falsification step is embedded within this module to assess generalisation performance and trigger structural refinement where necessary, ensuring that the final model captures the underlying physical behaviour rather than overfitting the training data.

For a multi-input single-output (MISO) system, a single-hidden-layer eng-genes

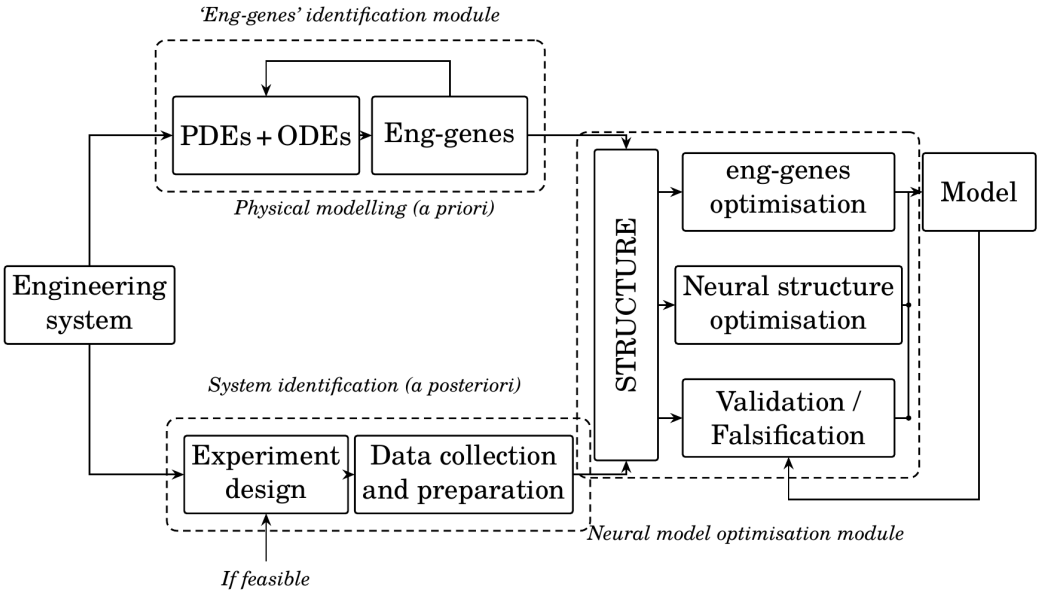


Figure 2.3: The Eng-Gene neural modelling framework.

network can be formulated as:

$$y_k = \theta_0 + \sum_{i=1}^{p} \theta_i \varphi_i(\mathbf{u}_k; \mathbf{c}_i) \tag{2.4}$$

where φ_i represents the i-th eng-gene activation function, $\mathbf{u}_k$ denotes the input vector, $\mathbf{c}_i$ encapsulates the associated parameters (weights and biases) of the i-th neuron, and θ_i are the output layer weights. Crucially, the activation functions φ_i are not predetermined standard functions but are instead selected from a library of physically meaningful nonlinearities:

$$\varphi_i(\mathbf{u}_k; \mathbf{c}_i) = \begin{cases} \psi_1(\mathbf{u}_k; \mathbf{c}_i) & \text{if } 0 \le i < m_1 \\ \psi_2(\mathbf{u}_k; \mathbf{c}_i) & \text{if } m_1 \le i < m_2 \\ \vdots & \vdots \\ \psi_q(\mathbf{u}_k; \mathbf{c}_i) & \text{if } m_{q-1} \le i < m_q = p \end{cases} \tag{2.5}$$

where ψ_j represents distinct classes of eng-genes (e.g., Arrhenius-type exponentials, rational functions, power laws), and $(m_j - m_{j-1})$ indicates the number of hidden neurons employing the j-th eng-gene type.

The eng-genes modelling workflow consists of five systematic steps:

1. **Physical mechanism establishment**: Formulate the fundamental governing laws of the system, typically as PDEs or ODEs describing mass, energy, and momentum conservation, along with reaction kinetics or thermodynamic relationships.

2. **Eng-gene extraction**: Identify characteristic static nonlinear functions that recur in the governing equations. For instance, in combustion systems, Arrhenius kinetics yield exponential functions of the form $\varphi_1(x) = e^{-E_a/(RT)}$, while species concentration dependencies may produce rational functions $\varphi_2(x) = c_1/(c_2 + bx)$.

3. **Data acquisition and preprocessing**: Collect experimental or simulation data spanning the operational envelope of interest, ensuring sufficient excitation of system nonlinearities.

4. **Network optimisation**: Employ evolutionary algorithms (typically genetic algorithms with mixed coding schemes) to simultaneously optimise: (a) network topology (selection of eng-genes and hidden layer size), (b) synaptic weights, and (c) eng-gene parameters.

5. **Model validation**: Assess generalisation performance on unseen test data to ensure the learned model captures underlying physics rather than overfitting training patterns.

Grey-box modelling has been widely explored in battery energy systems, where electro–thermal behaviour is strongly coupled and subject to ageing-dependent parameter drift. A common strategy is to adopt an interpretable electrical model (e.g., equivalent circuit models) and augment it with learned mappings for temperature-dependent resistance/capacitance, hysteresis, or unmodelled diffusion effects, thereby improving accuracy without losing physical interpretability required for battery management applications. Similarly, for battery thermal modelling, lumped thermal networks provide a tractable physics structure, while data-driven modules can capture nonlinear heat generation and convection variations under different operating conditions. In industrial thermal processes such as reheating furnaces, reduced-order physics models (e.g., zone

models or discretised heat-transfer representations) provide interpretable structure for energy and temperature evolution, while learning-based terms can compensate for uncertainties in radiative exchange, boundary disturbances, material property variations, and production scheduling effects. In these settings, grey-box methods often deliver improved predictive fidelity and robustness compared with purely mechanistic models, while being more trustworthy and controllable than fully black-box neural surrogates.

Overall, both PINNs and grey-box modelling represent important directions toward physics-consistent and data-efficient modelling for industrial-scale systems. PINNs offer a general learning-based PDE-solving framework, yet face efficiency and scalability challenges in large-scale industrial scenarios. Grey-box approaches offer strong interpretability and deployment compatibility, but require systematic design to effectively integrate learning modules without compromising stability or physical plausibility. These considerations motivate the development of control-oriented and deployable physics-informed data-driven models, which form an important methodological basis for the subsequent chapters of this thesis.

2.3 Learning-Enhanced Predictive Control

2.3.1 Foundations of Predictive and Iterative Learning Control

Model Predictive Control (MPC) has established itself as one of the most prominent advanced control strategies in modern engineering, owing to its systematic handling of multi-variable systems, explicit constraint satisfaction, and receding horizon optimisation mechanism. The fundamental principle of MPC lies in solving a finite-horizon optimal control problem at each sampling instant, applying the first element of the resulting optimal input sequence, and then repeating this process with updated state measurements. This receding horizon strategy endows MPC with inherent feedback properties and robustness against model-plant mismatches. Considerable research efforts have been devoted to the theoretical foundations and practical deployment of MPC across diverse engineering domains. Drgoňa et al. [50] conducted a comprehensive investigation

into the application of MPC for building energy management, systematically addressing modelling methodologies, simulation frameworks, and control algorithm design, thereby establishing a valuable reference architecture for energy-oriented MPC applications. In the domain of power electronics, Karamanakos et al. [51] examined both direct and indirect MPC formulations, providing rigorous analyses of closed-loop stability guarantees and the associated computational complexity, which remains a persistent challenge in real-time embedded implementations.

Despite its widespread adoption, conventional MPC suffers from a fundamental dependence on the fidelity of the underlying predictive model. Inaccuracies in system identification, unmodeled dynamics, and time-varying parameters can substantially degrade closed-loop performance and even compromise stability. This limitation has motivated significant research into sampling-based MPC variants, which relax the requirement for gradient-based optimisation by employing stochastic sampling to approximate the optimal control sequence. Methods such as Model Predictive Path Integral (MPPI) control leverage importance-weighted Monte Carlo sampling to evaluate a large ensemble of candidate trajectories in parallel, enabling gradient-free optimisation that is naturally amenable to GPU acceleration [52]. These sampling-based approaches exhibit improved robustness to non-convex cost landscapes and discontinuous dynamics, though they introduce their own trade-offs in terms of sample efficiency and variance of the resulting control policy. Iterative Learning Control (ILC) offers a complementary framework to MPC, motivated by the observation that many industrial processes execute repetitive tasks under similar initial conditions. ILC systematically exploits the error information accumulated over previous task repetitions to iteratively refine the feedforward control input, progressively approaching the desired trajectory tracking performance. The convergence properties of ILC algorithms, including both P-type and D-type update laws as well as norm-optimal formulations, have been extensively studied in the literature [53]. However, a fundamental limitation of conventional ILC lies in its batch nature: the control update is computed between trials rather than within a single execution, which precludes real-time adaptation to mid-trial disturbances or changes in

operating conditions. Furthermore, ILC relies on the assumption of repeatable dynamics and consistent initial conditions across trials, assumptions that are frequently violated in practical engineering scenarios involving parameter drift or varying load profiles. The integration of ILC principles with predictive control frameworks has therefore attracted growing attention as a means of combining the trial-to-trial learning capability of ILC with the real-time optimisation and constraint-handling strengths of MPC.

2.3.2 Learning-Based and Data-Driven Formulations of Model Predictive Control

The fusion of data-driven methodologies with the MPC framework has emerged as an active and rapidly evolving research direction, motivated by the practical difficulties of constructing accurate first-principles models for complex systems. Data-driven MPC approaches broadly fall into two categories: those that employ historical data to construct or refine the predictive model used within a conventional MPC optimisation, and those that derive the control policy or system representation entirely from data without explicit physical modelling.

Within the model-construction category, system identification techniques such as subspace methods, Gaussian process regression, and neural network-based surrogate models have been widely integrated into MPC pipelines. Gaussian process MPC is particularly notable in this regard, as it provides not only a data-driven predictive model but also principled uncertainty quantification in the form of posterior variance estimates, which can be propagated through the prediction horizon to yield probabilistic constraint satisfaction guarantees. The resulting stochastic MPC formulations account for both aleatoric uncertainty stemming from process noise and epistemic uncertainty arising from limited training data, offering a more principled treatment of model uncertainty than deterministic tube-based approaches. In the direct data-driven category, the Willems' fundamental lemma has provided a powerful theoretical underpinning for non-parametric data-driven control [54]. This result establishes that the input-output behaviour of a linear time-invariant system can be fully characterised by a Hankel matrix constructed from a single

persistently exciting trajectory, eliminating the need for explicit system identification. Building upon this foundation, Data-enabled Predictive Control (DeePC) directly embeds the Hankel matrix into the MPC optimisation, replacing the model-based prediction with a data-driven behavioural representation. Persis et al. [55] further advanced this line of research by employing input-output data to formulate a data-dependent system matrix, thereby circumventing the intricate processes of physical modelling and parameter identification. Extensions of these frameworks to handle nonlinear systems, output feedback settings, and closed-loop data collection scenarios remain active areas of investigation. Rueda-Escobedo et al. [56] developed a data-driven control algorithm specifically targeting discrete time-delay systems, wherein historical data information is systematically incorporated into both the model development phase and the state feedback gain design, demonstrating the potential of data-centric approaches in handling systems with complex dynamic structures.

The practical deployment of data-driven MPC in engineering applications has also been actively explored. In the context of building climate control and energy systems, data-driven predictive models derived from operational data have demonstrated competitive performance relative to physics-based MPC, with the additional advantage of automatic adaptation to seasonal variations and occupancy pattern changes. For battery management and electrochemical systems, data-driven MPC approaches offer a compelling alternative to equivalent circuit or physics-based electrochemical models, which typically require extensive parameterisation and are sensitive to ageing-induced parameter drift. In repetitive batch processes, Zhou et al. [57] demonstrated how historical trajectories can be used for data-efficient constrained learning and value-function approximation, providing a relevant foundation for the historical-data mechanism developed in Chapter 4.

2.3.3 Reinforcement Learning for Control

Reinforcement Learning (RL) has attracted considerable attention as a model-free approach to deriving optimal control policies through direct interaction with the environ-

ment. The RL framework frames the control problem as a Markov Decision Process, wherein an agent learns to select actions that maximise a cumulative discounted reward signal through repeated trial-and-error exploration. This framework is particularly attractive for complex, high-dimensional control problems where deriving analytical optimal policies is intractable, and where the system dynamics are either unknown or too complex to model accurately.

Deep reinforcement learning algorithms, which leverage deep neural networks as function approximators for the policy and value functions, have demonstrated remarkable empirical success across a broad range of control tasks. Deep Deterministic Policy Gradient (DDPG) [58], Proximal Policy Optimization (PPO) [59], Soft Actor-Critic (SAC) [60], and Twin Delayed DDPG (TD3) [61] represent prominent actor-critic architectures that have been applied to continuous-action control problems in robotics, process control, and energy management [62]. The soft actor-critic algorithm, in particular, incorporates a maximum entropy objective that encourages stochastic exploration and has demonstrated superior sample efficiency and robustness compared to earlier deterministic policy gradient methods. Despite these empirical successes, RL-based control strategies face several fundamental challenges that hinder their deployment in safety-critical engineering applications. First, the sample inefficiency of model-free RL is a significant concern: learning a satisfactory control policy may require millions of environment interactions, which is impractical when real-world trials are costly, time-consuming, or potentially hazardous. Second, the exploration strategies employed during training may generate unsafe actions that violate physical or operational constraints, posing risks to equipment integrity and process safety. Third, the convergence behaviour and closed-loop stability properties of deep RL policies lack the formal guarantees that are expected in industrial control applications. Fourth, the black-box nature of neural network policies renders interpretability and systematic tuning difficult for control engineers accustomed to classical and model-based design frameworks. Model-based reinforcement learning (MBRL) attempts to address the sample efficiency limitation by learning a dynamics model from collected data and using this model to generate synthetic experience for policy

optimisation or to perform planning. Methods such as Dyna-style algorithms interleave real and simulated experience [63], while Model Predictive Control with learned models effectively bridges MBRL and data-driven MPC. The integration of RL with MPC frameworks, sometimes referred to as RL-MPC or learning-based MPC, is gaining traction as a means of combining the asymptotic optimality and adaptability of RL with the constraint satisfaction and interpretability of MPC [64]. In these hybrid architectures, RL may be employed to learn the terminal value function or terminal set for the MPC optimisation, to tune the cost function weights online, or to provide a warm-start solution that reduces the online computational burden of the MPC solver.

In the context of battery charging control and similar electrochemical process applications, RL-based approaches face additional challenges related to the slow timescales of degradation dynamics, the difficulty of defining reward functions that adequately capture long-term ageing costs, and the limited availability of diverse training data spanning the full operating envelope. These considerations motivate the development of alternative learning-enhanced control strategies that retain the predictive optimisation structure of MPC while incorporating historical operational data in a more structured and sample-efficient manner, as pursued in the present work.

2.4 Thermal modelling and Control in Steel Industrial Reheating Furnaces

Reheating furnaces occupy a central position in the steelmaking production chain, serving as the thermal processing stage through which intermediate steel products, including slabs, blooms, and billets, are brought to a target discharge temperature, typically in the range of 1100–1250°C, with permissible temperature non-uniformity of no more than 20–30°C across the cross-section, prior to hot forming or rolling operations [65]. The reheating process is energy-intensive by nature: large continuous reheating furnaces with heat inputs up to 200 MW are commonplace in modern steel plants, and the average net energy consumption for such facilities burning natural gas exceeds $1.2GJ$

per tonne of steel produced [66]. Beyond energy concerns, inadequate temperature control results in product defects, increased rejection rates, and operational disruptions during downstream rolling processes. It has been estimated that reducing product rejection rates from 1.5% to 0.2% alone could yield energy savings of approximately 9%, equivalent to roughly $0.3GJ$ per tonne [67]. These economic and environmental imperatives have driven sustained research interest in developing accurate thermal models and effective control strategies for reheating furnaces over several decades.

This section reviews the principal approaches to thermal modelling of reheating furnaces and their associated control strategies, drawing upon both classical methods grounded in heat transfer physics and more recent developments in machine learning and physics-informed deep learning.

2.4.1 Heat Conduction modelling within Steel Stock

Accurate prediction of the internal temperature distribution within the steel stock is indispensable for real-time product quality monitoring and adaptive control of the discharge temperature. Surface temperature measurements obtained through radiometric imaging cameras or thermocouples provide only boundary information and cannot directly reveal the through-thickness temperature gradient, which is critical for ensuring the required temperature uniformity prior to rolling. Figure 2.4 illustrates a typical walking-beam reheating furnace and the intermediate steel stock, showing the spatial arrangement that governs the heat transfer environment addressed in this section [68].

2.4.1.1 Conventional Numerical and Analytical Methods

The accurate prediction of transient temperature distribution within steel stock during the reheating process has been a longstanding research challenge, and a substantial body of literature has addressed this problem through classical numerical and analytical solution methods. These approaches seek to solve the governing Fourier heat conduction equation, or its discretised approximation, subject to appropriate initial and boundary conditions representing the thermal environment imposed by the furnace.

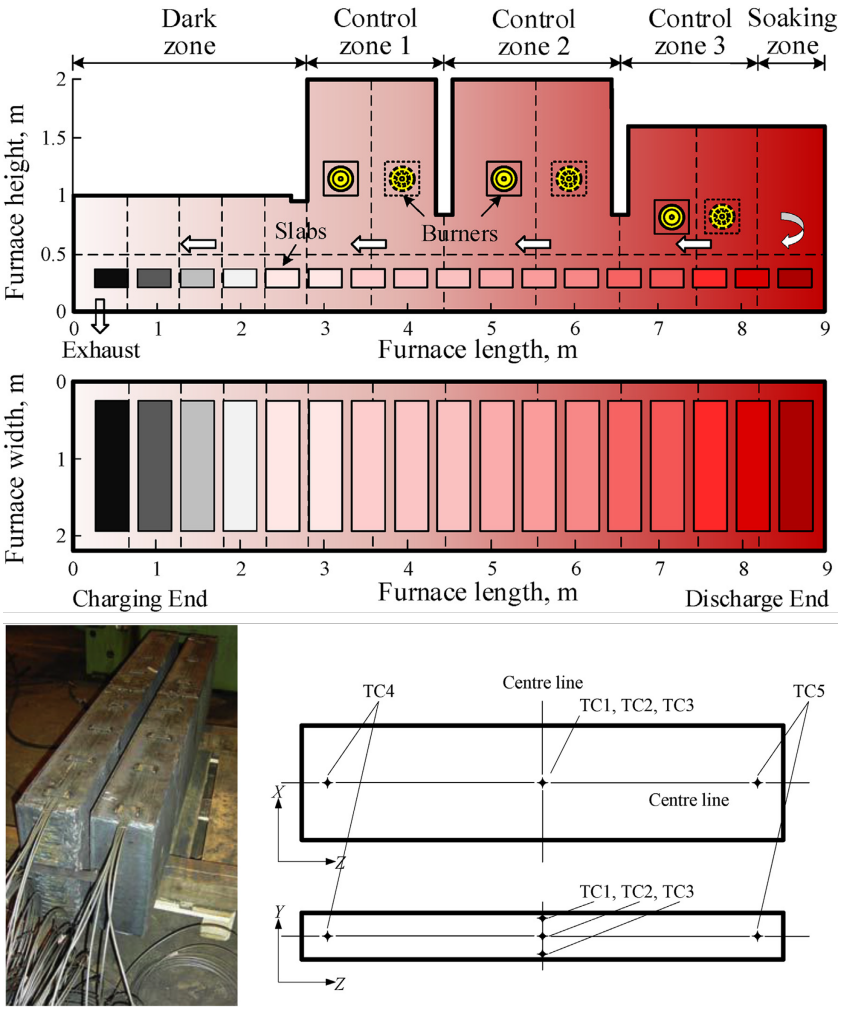


Figure 2.4: The reheating furnace and the steel stock

Analytical solutions to the transient heat conduction equation exist for simplified geometries and boundary conditions. For one-dimensional problems with constant material properties and idealised boundary conditions, series solutions based on separation of variables and eigenfunction expansion have been well established since the foundational work of Carslaw and Jaeger [69]. Such solutions offer valuable physical insight into the characteristic timescales and spatial profiles of thermal diffusion within the stock and have been used in early furnace control systems to estimate bloom core temperatures from surface measurements. However, their applicability is fundamentally constrained by the requirements of geometric regularity, spatially uniform boundary conditions, and critically temperature-independent material properties. In practice, the thermal

conductivity and specific heat capacity of steel exhibit strong and nonlinear temperature dependence across the heating range encountered in reheating furnaces, rendering constant-property analytical solutions insufficiently accurate for quantitative control applications.

To accommodate temperature-dependent material properties and arbitrary boundary conditions, numerical discretisation methods have been widely adopted. The Finite Difference Method (FDM) provides a straightforward and computationally efficient framework for discretising the heat conduction PDEs on a structured grid. Among the various time integration schemes applicable within the FDM framework, the Crank-Nicolson method introduced by Crank and Nicolson [70] has been particularly widely adopted in furnace modelling applications, owing to its unconditional stability and second-order temporal accuracy, which allow the use of relatively large time steps without numerical divergence. The Finite Element Method (FEM) has also been applied to steel heat conduction problems [71], offering greater geometric flexibility and the ability to handle irregular stock cross-sections, though at higher implementation complexity and computational cost compared with FDM on structured grids.

For two-dimensional transient heat conduction problems, which are necessary to resolve the through-thickness and through-width temperature gradients in bloom and slab cross-sections, the direct application of implicit finite difference schemes leads to large coupled linear systems that must be solved at every time step, which can be computationally demanding. The Alternating Direction Implicit (ADI) method addresses this challenge by splitting the 2D problem into two sequential sets of one-dimensional tridiagonal systems, solved alternately along the x and y spatial directions within each time step [72]. The resulting tridiagonal systems can be solved efficiently using the Thomas algorithm, achieving unconditional stability while dramatically reducing the computational cost relative to fully implicit 2D solvers. The ADI method has been widely applied to heat conduction modelling in steel processing [73–75]; Croft et al. provided an early and influential treatment of ADI-based heat transfer calculations using finite difference equations that has been extensively referenced in subsequent furnace modelling

literature [72]. More recent implementations have incorporated temperature-dependent polynomial representations of thermal conductivity and specific heat capacity, calibrated against experimental measurements for specific steel grades, enabling more accurate simulation of the nonlinear thermal behaviour encountered in industrial reheating [65, 76].

Beyond FDM and FEM, the Finite Volume Method (FVM) has also been applied to steel heat conduction problems, particularly in configurations where it is desirable to ensure strict local energy conservation at the discrete level. FVM formulations discretise the integral form of the energy conservation equation over control volumes surrounding each mesh node, offering a natural connection to the energy balance perspective adopted in zone-based furnace models and ensuring that the numerical scheme respects the fundamental conservation laws of thermodynamics by construction.

In the specific context of reheating furnace modelling, numerical conduction models are typically coupled to a furnace-level radiation and energy balance model, such as the zone method, which supplies the surface heat flux boundary conditions at each time step. This hierarchical coupling, in which the zone model resolves the furnace enclosure thermal field and the conduction model resolves the internal stock temperature distribution, has been demonstrated to yield bloom temperature predictions in close agreement with plant trial data for both static and dynamic operating conditions [30, 77]. The surface heat flux boundary condition incorporates both radiative and convective contributions from the furnace gas and wall zones, and its accurate specification is essential for realistic conduction modelling.

Despite the maturity and physical rigour of conventional numerical methods, their application in real-time furnace control is constrained by computational cost. For a fine spatial mesh resolving the through-thickness temperature distribution of a large bloom, the repeated solution of tridiagonal systems, or their higher-dimensional equivalents, at every time step across all blooms simultaneously present in the furnace can impose significant computational latency. This challenge is compounded in dynamic operating scenarios where bloom dimensions or material grades change between production

batches, necessitating reconfiguration of the mesh and material property functions. Furthermore, the sequential, time-step-by-time-step nature of numerical PDE solvers limits opportunities for the parallelisation that modern computing hardware increasingly offers through GPU acceleration. These considerations have motivated growing interest in surrogate modelling approaches that seek to replicate the predictive accuracy of numerical conduction models at substantially reduced online inference cost.

2.4.1.2 Machine Learning and Physics-Informed Approaches

The inherent computational burden associated with conventional numerical methods for solving transient heat conduction PDEs has motivated growing interest in data-driven and hybrid modelling approaches as surrogate alternatives. With the rapid advancement of deep learning in the past decade, neural network-based models have emerged as promising candidates for approximating the solutions to complex governing equations, offering the potential for substantially reduced online inference time once the model has been trained offline.

Early data-driven efforts applied relatively shallow machine learning models, including support vector regression and feedforward neural networks, to predict surface or average temperatures of steel stock as functions of process inputs, treating the heat conduction process as a black-box input-output mapping [78, 79]. While such approaches demonstrated feasibility in controlled experimental settings, they suffered from poor generalisation to operating conditions outside the training distribution and lacked any mechanism to enforce physical consistency in their predictions. The absence of physics constraints also meant that such models required large volumes of labelled experimental data, which is notoriously difficult and costly to acquire in industrial reheating environments where instrumentation access is limited.

The application of Convolutional Neural Networks (CNNs) to spatially distributed thermal fields represented a significant methodological advance. The spatially structured nature of a 2D temperature distribution, which is analogous to an image, makes CNN architectures naturally well-suited to feature extraction and spatial pattern learning in this

context. Fonda et al. applied a UNet-based CNN model to study heat transfer in Rayleigh-Bénard convection, demonstrating that deep learning models could capture the salient spatial structures of temperature fields under varying Rayleigh numbers [80]. Yuan et al. proposed a dynamic CNN strategy for soft sensor modelling of industrial process data, illustrating the broader applicability of convolutional architectures to temporal-spatial feature extraction in manufacturing processes [81]. These works collectively established CNN-based models as effective tools for learning from high-dimensional spatial data in heat transfer and industrial process contexts.

Despite their expressive power, purely data-driven deep learning models are widely recognised as "black-box" models whose predictive accuracy depends critically on the quantity and quality of available labelled data, and whose generalisation to unseen conditions can be unreliable [82]. These limitations are particularly acute in the context of industrial steel reheating, where the cost of conducting controlled experiments under diverse operating conditions is prohibitive, and where the consequences of prediction errors, in the form of product defects or energy waste, are significant. Furthermore, without physical constraints, black-box models may produce predictions that violate fundamental conservation laws, undermining their trustworthiness in safety-critical process control applications.

Physics-Informed Neural Networks (PINNs) offer a principled approach to addressing these limitations by embedding the governing PDEs directly into the neural network training process through the loss function formulation. Rather than learning from labelled input-output pairs alone, PINNs are trained to satisfy both data observations and the physical constraints expressed by the governing equations, initial conditions, and boundary conditions. This approach reduces reliance on large labelled datasets and promotes physically consistent extrapolation. The PINN framework has been applied to a range of heat transfer problems. For instance, Cai et al. demonstrated the application of PINNs to heat convection problems in thermal and velocity fields under unknown boundary conditions [45]. Mao et al. used PINNs to solve high-speed compressible flow problems [44]. Zhang et al. developed a multi-domain PINN formulation to address

transient heat conduction in multi-layer materials [83]. These studies confirmed the broad applicability of physics-informed learning to engineering heat transfer problems and its advantages in data-scarce regimes.

In the specific context of steel reheating furnaces, Wang et al. developed a mathematical model using the zonal method to study heat conduction within beam blanks, highlighting the challenge of predicting the internal temperature distribution in complex stock geometries [84]. Ahmed et al. constructed a steady-state model approximating the furnace environment for slab temperature prediction [85]. However, neither approach was formulated for real-time inference at the computational speeds demanded by closed-loop control systems. The application of PINNs to the transient heat conduction problem within steel stock, particularly in a form compatible with the real-time control loop of a reheating furnace, thus represents an active and important research frontier.

Conventional PINN implementations employing fully connected networks (FCNs) as the backbone face limitations in scalability and computational efficiency when applied to large-scale, high-dimensional spatial problems. The per-iteration cost of computing PDE residuals through automatic differentiation in FCN-based PINNs grows with problem dimension, and the sequential nature of FCN computation does not naturally exploit the spatial regularity of the temperature field. CNN-based architectures, by contrast, leverage parallelised convolution operations that align naturally with the grid-structured spatial domain of heat conduction problems, offering significant computational advantages for 2D and higher-dimensional temperature field prediction.

Taken together, the trajectory of research in this area reveals a clear progression from purely data-driven black-box models, through physics-informed FCN-based PINNs, toward physics-informed CNN architectures that combine spatial inductive biases with principled physical constraints and domain knowledge embedding. Key open challenges include extending these approaches to handle strongly non-uniform heating profiles and multi-material configurations, developing robust uncertainty quantification frameworks for safety-critical process control contexts, and achieving seamless online integration of trained surrogate models within the full dynamic zone model and control architecture of

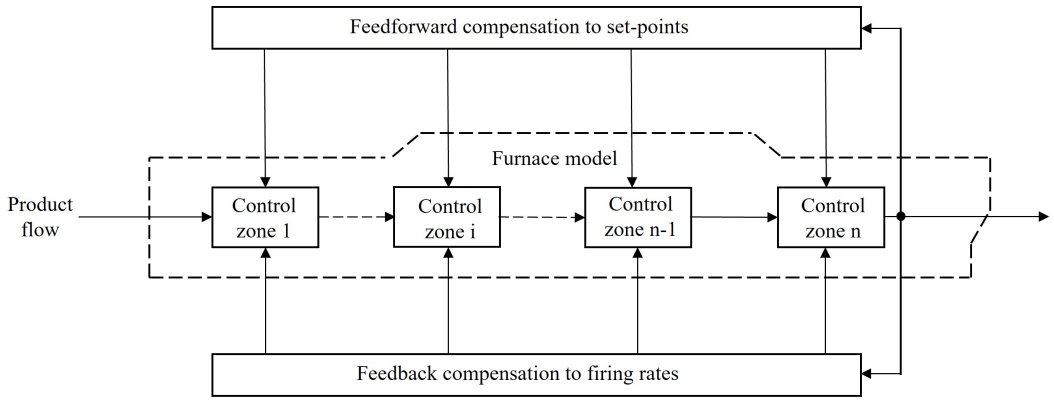


Figure 2.5: Control zones

industrial reheating furnaces.

2.4.2 Furnace Control Strategies

Supervisory control systems for reheating furnaces are conventionally structured at two levels. At Level 1, the burner firing rate is modulated through either amplitude control by varying valve opening to adjust heat generation intensity, or frequency control, in which burners are pulsed on and off at varying duty cycles. At Level 2, a mathematical furnace model computes the heat transfer rate to the stock, informing the temperature set-points for each control zone. Figure 2.5 illustrates the typical zone layout of an industrial reheating furnace control system [30].

The most widely deployed feedback control strategy in industrial reheating furnaces is the proportional-integral-derivative (PID) controller, which computes a corrective temperature compensation signal based on the error between the measured zone temperature and the set-point [65]. The controller output is normalised and used to modulate the burner fuel flow rate within prescribed minimum and maximum firing rate bounds. While PID control provides reliable steady-state performance under fixed operating conditions, its effectiveness is inherently limited when confronted with the large inertia, significant hysteresis, and measurable disturbances characteristic of dynamic furnace operation, including batch variations, changes in stock dimensions and material properties, and production delays.

A further and often underappreciated limitation of conventional furnace control lies in its reliance on wall or gas zone temperature measurements as proxies for stock thermal state. In practice, the internal temperature distribution within the steel stock, which governs product quality, temperature uniformity, and rolling readiness, cannot be directly measured during furnace operation. Radiometric imaging cameras and surface thermocouples provide only boundary observations, leaving the through-thickness temperature gradient unobserved. As a consequence, conventional PID-based control loops regulate furnace zone temperatures rather than the stock temperature distribution itself, creating a fundamental disconnect between the controlled variable and the quantity of actual engineering interest. This observability gap means that transient phenomena such as the thermal lag between furnace atmosphere and stock core, or the temperature non-uniformity induced by water-cooled skid marks, are not directly compensated by the control action. Bridging this gap requires the integration of a real-time stock temperature prediction model within the control loop, capable of estimating the full internal temperature field at every control step based on available boundary measurements and the current furnace operating state.

Motivated by both the limitations of standard PID control and the need for real-time stock temperature state estimation, an adaptive predictive control strategy has been developed and validated on a PINN-based simulation platform [30]. This strategy recognises that stocks pass through the furnace in a sequential, unidirectional manner, so that the interaction between control zones is essentially cascaded, with each zone primarily influencing the thermal state of the subsequent zone. The control architecture consequently comprises two coupled loops: a feedforward loop, which employs a PID controller to adjust the temperature set-point of each control zone in response to measurable disturbances, and a feedback loop, which employs a normalised PID controller to modulate the burner firing rate to track the adjusted set-point and compensate for unmeasurable disturbances. The defining feature of this adaptive strategy is that the temperature set-points are not prescribed from historical data but are generated dynamically by the zone model itself through predictive simulation. Specifically, the feedforward loop

compares the predicted discharge temperature at the next time step, computed through a one-step-ahead simulation of the zone model, against the target discharge temperature. If the predicted discharge temperature deviates from the target, the set-points of all heating zones are proportionally adjusted to correct the future temperature trajectory. Crucially, this predictive mechanism depends on an accurate and timely estimate of the current internal stock temperature distribution, since the zone model uses this as the initial condition for its one-step-ahead prediction. The fidelity of the stock temperature estimate therefore directly determines the accuracy of the feedforward compensation and, by extension, the quality of discharge temperature regulation. This dependency places a stringent requirement on the computational efficiency of the stock temperature prediction model: it must complete its inference within the control time step, typically on the order of seconds in industrial reheating furnaces, to avoid introducing latency that degrades closed-loop performance.

It is further noted that the PID controllers within this adaptive framework may be substituted with more advanced controllers such as Model Predictive Control (MPC), depending on the performance requirements of the specific application, without altering the fundamental two-loop control architecture. Regardless of the choice of controller, however, the real-time availability of accurate stock internal temperature estimates remains a prerequisite for the effective operation of any model-based furnace control strategy, underlining the central importance of computationally efficient heat conduction modelling to the broader furnace control problem.

2.5 Modelling and Control in Battery Energy Systems

Lithium-ion batteries (LIBs) have emerged as the dominant energy storage technology for electric vehicles (EVs) and grid-scale renewable energy integration, owing to their high energy density, long cycle life, and declining cost [86]. However, the efficient and safe operation of LIBs remains a significant engineering challenge. Accurate modelling of battery states, including state of charge (SOC), state of health (SOH), and temperature,

is a prerequisite for designing effective battery management systems (BMS), which are responsible for preventing overcharging, over-discharging, and thermal runaway [87]. In parallel, the control strategies governing charging and discharging behaviour critically determine battery performance, longevity, and safety. This section provides a structured review of the major battery modelling approaches and control strategies reported in the literature, with particular emphasis on the growing role of artificial intelligence (AI) and data-driven methods in advancing BMS capability.

2.5.1 Battery Modelling Approaches

Battery modelling methods can be broadly categorised into physics-based models, data-driven models, and hybrid grey-box models. Each framework presents distinct trade-offs between physical interpretability, computational complexity, and predictive accuracy.

2.5.1.1 Physics-Based Models

Physics-based electrochemical models, such as the pseudo-two-dimensional (P2D) model and the single particle model (SPM), are derived from first-principle descriptions of the internal electrochemical reactions within the battery cell [88, 89]. These models offer high physical fidelity and are capable of accurately capturing battery dynamics under a broad range of operating conditions. Nevertheless, they involve numerous coupled partial differential equations, resulting in substantial computational complexity that hinders their real-time deployment in embedded BMS hardware [90]. Equivalent circuit models (ECMs), such as the Thevenin model and the Rint model, simplify the battery representation using common electrical components (resistors and capacitors) to approximate charge/discharge behaviour [91, 92]. Although ECMs are structurally simple and computationally efficient, they do not adequately represent the battery's internal electrochemical dynamics, which limits their accuracy — particularly under highly dynamic load profiles or across widely varying temperature conditions [22]. Figure 2.6 illustrates representative ECM configurations commonly employed in BMS applications.

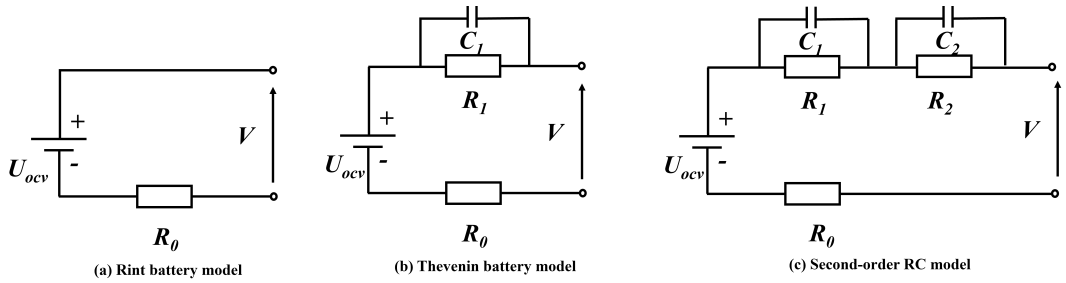


Figure 2.6: Battery equivalent circuit models

2.5.1.2 Neural Network-Based Models

The limitations of conventional physics-based approaches have motivated substantial interest in data-driven modelling, particularly through artificial neural networks (ANNs) and their deep learning variants. Neural networks possess strong nonlinear approximation capabilities and can learn the input–output relationship of complex battery systems directly from operational data, without requiring explicit physical equations [6]. Wang et al. proposed a thermal-coupled equivalent circuit model integrating an ANN to correlate battery temperature with SOC, validating its generalisation under both constant current and pulse current discharge profiles [93]. Given the temporal dynamics inherent in battery operation, recurrent architectures — specifically long short-term memory (LSTM) networks and recurrent neural networks (RNNs) — have been widely adopted for battery state estimation. Deng et al. applied LSTM to develop an SOH estimation model incorporating multiple input–output mapping structures, achieving significant improvements in estimation accuracy [94]. Zhang et al. combined LSTM with particle filtering (PF) to refine SOC and SOH predictions, demonstrating superior performance compared to traditional electrochemical models [95]. Furthermore, convolutional neural networks (CNNs) have been applied to battery capacity estimation. Li et al. introduced a fast recursive pruning algorithm that reduced computational complexity by approximately 80% while preserving estimation accuracy [96]. Despite these advances, neural network models remain heavily dependent on the volume and quality of training data, and their lack of physical interpretability raises concerns regarding reliability in safety-critical BMS applications.

2.5.1.3 Statistical and Probabilistic Models

Statistical modelling methods provide a principled framework for quantifying prediction uncertainty, which is particularly valuable for SOH estimation and degradation prognosis. Gaussian process regression (GPR) has attracted considerable attention due to its suitability for small datasets, its capacity for nonlinear regression, and its inherent uncertainty quantification. Yang et al. developed a novel GPR model using four charging-curve-derived features as inputs, exploiting their strong correlation with battery ageing [97]. Sun et al. employed a multi-island genetic algorithm in conjunction with GPR for SOH estimation, leveraging grey relational analysis to identify meaningful degradation features [98]. Lin et al. proposed a multi-kernel GPR framework in which the kernel function combined squared exponential and cyclic covariance terms, with hyperparameters optimised via particle swarm optimisation (PSO) to improve estimation stability [99]. Particle filter (PF) methods, grounded in sequential Bayesian estimation, have also been extensively applied to joint state and parameter estimation. He et al. formulated a dynamic Bayesian network (DBN) for online SOH estimation, treating SOC as a hidden state and terminal voltage as the observed variable [100]. Schwunk et al. used Monte Carlo sampling within a particle filter framework to estimate both SOC and state of power (SOP) for lithium iron phosphate cells [101]. A gradient-corrected particle filter was proposed by Tang et al. to track the battery SOH trajectory, with model regularisation applied to reduce sensitivity to local fluctuations in the ageing curve [102]. Although statistical models are well-suited to periodic and degradation-pattern data, they generally require access to full or partial charge/discharge cycle measurements, which may be unavailable in certain operational scenarios.

2.5.1.4 Grey-Box Hybrid Models

To reconcile the interpretability of physics-based models with the flexibility of data-driven approaches, grey-box (hybrid) modelling frameworks have been proposed. The fundamental concept is to use a physics-based model to capture the dominant battery behaviour, while a data-driven component compensates for residual errors and local

variations that the empirical model fails to represent [103]. Zhang et al. introduced a hybrid SOH estimation method in which an empirical degradation model characterised the overall capacity fade trajectory, while least squares support vector regression (LS-SVR) accounted for localised deviations [104]. Han et al. developed a semi-empirical capacity degradation model incorporating battery internal resistance and temperature, with genetic algorithm (GA)-based parameter identification and particle filter-based online parameter updating, enabling robust health management under real operating conditions [103]. Grey-box models offer improved accuracy and generalisation compared to purely data-driven models, while maintaining greater computational tractability than full electrochemical models. However, the design of the hybrid architecture, particularly the coupling interface between physical and learned components, can be complex and time-intensive.

2.5.2 Control Strategies in Battery Management Systems

Beyond state estimation, the control strategies governing battery charging, discharging, and pack balancing are equally critical to BMS performance. Ineffective control can accelerate battery degradation, compromise safety, or fail to meet user requirements such as fast charging. The literature identifies three broad categories: open-loop, closed-loop, and data-driven control strategies.

2.5.2.1 Open-Loop Charging Strategies

Classical open-loop battery charging strategies, namely constant current (CC), constant voltage (CV), and combined CC-CV, remain widely deployed due to their straightforward implementation and minimal parameter requirements [105]. CC charging maintains a fixed current throughout the charging process; however, it may inflict stress on the battery towards the end of the charge cycle. CV charging holds the terminal voltage constant, causing a gradual reduction in current as the battery approaches full charge, but it imposes large initial currents that can degrade battery health. The CC-CV strategy mitigates the respective drawbacks of both approaches by sequentially applying CC

followed by CV [106]. While these strategies are reliable and easy to implement in commercial BMS, their inherent open-loop structure precludes adaptation to dynamic battery states or varying operational environments, limiting their robustness, particularly in battery pack applications involving cells with heterogeneous characteristics.

2.5.2.2 Closed-Loop Control Strategies

Closed-loop control strategies incorporate real-time feedback of battery states, enabling more sophisticated management objectives such as thermal regulation and pack balancing. The proportional-integral-derivative (PID) controller is the most widely adopted closed-loop approach in industrial systems, owing to its structural simplicity and well-established tuning methodologies. Kaleem et al. implemented a PID-based constant-temperature constant-voltage (CT-CV) charging strategy, demonstrating effective temperature regulation during fast charging [107]. Samanta et al. extended this by integrating fuzzy logic with PID to improve the equalisation speed and safety of battery pack balancing [108]. Optimal control (OC) formulations, which minimise an objective function subject to state equations and operational constraints, have also been applied to battery charging. Parvini and Vahidi employed the Thevenin ECM within an OC framework to maximise charging efficiency by minimising resistive losses within prescribed SOC and time bounds [109]. Moura et al. formulated an electrochemical model-based optimal charging strategy designed to minimise the growth of the solid electrolyte interphase (SEI) film during charging [110]. Model predictive control (MPC) has gained particular traction due to its ability to handle multi-variable systems with explicit constraint satisfaction. Chen et al. applied MPC with a Thevenin circuit model to compute optimal charging current trajectories subject to temperature constraints, achieving superior thermal management compared to CC-CV [111]. Zou et al. proposed a linear time-varying MPC framework based on a control-oriented electrochemical thermal model, incorporating physics-based state constraints to prevent battery damage during aggressive charging [62]. Despite the performance advantages of closed-loop strategies, they are critically dependent on the accuracy of the underlying battery model, which can limit their

generalisation to varying battery chemistries, ageing states, and operating conditions.

2.5.2.3 Data-Driven Control Strategies

Data-driven control approaches have gained significant momentum as an alternative to model-dependent closed-loop methods, particularly for complex battery pack management problems. Reinforcement learning (RL) has emerged as a promising framework for designing adaptive charging strategies without requiring explicit battery models. Park et al. developed a model-free RL-based fast-charging strategy employing the actor-critic policy gradient algorithm to handle a continuous action space, demonstrating adaptability to changing environmental parameters in simulation [112]. Neural network-based control has been applied to battery pack balancing by Duraisamy and Kaliyaperumal, who evaluated backpropagation neural networks (BPNN), radial basis neural networks (RBNN), and LSTM for system performance prediction, achieving improved balancing time and reduced power loss compared to conventional strategies [113]. An adaptive MPC policy encoded within a neural network was proposed by Kazhamiaka et al., enabling the controller to learn battery control trajectories and adapt to fluctuating operating environments when labelled data are available [114]. Lin et al. developed a digital signal processor-oriented probabilistic fuzzy neural network (PFNN) for lithium-ion battery pack charging, effectively reducing voltage and current discontinuities at the CC-to-CV mode transition [115]. Although data-driven control methods offer strong nonlinear learning capability and bypass the need for precise physical models, they require substantial high-quality labelled data and tend to exhibit limited generalisation, particularly when deployed outside the distribution of their training conditions.

2.6 Chapter Summary

This chapter has reviewed the principal modelling and control methodologies relevant to the research pursued in this thesis, progressing from cross-domain frameworks, including physics-informed neural networks, grey-box hybrid models, and learning-

enhanced predictive control, to their application in two specific engineering domains. A consistent finding across Sections 2.1 to 2.3 is that deploying high-fidelity models within real-time control loops is hindered by two intertwined bottlenecks, namely inference latency and the data efficiency and generalisation demanded of models trained on scarce industrial data, and that physics-informed machine learning represents the most structurally sound pathway to resolving this tension.

In the domain of steel industrial reheating furnaces (Section 2.4), the zone method coupled with ADI-based finite difference solvers provides the most practical basis for stock temperature prediction, yet the computational cost of iterative numerical integration remains a persistent obstacle to genuinely real-time closed-loop control. In battery energy systems (Section 2.5), analogous limitations arise: physics-based electrochemical models are too computationally demanding for embedded deployment, whilst purely data-driven models sacrifice the physical interpretability required for safety-critical battery management; and existing control strategies, whether open-loop, MPC-based, or reinforcement learning, each exhibit unresolved trade-offs between model dependence, adaptability, and generalisation under realistic operating conditions.

Across both domains, the literature consistently points to the same unresolved gap: the absence of frameworks that simultaneously achieve physical rigour, computational efficiency, and robustness to time-varying system dynamics. This gap directly motivates the research contributions presented in the following chapters of this thesis.

To address these challenges, the following three chapters progress methodologically from modelling, to control, to an integrated modelling-and-control framework, each targeting a distinct aspect of the trade-off identified above. Chapter 3 addresses the physical-rigour and computational-efficiency trade-off in modelling, developing EngGeneNet, a physics-informed CNN that embeds heat conduction physics to achieve real-time, physically consistent temperature field prediction in the reheating furnace case study. Chapter 4 turns to the robustness and adaptability trade-off in control, developing a historical data-driven self-learning MPC framework that exploits repetitive operational data to refine battery single-cell charging trajectories under thermal safety constraints. Chapter

5 then proposes a unified theoretical framework that integrates AI-driven modelling with a corresponding predictive control strategy that explicitly accounts for heterogeneous, time-varying electrical-thermal dynamics in multi-cell battery pack balancing.

CHAPTER 3

PHYSICS-INFORMED ENGGENE NEURAL MODELLING FOR 2D TRANSIENT HEAT CONDUCTION

3.1 Introduction

For the industrial reheating-furnace problem considered in this thesis, accurate and fast transient temperature distribution prediction is a central requirement for real-time operation and control. The internal stock temperature field cannot be measured directly by the radiometric imaging system used in the furnace, while the governing Fourier-law PDE must be solved at sufficient spatial resolution to characterise the through-thickness thermal state. The fitting capability and parallel computation of deep learning models make them attractive as surrogate models for this process when their accuracy and execution time are validated under the operating conditions of interest. However, collecting and processing labelled data remains labour-intensive, and model generalisation can degrade when the process varies across product batches and thermal conditions.

The reheating furnace is a representative and industrially significant application, where intermediate steel products such as blooms, billets, or slabs (known as stock)

must be heated to a target temperature with desired temperature uniformity before hot-rolling [65]. Accurate control of the heating profile is important for product quality and energy efficiency. In this setting, conventional numerical approaches such as the finite difference method (FDM) and the alternating direction implicit (ADI) method can be computationally expensive when applied at the spatial and temporal resolution required for real-time prediction and control.

To address this modelling problem for the furnace application studied here, this chapter proposes a physics-informed deep learning framework termed **EngGeneNet** (Engineering-Genes Network) [26]. The framework integrates three synergistic components:

1. A physics-informed 2D loss function derived from the discretised governing equations with hard-constrained boundary conditions;

2. An Engineering-Gene (*Eng-Gene*) module that embeds *a priori* physical knowledge, specifically the temperature-dependent Fourier number, directly into the neural architecture to enhance generalisation and accelerate convergence [49, 116];

3. A knowledge distillation strategy, wherein the ADI method generates sufficiently high-fidelity training samples, combined with transfer learning to efficiently adapt the model to varying product batches and operating conditions.

The remainder of this chapter is organised as follows. Section 3.2 presents the mathematical formulation of the 2D transient heat conduction problem, including the governing equations, boundary conditions, and numerical discretisation. Section 3.3 details the physics-informed learning framework and constraint embedding strategy. Section 3.4 describes the EngGeneNet architecture, the Eng-Gene module design, and the knowledge distillation and transfer learning strategies. Section 3.5 presents a comprehensive case study on a pilot-scale walking-beam steel reheating furnace, including ablation studies, sensitivity analysis, and computational efficiency evaluation. Section 3.6 summarises the key contributions and findings.

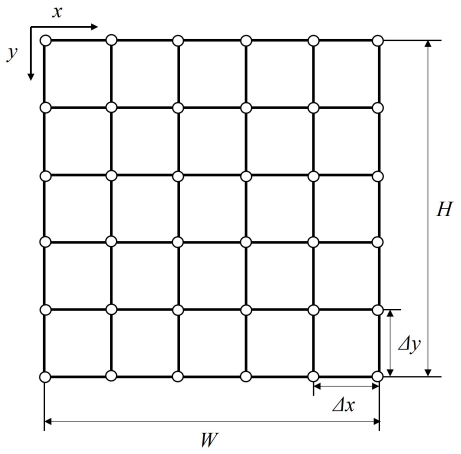


Figure 3.1: 2D heat conduction problem.

3.2 General Problem Formulation for 2D Transient Heat Conduction

3.2.1 Governing Equations and Modelling Assumptions

As shown in Figure 3.1, the heat conduction problem considered in this chapter is formulated as a general transient two-dimensional process defined over a rectangular domain with dimensions (x, y). The governing physics is characterised by spatially varying thermal properties and an initial temperature distribution prescribed over the interior of the domain. Thermal energy diffuses inward from the boundaries, where heat exchange with the surrounding environment occurs across all four edges. This formulation is broadly applicable to a range of engineering heat transfer scenarios; in the context of this thesis, it serves as the basis for modelling the thermal behaviour of steel stock inside an industrial reheating furnace.

Under the assumption of 2D transient heat conduction, the thermal behaviour of the stock is governed by Fourier's law:

$$\frac{\partial T}{\partial t} = \frac{k(T)}{\rho C_p(T)} \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right), \tag{3.1}$$

$$T = T_t(x, y), \tag{3.2}$$

where x is the steel product width direction, y is the steel product height direction, $k(T)$ and $C_p(T)$ are temperature-dependent material properties representing the thermal conductivity and specific heat capacity, respectively, and ρ is the material density.

3.2.2 Boundary Conditions

To obtain accurate and stable solutions to the governing equation, appropriate boundary conditions must be specified. In the reheating furnace, the steel stock exchanges heat with the surrounding environment through both radiation and convection at its surface. Accordingly, a hard boundary heat transfer condition is described by:

$$-k(T)\frac{\partial T}{\partial \vec{n}} = h_c(T - T_A) + \sigma e(T^4 - T_A^4), \tag{3.3}$$

where $k(T)$ is the thermal conductivity, h_c denotes the convective heat transfer coefficient, $\sigma = 5.6687 \times 10^{-8}\,\mathrm{Wm^{-2}K^{-4}}$ is the Stefan–Boltzmann constant, e stands for the emissivity, $-k(T)\partial T/\partial \vec{n}$ is the heat flux normal to the surface, and T_A is the ambient temperature.

The computational domain is partitioned into an interior domain D_I and a boundary domain ∂D:

$$D_I = \{(x, y) \mid 0 < x < W,\ 0 < y < H\}, \tag{3.4}$$

$$\partial D = \{(x, y) \mid 0 \le x \le W,\ y = 0 \text{ or } H;\ 0 \le y \le H,\ x = 0 \text{ or } W\}, \tag{3.5}$$

where W and H are the width and height of the stock, respectively.

3.2.3 Numerical Discretisation and the Crank–Nicolson Scheme

To derive the physics-informed loss function, the continuous governing equation is discretised over a uniform square mesh with spacing $d = \Delta x = \Delta y$ (illustrated in Figure 3.1). A forward difference approximation is adopted for the time derivative:

$$\frac{\partial T}{\partial t} = \frac{T_{t+1}(x, y) - T_t(x, y)}{\Delta t} + \mathcal{O}(\Delta t). \tag{3.6}$$

For interior domain nodes D_I, the second-order spatial derivatives are:

$$\frac{\partial^2 T}{\partial x^2} = \frac{T_t(x+1, y) - 2T_t(x, y) + T_t(x-1, y)}{(\Delta x)^2} + \mathcal{O}(\Delta x)^2, \tag{3.7}$$

$$\frac{\partial^2 T}{\partial y^2} = \frac{T_t(x,y+1) - 2T_t(x,y) + T_t(x,y-1)}{(\Delta y)^2} + \mathcal{O}(\Delta y)^2, \tag{3.8}$$

where $\mathcal{O}(\Delta t)$, $\mathcal{O}(\Delta x)^2$, and $\mathcal{O}(\Delta y)^2$ are the truncation errors. By substituting equations 3.6, 3.7 and 3.8 into the governing equation 3.1, the temperature evolution on each node within the domain can be computed. To further enhance the accuracy and stability of the solutions, the Crank–Nicolson method [70] is employed, which approximates the second-order spatial derivative at an intermediate time step between t and $t+1$:

$$\frac{\partial^2 T}{\partial x^2} = \epsilon \frac{\partial^2 T}{\partial x^2}\bigg|_{t+1} + (1-\epsilon)\,\frac{\partial^2 T}{\partial x^2}\bigg|_{t}, \tag{3.9}$$

$$\frac{\partial^2 T}{\partial y^2} = \epsilon \frac{\partial^2 T}{\partial y^2}\bigg|_{t+1} + (1-\epsilon)\,\frac{\partial^2 T}{\partial y^2}\bigg|_{t}, \tag{3.10}$$

where ϵ is a constant determining the weight of time-steps t and $t+1$, with $0 < \epsilon \leq 1$. As ϵ approaches 1, accuracy decreases progressively for a given Δt, Δx, and Δy. When $\epsilon = 0.5$, the minimum accumulated truncation error is achieved, yielding the Crank–Nicolson scheme with high stability. Accordingly, the second-order spatial derivative is reformulated as:

$$\frac{\partial^2 T}{\partial x^2} = \frac{1}{2}\frac{T_{t+1}(x+1,y) - 2T_{t+1}(x,y) + T_{t+1}(x-1,y)}{(\Delta x)^2} + \frac{1}{2}\frac{T_t(x+1,y) - 2T_t(x,y) + T_t(x-1,y)}{(\Delta x)^2}, \tag{3.11}$$

$$\frac{\partial^2 T}{\partial y^2} = \frac{1}{2}\frac{T_{t+1}(x,y+1) - 2T_{t+1}(x,y) + T_{t+1}(x,y-1)}{(\Delta y)^2} + \frac{1}{2}\frac{T_t(x,y+1) - 2T_t(x,y) + T_t(x,y-1)}{(\Delta y)^2}. \tag{3.12}$$

Setting $\Delta x = \Delta y = d$ and substituting into equation 3.1, the discretised governing equation for interior nodes can be expressed as:

$$\begin{aligned} &T_{t+1}(x+1,y) + T_{t+1}(x-1,y) + T_{t+1}(x,y+1) + T_{t+1}(x,y-1) - 4T_{t+1}(x,y) \\ &+ T_t(x+1,y) + T_t(x-1,y) + T_t(x,y+1) + T_t(x,y-1) - 4T_t(x,y) \\ &= 2\big(T_{t+1}(x,y) - T_t(x,y)\big)/Fo, \end{aligned} \tag{3.13}$$

where the *Fourier number* is defined as:

$$Fo = \frac{k(T)\Delta t}{\rho\, C_p(T)\, d^2}. \tag{3.14}$$

Rearranging all terms to one side and grouping like terms, equation 3.13 becomes:

$$T_{t+1}(x+1,y)+T_{t+1}(x-1,y)+T_{t+1}(x,y+1)+T_{t+1}(x,y-1)+4\left(-1-\frac{1}{2Fo}\right)T_{t+1}(x,y) \\ +T_t(x+1,y)+T_t(x-1,y)+T_t(x,y+1)+T_t(x,y-1)+4\left(-1+\frac{1}{2Fo}\right)T_t(x,y)=0, \tag{3.15}$$

which forms the foundational expression for the physics-informed 2D loss function used to supervise the training of the deep learning model.

3.3 Physics-Informed Learning Framework and Constraint Embedding

In this study, a physics-informed EngGeneNet framework is proposed to approximate the solutions to the governing PDEs for heat conduction by formulating a 2D loss function using the finite difference method. The deep learning model is then employed as a surrogate PDE solver to predict the transient temperature distribution of steel stocks in reheating furnaces. To implement the proposed physics-informed 2D loss, customised convolution kernels are designed to efficiently approximate spatial derivatives. Moreover, hard-constrained boundary conditions are embedded within the loss function to ensure stable model training and to account for environmental variations across different operating conditions. In addition, an 'Eng-Gene' module is integrated into the deep learning model architecture to enhance generalisation capability and accelerate convergence. The overall framework is illustrated in Figure 3.2.

3.3.1 Formulation of the 2D Loss Function

First of all, a square mesh is constructed to discretise the cross-section of the rectangular steel products, with the objective of predicting the temperature evolution at each mesh node. Owing to the image-like spatial temperature distribution, a CNN architecture is adopted as the backbone of the surrogate model for efficient approximation of PDE solutions. Then, the developed physics-informed loss function plays an important role

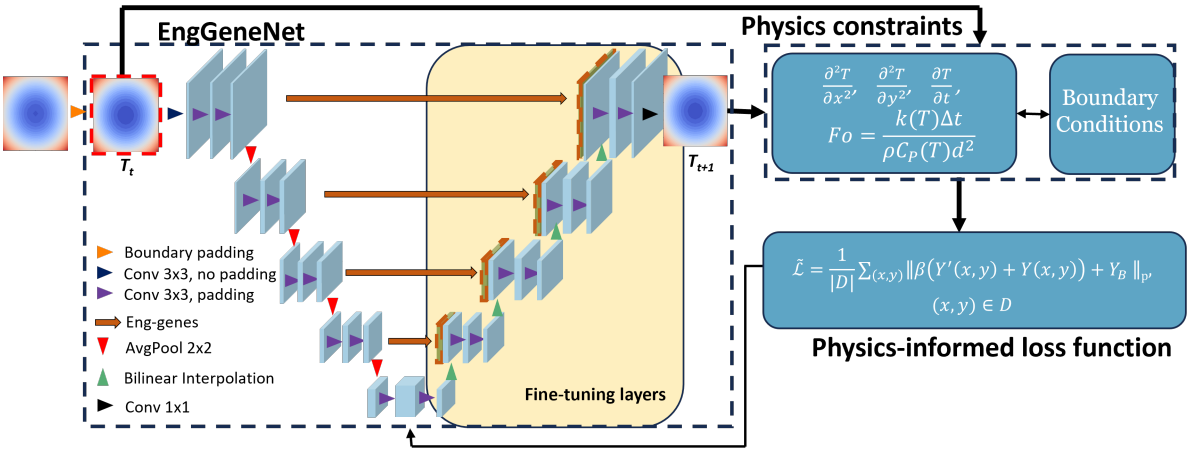


Figure 3.2: The framework of the proposed physics-informed EngGeneNet.

in guiding the model training direction and shaping the model behaviour. Therefore, the accurate formulation of the loss function based on the underlying physical laws is essential for ensuring both stability and fidelity in model predictions.

In this study, the 2D temperature distribution $T_t(x,y)$ is divided into the interior domain D_I and boundary domain ∂D as defined in equations 3.4 and 3.5. The discretised computation domain with dimensions $m \times n$ for the model input $T_t(x,y)$ and model output $\widetilde{T}_{t+1}(x,y)$ is illustrated in Figure 3.1, where $m = W/\Delta x$ and $n = H/\Delta y$.

To implement the proposed physics-informed 2D loss function within the deep learning framework, a customised convolution kernel is designed to compute the approximation error during model training. Specifically, the discrete-time governing equation 3.15 is decomposed into two components: T_t, representing the temperature distribution at the current time step, and T_{t+1}, representing the temperature distribution at the subsequent time step, the latter corresponds to the predicted output and is denoted as $\widetilde{T}_{t+1}$. As a result, the model training can be considered as the following optimisation problem:

$$\begin{aligned} \min_{\widetilde{T}_{t+1}(x,y)} \frac{1}{|D_I|} \sum_{(x,y)} \left\| \beta \cdot [Y'(x,y) + Y(x,y)] \right\|_p \\ \text{s.t.} \quad (x,y) \in D_I, \\ -k(T)\frac{\partial T}{\partial \vec{n}} = h_c(T - T_A) + \sigma e(T^4 - T_A^4), \end{aligned} \tag{3.16}$$

where D_I is the computation domain, p is the order of the norm, and β is the weighting factor regulating the relative importance assigned to the interior computational domain.

$Y'(x,y)$ is defined to include all terms needed to predict the temperature distribution at the next time step in equation 3.15:

$$\begin{aligned} Y'(x,y) = {} & \widetilde{T}_{t+1}(x+1,y) + \widetilde{T}_{t+1}(x-1,y) \\ & + \widetilde{T}_{t+1}(x,y+1) + \widetilde{T}_{t+1}(x,y-1) \\ & + 4\left(-1 - \frac{1}{2Fo}\right)\widetilde{T}_{t+1}(x,y), \end{aligned} \tag{3.17}$$

$Y(x,y)$ is defined to include all terms from the current temperature distribution at time-step t in equation 3.15:

$$\begin{aligned} Y(x,y) = {} & T_t(x+1,y) + T_t(x-1,y) \\ & + T_t(x,y+1) + T_t(x,y-1) \\ & + 4\left(-1 + \frac{1}{2Fo}\right)T_t(x,y). \end{aligned} \tag{3.18}$$

Finally, the objective function in the optimisation problem is reformulated as the inner domain loss function for the neural model training:

$$\widetilde{\mathscr{L}} = \frac{1}{|D_I|}\sum_{(x,y)\in D_I} \left\| \beta[Y'(x,y) + Y(x,y)] \right\|_p, \tag{3.19}$$

where $Y'(x,y)$ and $Y(x,y)$ represent the residual components associated with the predicted and input temperature fields, respectively.

3.3.2 Customised Loss Convolution Kernels

To ensure consistency between model training and loss evaluation, two customised convolution kernels are designed to efficiently compute $Y'(x,y)$ and $Y(x,y)$ based on the discretised physics-informed formulation. The structure of these convolution kernels is illustrated in Figure 3.3, with their corresponding coefficients α' and α computed as:

$$\alpha' = 4\left(-1 - \frac{1}{2Fo}\right), \tag{3.20}$$

$$\alpha = 4\left(-1 + \frac{1}{2Fo}\right), \tag{3.21}$$

where Fo is the Fourier number as previously defined in equation 3.14.

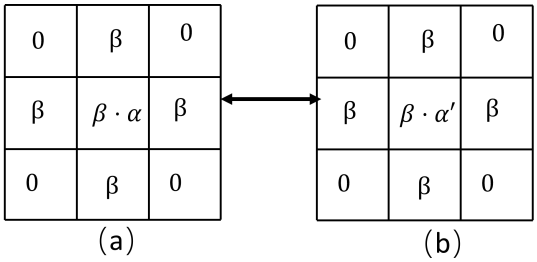


Figure 3.3: (a) and (b) are the customised convolution kernels for computing $Y'(x, y)$ and $Y(x, y)$, respectively.

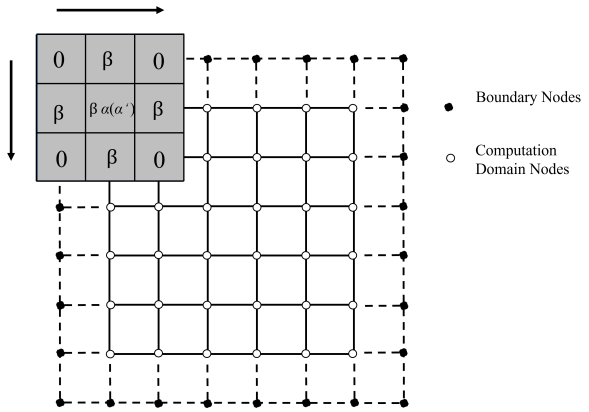


Figure 3.4: Customised loss function convolution process.

Figure 3.4 presents a schematic diagram illustrating the application of the customised convolution kernels to both the model input and output. The interior-domain loss value is subsequently computed based on the resulting convolution outputs. The spatial structure of these kernels directly mirrors the Crank–Nicolson stencil, ensuring that kernel-based residual evaluation is mathematically equivalent to the finite difference formulation derived in section 3.2.3.

3.3.3 Constrained Boundary Condition Integration

Accurate boundary condition constraints are essential for ensuring physically consistent solutions to the governing equations in heat conduction problems. In this study, a hard-constrained boundary condition is incorporated into the model loss to explicitly enforce both convective and radiative boundary conditions. This design enhances the robustness and stability of the neural network, particularly under diverse thermal environments.

1	1	1	1	1	1	1
1	0	0	0	0	0	1
1	0	0	0	0	0	1
1	0	0	0	0	0	1
1	0	0	0	0	0	1
1	0	0	0	0	0	1
1	1	1	1	1	1	1

Figure 3.5: Mask of boundary constraints.

Specifically, by applying the inward finite difference approximation to the boundary condition and rearranging the terms into a residual form, the boundary condition loss Y_B at each boundary node is defined as:

$$Y_B = k(T)\frac{T_B - T_P}{d} + h_c(T_B - T_A) + \sigma e(T_B^4 - T_A^4), \tag{3.22}$$

where T_B denotes the boundary node temperature and T_P is the adjacent interior node. To account for the different spatial orientations of the boundary, the loss terms are formulated separately for each edge of the computational domain:

$$\begin{cases} Y_B(0,y) = k(T)\dfrac{T(1,y) - T(0,y)}{d} + h_c(T(0,y) - T_A) + \sigma e(T(0,y)^4 - T_A^4), & 0 \le y \le n, \\ Y_B(m,y) = k(T)\dfrac{T(m,y) - T(m-1,y)}{d} + h_c(T(m,y) - T_A) + \sigma e(T(m,y)^4 - T_A^4), & 0 \le y \le n, \\ Y_B(x,0) = k(T)\dfrac{T(x,1) - T(x,0)}{d} + h_c(T(x,0) - T_A) + \sigma e(T(x,0)^4 - T_A^4), & 0 \le x \le m, \\ Y_B(x,n) = k(T)\dfrac{T(x,n) - T(x,n-1)}{d} + h_c(T(x,n) - T_A) + \sigma e(T(x,n)^4 - T_A^4), & 0 \le x \le m. \end{cases} \tag{3.23}$$

To efficiently integrate this constrained boundary loss into the CNN training framework, a binary computation mask (illustrated in Figure 3.5) is employed to localise the loss computation strictly to the boundary region ∂D. This mask-based approach enables tensor-wise parallelisation and is fully compatible with modern deep learning frameworks.

The resulting boundary loss Y_B is then incorporated into the physics-informed 2D loss function to formulate the final composite loss function for model training:

$$\widetilde{\mathscr{L}} = \frac{1}{|D|} \sum_{(x,y)\in D} \left\| \beta[Y'(x,y) + Y(x,y)] + Y_B \right\|_p, \tag{3.24}$$

where β is the weighting factor regulating the relative importance assigned to the interior computational domain and the boundary conditions. The integration of Y_B from the hard-constrained boundary conditions not only enforces physical consistency at the domain boundaries but also ensures that the learning process remains stable and grounded in domain-specific laws. This hybrid formulation, combining data-driven learning with physically meaningful constraints, significantly enhances the model robustness and generalisation under various thermal boundary conditions.

3.4 Engineering Gene Network Architecture and Training Strategy

3.4.1 CNN Backbone: UNet as a Surrogate PDE Solver

In this work, the CNN architecture is adopted as the model backbone to approximate the proposed 2D loss function, which can significantly reduce computation time by leveraging the parallel computing capabilities of convolution operations. This approach has significant potential for the real-time operation and control of the reheating furnace. The UNet is a U-shaped CNN architecture that includes both contracting and expansion paths [117]. The contraction path captures global contextual information, while the expansion path provides high-resolution spatial details. These feature maps generated by the two paths are then concatenated through skip connections. This unique structure has achieved significant success in various image-related tasks. In contrast to the conventional applications of UNet in image segmentation, this study reformulates the task as an image-to-image regression problem, wherein the objective is to predict continuous temperature fields, i.e. learning a mapping $T^t(x,y) \mapsto \widetilde{T}^{t+1}(x,y)$.

The ambient conditions are implemented around the computational domain through a designed padding function; consequently, the first convolution layer is left unpadded to preserve the original data shape. Since both the input and the model output are required to calculate the physics-informed loss function during model training, a skip connection is introduced between the two ends of the neural model.

3.4.2 Engineering-Gene (Eng-Gene) Module Design

In addition, key material properties such as specific heat capacity $C_p(T)$ and thermal conductivity $k(T)$ exhibit strong nonlinearity and temperature dependence during the heat conduction process. In this study, these dependencies are modelled using polynomial functions of temperature:

$$\begin{cases} C_p(T_m) = a_1 + a_2 T_m + a_3 T_m^2 + a_4 T_m^3 + a_5 T_m^4 + a_6 T_m^5, \\ k(T_m) = b_1 + b_2 T_m, \end{cases} \tag{3.25}$$

where T_m is the mean value over the whole computation domain at the current time step. The polynomial coefficients a_i and b_i determine the heat transfer properties of different materials and are provided in Table 3.1 [65].

Table 3.1: Polynomial coefficients a_i for $C_p(T)$ and b_i for $k(T)$.

	$a_1(b_1)$	$a_2(b_2)$	a_3	a_4	a_5	a_6
$C_p(T)$	0.377E+03	0.125E+00	−0.197E-04	−0.253E-06	0.398E-09	−0.180E-12
$k(T)$	0.385E+03	−0.481E-01	—	—	—	—

Furthermore, the Fourier number Fo (defined in equation 3.14) is a dimensionless parameter characterising the ratio of the heat diffusion rate to the time-scale of a transient thermal process. It plays a critical role in characterising the transient temperature evolution of materials and is directly influenced by $C_p(T)$ and $k(T)$. As shown in equation 3.13, the governing equation depends on $T(x,y)/Fo$, indicating the presence of a salient fundamental physical relationship within the physics model.

When solving PDEs analytically, explicitly incorporating such temperature-dependent material properties can substantially increase computational complexity, especially for

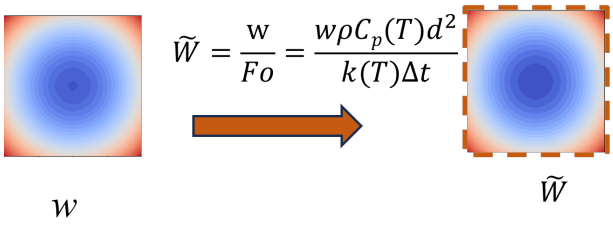


Figure 3.6: 'Eng-Gene' module.

large-scale problems. Conversely, relying solely on data-driven models such as CNNs to learn these complex physical relationships may result in poor generalisation and unreliable performance.

To address this limitation, *a priori* engineering knowledge is embedded into the neural network training process through the design of specialised convolution layers that encode such explicit physical relationships. This approach, referred to as the 'Eng-Gene' method, extracts fundamental functional components that capture nonlinearities inherent in the governing physical laws of engineering systems. These components act as thermal 'genetic' information, enriching the learning process [116].

In this work, the 'Eng-Gene' module associated with the Fourier number is defined as:

$$\widetilde{W} = \frac{w}{Fo} = \frac{w\,\rho\,C_p(T)d^2}{k(T)\,\Delta t}, \tag{3.26}$$

where w is the model input data and $\widetilde{W}$ is the output of the designed 'Eng-Gene' module; $C_p(T)$ and $k(T)$ are temperature-dependent properties. $\widetilde{W}$ can be regarded as an intermediate variable of the neural model, as illustrated in Figure 3.6. It helps to extract the underlying nonlinear relationship between the input and output, thereby enabling the model to more accurately and quickly minimise the physics-informed loss function.

The overall framework of the proposed physics-informed EngGeneNet is illustrated in Figure 3.2, where the input is a 2D temperature distribution T_t at time-step t, and the output is the predicted transient temperature distribution $\widetilde{T}_{t+1}$ at time-step $t+1$. Considering that the convolution layers at different levels in the UNet architecture are responsible for extracting features of varying complexities, the 'Eng-Gene' module is embedded into each level of the convolution layers in the UNet architecture to enhance

the model prediction performance and the generalisation ability, as shown on the orange path in Figure 3.2. Notably, due to the plug-and-play feature, the 'Eng-Gene' module can not only be easily integrated into the UNet framework but can also be flexibly integrated with other neural network architectures.

3.4.3 Knowledge Distillation for Training Data Generation

A conventional method for capturing the temperature profile of steel blooms is the use of radiometric imaging cameras. However, this technique only provides surface temperature measurements and fails to capture the internal thermal distribution throughout the entire stock. Moreover, acquiring sufficient experimental data for training pure data-driven deep learning models in complex industrial settings is often prohibitively expensive and time-consuming. To overcome these limitations, numerical methods such as the ADI method have been widely employed to generate operational data. This approach was also adopted in previous work by [65].

The ADI method decomposes the energy balance equation at each time step into two sequential sets of finite difference equations (FDEs), solving them alternately along different spatial directions. In the first half-step, the terms involving the y-direction are treated as unknowns, and the governing equation for this half-time step is:

$$\begin{aligned}&(T_{t+1}(x,y-1)-T_{t+1}(x,y))+(T_{t+1}(x,y+1)-T_{t+1}(x,y))\\&+(T_t(x-1,y)-T_t(x,y))+(T_t(x+1,y)-T_t(x,y))\\&=\left(\frac{\rho C_p(T)d^2}{k(T)}\right)\left(\frac{T_{t+1}(x,y)-T_t(x,y)}{\Delta t/2}\right).\end{aligned} \tag{3.27}$$

In the second half-step, the roles of spatial directions are reversed, and the x-direction terms are treated as unknowns:

$$\begin{aligned}&(T_t(x,y-1)-T_t(x,y))+(T_t(x,y+1)-T_t(x,y))\\&+(T_{t+1}(x-1,y)-T_{t+1}(x,y))+(T_{t+1}(x+1,y)-T_{t+1}(x,y))\\&=\left(\frac{\rho C_p(T)d^2}{k(T)}\right)\left(\frac{T_{t+1}(x,y)-T_t(x,y)}{\Delta t/2}\right).\end{aligned} \tag{3.28}$$

The boundary conditions are consistent with those defined in equation 3.23. The final temperature distribution is obtained by alternately solving these finite difference equations using the Gaussian elimination method at each time step.

3.4.4 Transfer Learning for Operating Condition Adaptation

Most deep learning techniques face challenges when addressing the issue of feature space deviation among different datasets. A commonly adopted solution to mitigate this issue is transfer learning (TL). In this study, the variability in operating conditions is also considered. Specifically, differences in size, density, and other material properties across production batches can lead to shifts in the feature space of the temperature distribution profiles. To reduce model training time while maintaining comparable prediction accuracy, the transfer learning approach is employed.

In the implementation, the material property parameters within both the physics-informed loss function and the 'Eng-Gene' module are updated to adapt to the varying operating conditions. Subsequently, only the parameters in the expansion path of the UNet architecture (highlighted in yellow in Figure 3.2) are fine-tuned using the pre-trained baseline model, while the parameters in the contraction path are kept frozen. This design leverages the role of the contraction path in capturing global contextual features, while the expansion path is responsible for reconstructing higher-resolution spatial details. This fine-tuning process is performed using a relatively small subset of data samples and training epochs, making the adaptation process efficient. In addition, the integration of the 'Eng-Gene' module significantly enhances the effectiveness of the transfer learning strategy by embedding the physical variation into the model.

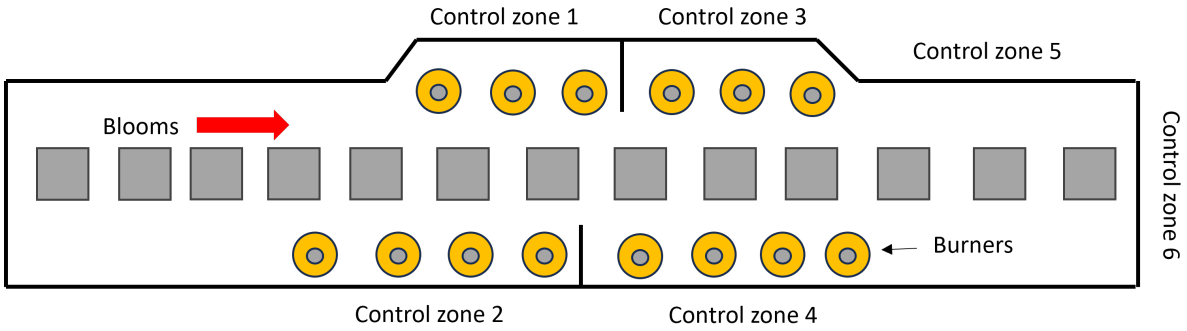


Figure 3.7: Schematic of the reheating furnace.

3.5 Case Study: Transient Heat Conduction in Steel Reheating Process

3.5.1 Furnace Operating Conditions and Product Batches

The furnace investigated in this study is a pilot-scale walking-beam bloom reheating furnace with dimensions of 36 m × 10 m (length × width) and a height ranging from 4.0 m to 4.7 m, as illustrated in Figure 3.7 [65]. The furnace model is divided into six control zones (CZs), where CZ1 and CZ3 are the master thermal control zones, while CZ2 and CZ4 are subordinate zones. The thermal inputs to the subordinate zones are proportionally regulated based on the thermal inputs of the corresponding master control zones. During operation, blooms enter the furnace from one end, progress through the heating zone, and are discharged from the opposite end.

As shown in Figure 3.8, the intermediate steel products considered in this study are primarily standard rectangular prisms with dimensions (x, y, z), having an initial temperature distribution upon entering the reheating furnace. For modelling simplicity and in line with practical considerations, the temperature gradient variation in the z-dimension of the stock is typically regarded as negligible compared with the other two directions. Therefore, the heat conduction of steel products can be modelled as a 2D process: heat exchange between the stock and the furnace environment occurs through the four boundaries, while thermal energy diffuses inward from the surface toward the interior.

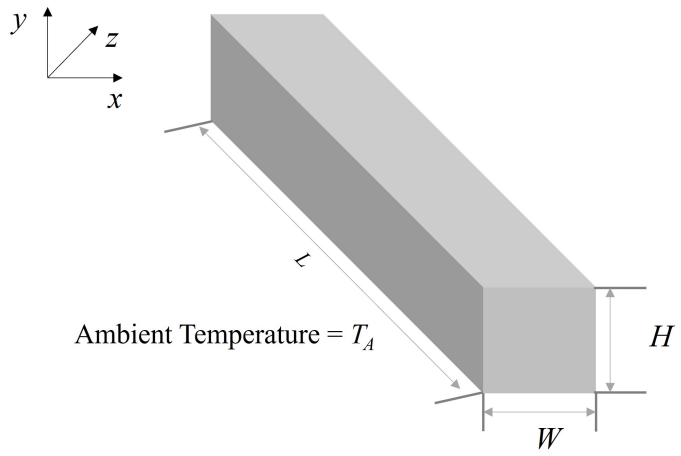


Figure 3.8: Schematic of the intermediate steel product.

In the experimental evaluation, two distinct batches of mild steel blooms, designated SB1 and SB2, were tested, with their material properties summarised in Table 3.2. Furthermore, to assess the model prediction performance under varying physical conditions, a range of operating conditions (OCs) was introduced, as detailed in Table 3.3. Specifically, OC1 serves as the baseline operating condition and is used to configure the hyperparameters of the baseline model. OC2 is designed to evaluate the transfer learning performance when adapting to different product batches. OC3 is employed to assess model robustness under varying data sampling intervals. OC4 is introduced to further validate the robustness and adaptability of the proposed EngGeneNet under diverse environmental conditions. These varying operating conditions and product types provide a comprehensive framework for evaluating the proposed EngGeneNet across several key performance metrics, including generalisation capability, transfer learning effectiveness, and training efficiency.

3.5.2 Knowledge Distillation and Dataset Generation for Model Training

In the experiments, a total of 12,000 two-dimensional temperature distribution samples under operating condition OC1 were generated using the ADI method to develop the

Table 3.2: Material properties in the experiment.

	Properties	Value	Unit
SB1	Mass	5353	kg
	Thermal conductivity($k(T)$)	Eqn. (25)	$Wm^{-1}K^{-1}$
	Specific heat capacity($C_p(T)$)	Eqn. (25)	$Jkg^{-1}K^{-1}$
	Convection heat transfer coefficient(h_c)	25	$Wm^{-2}K^{-1}$
	Stefan-Boltzmann constant(σ)	5.6687×10^{-8}	$Wm^{-2}K^{-4}$
	Emissivity	0.8	-
	Width	0.3	m
	Height	0.3	m
	Length	7.625	m
	d	0.01875	m
SB2	Mass	8827	kg
	Thermal conductivity($k(T)$)	Eqn. (25)	$Wm^{-1}K^{-1}$
	Specific heat capacity($C_p(T)$)	Eqn. (25)	$Jkg^{-1}K^{-1}$
	Convection heat transfer coefficient(h_c)	25	$Wm^{-2}K^{-1}$
	Stefan-Boltzmann constant(σ)	5.6687×10^{-8}	$Wm^{-2}K^{-4}$
	Emissivity	0.8	-
	Width	0.4	m
	Height	0.4	m
	Length	7.028	m
	d	0.025	m

Table 3.3: Operating conditions in experiments.

	Blooms	Init. Temp (°C)	Vert. Temp (°C)	Horiz. Temp (°C)	Δt (s)
OC1	SB1	20	1250	1250	1
OC2	SB2	20	1200	1200	2
OC3	SB1	35	1250	1250	2
OC4	SB2	20	**1200**	**1000**	1

baseline model. Additionally, 6,000 samples under OC2 and OC4, and 5,000 samples under OC3 were generated to fine-tune the physics-informed EngGeneNet and to evaluate model performance under varying operating conditions. The full dataset was divided into training, validation, and testing sets with a ratio of 8:1:1. In the model configuration, the padding values in the convolution layers are set to 'zeros'. The Adam optimiser is used to train the model. The base model was trained in 50 epochs; however, during the transfer learning, only 30 epochs of training were conducted on a relatively small training dataset,

achieving the same accuracy as the base model. All experiments were conducted on a workstation equipped with 8 Intel Core processors, 16 GB of RAM, and a GTX 2080 GPU with 8 GB of graphics memory. To quantitatively assess the performance of the proposed physics-informed EngGeneNet, the following evaluation metrics are used: the mean absolute error (MAE), the root mean square error for all nodes (RMSE), and the mean absolute error of maximum absolute errors for the dataset (MAE-MaxE).

3.5.3 Ablation Studies: Effectiveness of the Eng-Gene Module

To validate the effectiveness and adaptability of the proposed physics-informed framework, a series of ablation experiments involving different deep learning backbones were conducted, including VGG [118], Feature Pyramid Network (FPN) [119], and UNet. These architectures were selected to cover a broad spectrum of convolutional network designs, from the sequential convolutional layers in VGG, to the multi-scale feature fusion strategy in FPN, and the encoder-decoder structure in UNet. In each case, the model performance was evaluated both with and without the integration of the proposed 'Eng-Gene' module. Due to its modular and plug-and-play design, the 'Eng-Gene' module can be seamlessly incorporated into these networks without altering the overall architecture.

The results for both the training and test datasets, as presented in Table 3.4, demonstrate the superior prediction performance of the proposed physics-informed EngGeneNet model. First of all, integrating the 'Eng-Gene' module into the baseline models (VGG, FPN, and UNet) consistently leads to substantial performance improvements compared to their original backbone without the module. Furthermore, among the three deep learning models, EngGeneUNet exhibits the highest accuracy on both datasets, achieving at least 90% MAE reduction, 85% RMSE reduction, and 45% MAE-MaxE reduction on both the training and test datasets compared to the physics-informed UNet without the 'Eng-Gene' module. This confirms the significance of the 'Eng-Gene' module in improving the prediction performance of deep learning models. In addition, the values of MAE and RMSE on test datasets for EngGeneUNet are both below 0.1, showing that the proposed

EngGeneUNet can achieve equivalent accuracy to the conventional ADI method.

Table 3.4: Performance comparison of various deep learning backbones with and without 'Eng-Gene' in the physics-informed framework.

Models	Training dataset			Test dataset		
	MAE	RMSE	MAE-MaxE	MAE	RMSE	MAE-MaxE
PI-VGG	1.257	1.534	4.200	1.541	1.821	4.621
PI-EngGeneVGG	0.595 (↓52%)	0.727 (↓52%)	2.405 (↓42%)	0.598 (↓61%)	0.730 (↓59%)	2.416 (↓51%)
PI-FPN	0.477	0.530	1.193	0.699	0.753	1.437
PI-EngGeneFPN	0.169 (↓64%)	0.215 (↓59%)	0.715 (↓40%)	0.171 (↓75%)	0.217 (↓71%)	0.720 (↓49%)
PI-UNet	0.276	0.298	0.557	0.515	0.538	0.822
PI-EngGeneUNet	**0.026** (↓90%)	**0.044** (↓85%)	**0.305** (↓45%)	**0.029** (↓94%)	**0.048** (↓91%)	**0.313** (↓61%)

To further evaluate the generalisation ability of the proposed physics-informed EngGeneNet, Figure 3.9 presents a comparison of the performance gaps between training and test datasets in terms of three standard evaluation metrics: MAE, RMSE, and MAE-MaxE. These gaps are calculated as the absolute differences between the training and test errors:

$$\mathrm{MAE}_{\mathrm{gap}} = \left|\mathrm{MAE}_{\mathrm{test}} - \mathrm{MAE}_{\mathrm{train}}\right|, \tag{3.29}$$

$$\mathrm{RMSE}_{\mathrm{gap}} = \left|\mathrm{RMSE}_{\mathrm{test}} - \mathrm{RMSE}_{\mathrm{train}}\right|, \tag{3.30}$$

$$\mathrm{MaxE}_{\mathrm{gap}} = \left|\mathrm{MAE\text{-}MaxE}_{\mathrm{test}} - \mathrm{MAE\text{-}MaxE}_{\mathrm{train}}\right|. \tag{3.31}$$

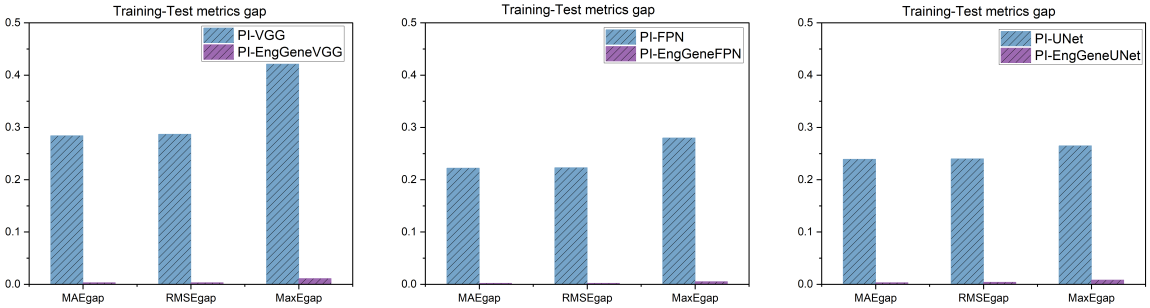


Figure 3.9: Metrics gap between training and test datasets of various deep learning models with and without the 'Eng-Gene' module.

Notably, all the physics-informed EngGeneNet models with the 'Eng-Gene' module exhibit far better generalisation performance on all three performance metrics than the deep learning models without the 'Eng-Gene' module. Furthermore, Figure 3.10

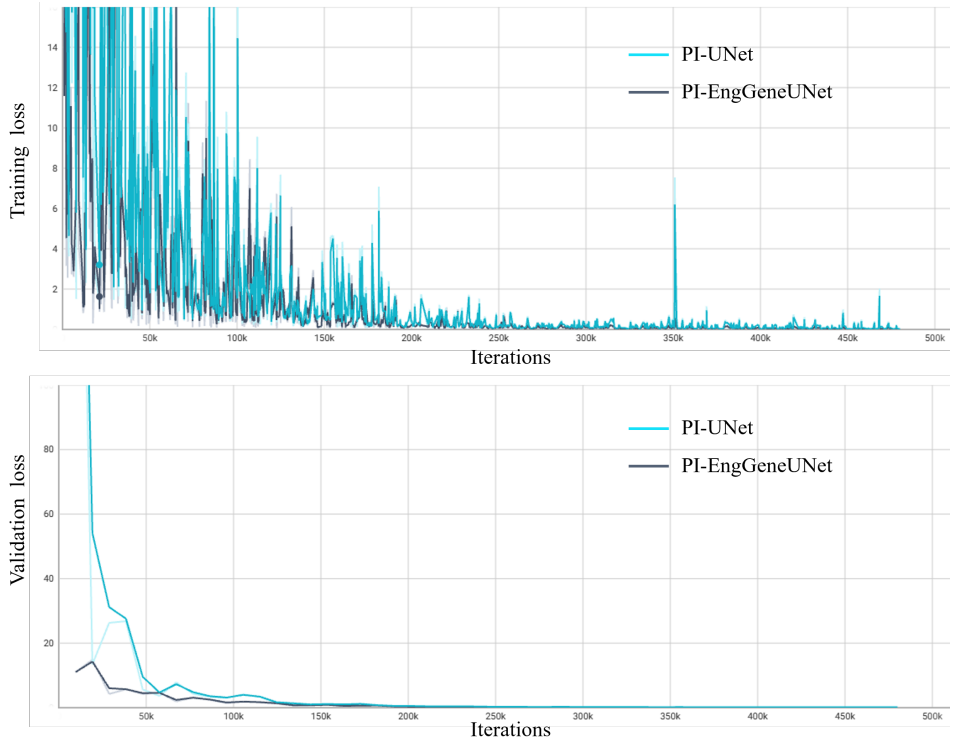


Figure 3.10: Training and validation loss evolution of PI-EngGeneUNet and PI-UNet.

illustrates the evolution of the loss function for both the physics-informed EngGeneUNet and physics-informed UNet across training and validation datasets under the same number of training epochs. The initial training and validation losses of EngGeneUNet are consistently lower, and the PI-EngGeneUNet model training also exhibits a significantly faster convergence speed. These analyses confirm that the 'Eng-Gene' module plays a key role in significantly enhancing both the generalisation ability and convergence speed of deep learning models.

In addition, a comparison experiment was conducted to further evaluate the effectiveness of the proposed EngGeneUNet. In this experiment, the 'Eng-Gene' module in the proposed physics-informed EngGeneUNet was replaced with an extra 1×1 convolution layer so that both models have the same complexity. The two models were then trained for the same number of epochs. Table 3.5 shows the performance of the two models for all performance metrics. The results show that the performance of the UNet model with an 'Eng-Gene' module is much better than the model with an extra convolution layer

when both models are trained with the same epochs, achieving 72% MAE reduction, 67% RMSE reduction, and 54% MAE-MaxE reduction on the test dataset. Furthermore, the EngGeneUNet requires only 23 epochs to achieve a MAE comparable to that of the UNet model with an extra convolution layer trained for 50 epochs. This again confirms that the 'Eng-Gene' module can accelerate the training speed with significantly improved model performance.

Table 3.5: Comparison results of the integration of the 'Eng-Gene' module or an extra convolution layer.

Module	Metrics	Training	Test	Epochs
Extra conv	MAE	0.099	0.106	50
	RMSE	0.139	0.148	
	MAE-MaxE	0.670	0.690	
Eng-Gene	MAE	0.026	**0.029 (↓72%)**	50
	RMSE	0.044	**0.048 (↓67%)**	
	MAE-MaxE	0.305	**0.313 (↓54%)**	
	MAE	0.090	0.093	**23**
	RMSE	0.105	0.109	
	MAE-MaxE	0.311	0.331	

To further validate the independent contribution of the 'Eng-Gene' module, an ablation study was conducted to investigate the role of different physical variables incorporated within the module. In this experiment, the variables d^2, ρ, $k(T)$, and $C_p(T)$ were individually removed from the model to assess their respective importance and contribution to the overall performance. Figure 3.11 illustrates the evolution of the validation loss after excluding each of the aforementioned variables. The results show that the thermal conductivity $k(T)$ and the specific heat capacity $C_p(T)$ have a notable impact on both the convergence process and the final predictive accuracy. In particular, the removal of thermal conductivity $k(T)$ significantly degrades the convergence speed and overall model performance, confirming the critical role of thermal conductivity in modelling heat conduction problems, particularly due to the strong nonlinearity of the term $1/k(T)$ within the proposed 'Eng-Gene' module. The final results on both training and test datasets are summarised in Table 3.6. Furthermore, the increased performance gap

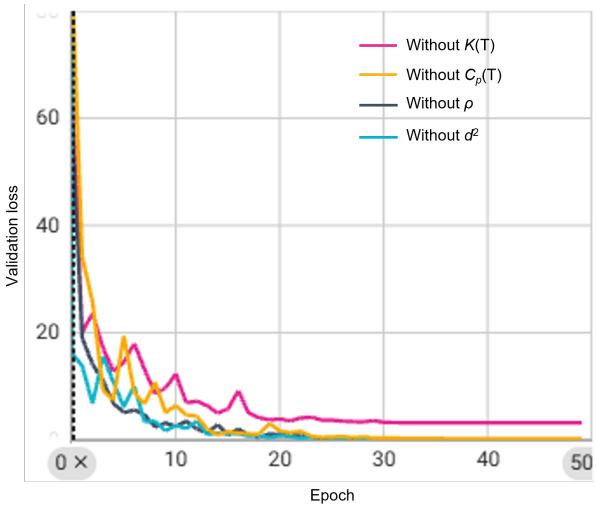


Figure 3.11: Validation loss evolution for 'Eng-Gene' module ablation study.

between the training and test datasets when $k(T)$ is removed indicates its substantial contribution to the generalisation ability of the model.

Table 3.6: Ablation study results on different variables in 'Eng-Gene' module.

Removed variables	Metrics	Training dataset	Test dataset
d^2	MAE	0.037	0.040
	RMSE	0.051	0.054
	MAE-MaxE	0.231	0.243
ρ	MAE	0.047	0.050
	RMSE	0.066	0.070
	MAE-MaxE	0.301	0.319
$C_p(T)$	MAE	0.065	0.068
	RMSE	0.095	0.097
	MAE-MaxE	0.472	0.496
$k(T)$	MAE	0.354	**0.555**
	RMSE	0.377	**0.577**
	MAE-MaxE	0.676	**0.891**

3.5.4 Sensitivity Analysis of Model Parameters

To evaluate the sensitivity and effectiveness of the proposed model, a sensitivity analysis of model parameters was conducted to investigate the impact of key hyperparameters,

including the initial learning rate, the decay rate in the Adam optimiser, and the weighting factor applied in the loss function. Among these, the initial learning rate directly influences the convergence speed and stability of training. Figure 3.12(a) illustrates the results of the experiments conducted with initial learning rates ranging from 10^{-1} to 10^{-3}, covering a range of commonly used values [117, 120]. Larger learning rates accelerated convergence but often caused instability or divergence. Among the tested values, 10^{-2} and 10^{-3} demonstrated a comparable trade-off between convergence speed and stability, yielding similar and satisfactory performance.

The effect of the decay factor in the Adam optimiser was further investigated. Figure 3.12(b) shows that moderate decay rates led to smoother convergence and better generalisation performance, while a very small decay factor resulted in slow convergence. These results suggest that both the initial learning rate and the decay factor significantly influence the convergence behaviour of the model.

Furthermore, the weighting factor β in the loss function regulates the relative importance assigned to the interior computational domain and the boundary conditions. In this work, an initial weighting factor $\beta = 0.25$ is introduced to balance the contributions from the interior and boundary domain. This value is chosen to normalise the coefficient associated with the convolution term $\alpha = 4(-1 + 1/(2Fo))$ and $\alpha' = 4(-1 - 1/(2Fo))$ in the customised kernels (see equations 3.21 and 3.20), ensuring comparable magnitudes between the inner nodes and the boundary nodes. The sensitivity of the results to this parameter was further evaluated under β values of 0.1 and 1.0, as shown in Figure 3.12(c). The model achieves relatively stable and accurate performance when β is set to either 0.25 or 0.1, confirming the critical role of properly weighting the boundary conditions.

Several comparison experiments with different batch sizes, activation functions, padding modes, and loss functions were conducted to assess the impacts of various components in the deep learning models. Table 3.7 summarises the experimental results. It is noteworthy that significant performance degradation occurs when the padding mode is changed to 'Replicate', which may cause disruptions to the initially stable boundary

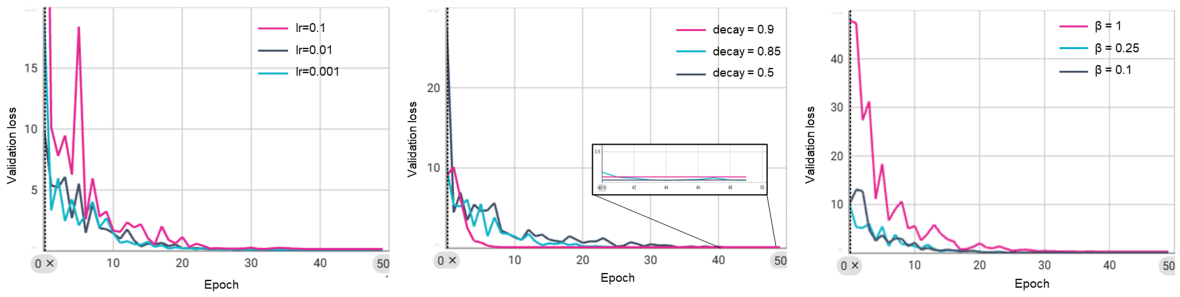


Figure 3.12: Training parameters analysis: (a) initial learning rate; (b) decay factor; (c) loss weighting factor β.

conditions. In addition, the activation function also has a significant impact on the prediction performance. The choice of different loss functions, for example the L2 loss function, has a mild impact on the accuracy of the training set. For different model training batch sizes, a smaller batch size leads to better performance on the test dataset. In this PDEs-solving task, a smaller batch size increases the number of iterations for approximating the physics-informed loss function, which may enhance model training stability.

Table 3.7: Comparison results on test dataset for different components in the framework.

Activation	Padding	Loss	Batch size	MAE	RMSE	MAE-MaxE
GELU	Replicate	L1 loss	1	0.278	0.297	0.582
GELU	Zeros	L2 loss	1	0.089	0.107	0.346
GELU	Zeros	L1 loss	1	**0.029**	**0.048**	**0.313**
RELU	Zeros	L1 loss	1	0.105	0.129	0.442
GELU	Zeros	L1 loss	15	0.048	0.070	0.372

3.5.5 Validation under Various Physical Constraints

A common step in machine learning is model retraining when the feature space of the dataset varies significantly. Transfer learning has been an efficient strategy to fine-tune the pre-trained model for new applications with significantly changed feature space and limited new training data. To verify the effectiveness of the 'Eng-Gene' module in re-training speed while improving the model generalisation performance in the transfer

learning, this section presents the experimental results for transfer learning of physics-informed EngGeneUNet under different operating conditions, where the basic models were first trained under OC1, then re-trained under OC2, OC3, and OC4. Table 3.8 compares the performance of the re-trained deep learning models under different operating conditions using physics-informed EngGeneUNet and UNet, respectively. Firstly, the performance of the fine-tuned models for the varying operating conditions is comparable to the performance of the pre-trained models. Furthermore, it is noteworthy that the PI-EngGeneUNet performs better than PI-UNet under the new operating conditions, achieving more than 90% MAE and RMSE reductions and 60% or above MAE-MaxE reduction on test datasets. This finding confirms the ability of the 'Eng-Gene' module to improve the generalisation ability of the deep learning models.

Table 3.8: Comparison results under different operating conditions in transfer learning.

Base models	Metrics	OC2		OC3		OC4	
		Train	Test	Train	Test	Train	Test
PI-UNet	MAE	0.319	1.216	0.312	0.727	0.447	0.643
	RMSE	0.349	1.246	0.334	0.748	0.483	0.680
	MAE-MaxE	0.778	1.724	0.610	1.010	1.002	1.237
PI-EngGeneUNet	MAE	0.026	**0.028 (↓97%)**	0.035	**0.038 (↓94%)**	0.046	**0.047 (↓92%)**
	RMSE	0.005	**0.012 (↓99%)**	0.027	**0.060 (↓91%)**	0.059	**0.062 (↓90%)**
	MAE-MaxE	0.221	**0.234 (↓86%)**	0.327	**0.366 (↓64%)**	0.268	**0.270 (↓78%)**

Figure 3.13 illustrates a few examples of the performance of the proposed EngGeneNet in different operating conditions. The first column shows the baseline temperature profile yielded by the conventional numerical ADI approach. The second column presents the temperature profile prediction by the proposed physics-informed EngGeneUNet. The third column shows the MAE error map between these two results. It is evident that the predicted 2D temperature profile predicted by the proposed physics-informed EngGeneUNet model is sufficiently close to the results produced by the conventional ADI approach. However, due to the formidable computational complexity of the ADI method for large-scale and high-dimensional problems, the proposed EngGeneUNet model could achieve higher computational efficiency once trained, which is more suitable for real-time applications.

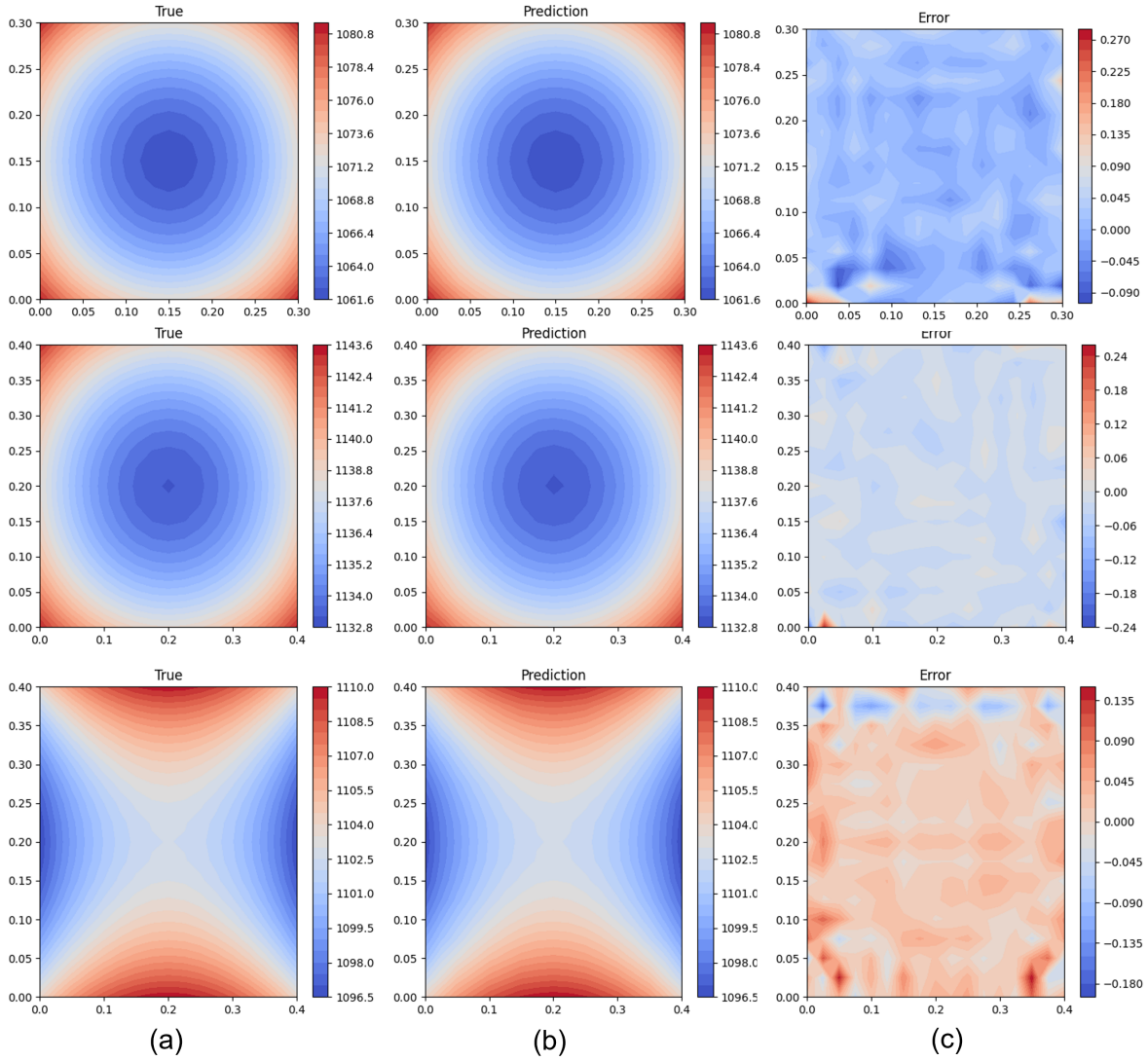


Figure 3.13: Several prediction results on test datasets: (a) ADI solutions; (b) PI-EngGeneNet solutions; (c) error map.

Table 3.9 summarises the model training performance and training time with 30 training epochs and different model training batch sizes under operating condition OC2. The results show that increasing the batch size in the transfer learning does not significantly compromise the model performance for both the training and test datasets. Therefore, the training time in the transfer learning has been significantly reduced by over 75% when the batch size is increased from 1 to 15.

Table 3.9: The influence of batch size for transfer learning under OC2 (30 epochs).

Batch size	Metrics	Training dataset	Test dataset	Training time (saving)
1	MAE	0.038	0.040	115.62 min
	RMSE	0.005	**0.011**	
	MAE-MaxE	0.208	**0.222**	
10	MAE	0.026	0.029	30.52 min
	RMSE	0.005	0.012	
	MAE-MaxE	0.220	0.232	
15	MAE	0.026	**0.028**	27.94 min (75%)
	RMSE	0.005	0.012	
	MAE-MaxE	0.221	0.234	

3.5.6 Transient Temperature Evolution Performance

Evaluating the continuous prediction performance of the proposed physics-informed EngGeneNet is essential, as it poses greater challenges due to the accumulation of prediction errors over time. Therefore, experiments were conducted to compare the mean bulk temperature and temperature distribution uniformity obtained using the conventional ADI method and the proposed physics-informed EngGeneUNet under varying operating conditions. Here, the temperature uniformity is defined as the standard error of the corresponding temperature distribution.

Figure 3.14(a) and (b) illustrate the evolutions of the mean temperature profile and the temperature uniformity for SB1 in a constant ambient environment of 1200 °C. There is a comparable consistency in the two profiles between the two methods. Figure 3.14(c) and (d) show the mean temperature and temperature uniformity evolution for SB2 under varying physical conditions, where the ambient temperature changed from 1000 °C to 1200 °C at $t = 8000$ s. It is noteworthy that the predicted temperature profiles by both the ADI and the EngGeneNet models exhibit consistently similar prediction performance under both conditions. Although the temperature uniformity exhibits a slight difference of about 1 °C in this operating condition, which is within an acceptable error range in this long-period evolution, these findings demonstrate the robustness of the proposed physics-informed EngGeneNet in predicting temperature profiles across

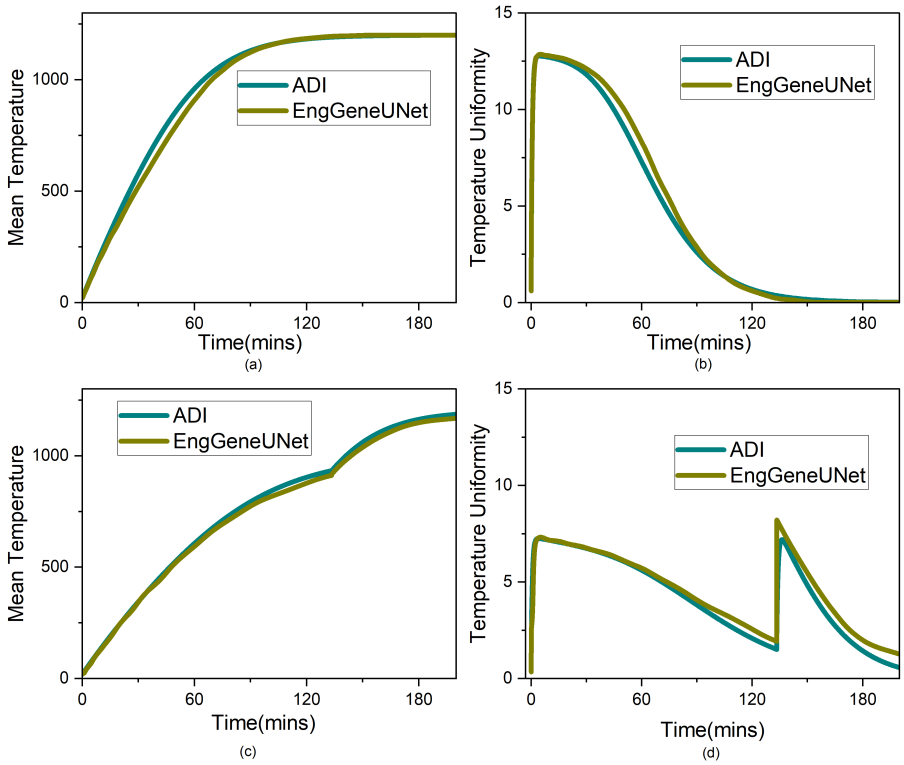


Figure 3.14: Predicted stock temperature history during the trial period.

varying operating conditions, as well as the effectiveness of model fine-tuning through transfer learning.

3.5.7 Model Computational Efficiency

Figure 3.15 compares the computational time for predicting per frame of 2D temperature distribution by both the pre-trained physics-informed EngGeneUNet model and the conventional ADI method. It is shown that the EngGeneUNet model can achieve 96% time reduction compared to the conventional ADI approach in predicting bloom temperature profile, confirming that the EngGeneNet models offer great potential for real-time operation and control.

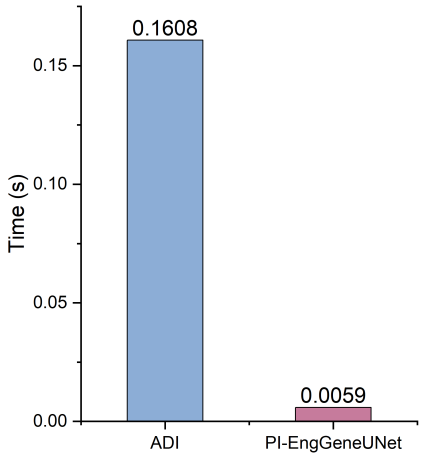


Figure 3.15: Computation time per frame of temperature distribution prediction.

3.6 Chapter Summary

This chapter has developed a physics-informed EngGeneNet for transient temperature profile prediction of large-scale intermediate steel stocks in reheating furnaces, highlighting the significance of incorporating the fundamental heat conduction physics knowledge into the backbone deep learning CNN models and the capability of the 'Eng-Gene' module in capturing salient features and interactions among the temperature-dependent material properties and variables. Formulating the 2D heat conduction modelling as an image-to-image task, the UNet is employed as the backbone of the EngGeneNet, and the physics-informed loss function was constructed by discretising the two-dimensional transient heat conduction governing equations. To improve the interpretability and generalisation performance while accelerating the training of deep learning models, an 'Eng-Gene' module is extracted from first principle equations and integrated into the deep learning model. To address the lack of training samples and the material property variations under different operating conditions, both knowledge distillation and transfer learning are employed [26].

The experimental results demonstrate that:

1. The constructed physics-informed loss function can effectively guide the training of deep learning models, achieving comparable prediction performance to the

numerically expensive ADI approach.

2. The proposed 'Eng-Gene' module improves prediction performance and supports generalisation under the conditions evaluated, while accelerating the training and retraining of deep learning models. It achieves at least 90% MAE reduction and 85% RMSE reduction relative to the physics-informed UNet baseline.

3. The knowledge distillation and transfer learning strategies enable efficient adaptation to new steel product batches with a 75% reduction in training time and greater than 90% reductions in prediction errors under new operating conditions.

4. The deployed model achieves a **96%** reduction in per-frame prediction time compared with the ADI method, enabling genuine real-time temperature field prediction.

Within the formulation investigated in this chapter, physics-informed EngGeneNet provides a flexible function-approximation framework for transient heat-conduction modelling while incorporating the governing physical relationships. The underlying design principles may be transferable to other physics-based modelling problems, subject to reformulation and validation. The results presented here also show that the 'Eng-Gene' module can improve training efficiency and predictive performance for the furnace problem considered.

While this chapter has addressed the modelling problem of achieving real-time, physically consistent temperature field prediction, an accurate model alone is insufficient for many engineering systems, as it must further be embedded within a closed-loop decision-making framework capable of reconciling multiple, often competing, real-time objectives, particularly in safety-critical and performance-driven processes such as battery fast charging. Motivated by this need to bridge modelling and control, the following chapter shifts the focus from passive modelling to active closed-loop control, developing a historical data-driven self-learning MPC framework for the battery single-cell fast charging problem that exploits repetitive operational data to iteratively refine control performance under explicit safety constraints.

CHAPTER 4

HISTORICAL DATA-DRIVEN SELF-LEARNING PREDICTIVE CONTROL FOR REPETITIVE PROCESSES

4.1 Introduction

The battery charging problem studied in this thesis is a representative repetitive industrial control task: the same charging routine is executed repeatedly under similar operating conditions, while each cycle generates a trajectory of state, input, and thermal-response data. In this setting, historical charging data are not merely a passive record; they encode information about previous trajectories, thermal transients, and feasible operating boundaries that can be exploited to improve future decisions. However, conventional MPC formulations typically treat each charging cycle independently and do not make explicit use of the accumulated historical information from prior executions.

Battery charging in electric vehicle (EV) and energy storage applications is therefore a suitable test bed for a self-learning control framework. Lithium-ion batteries are now widely used in ESS [121], smart grids [122], and EVs [123]. In these applications, the battery management system (BMS) must monitor battery states and manage thermal safety while balancing rapid charging against degradation and operational constraints

[124]. Excessive temperature rise can accelerate SEI growth, increase internal resistance, and in severe cases trigger thermal runaway [125]. It is therefore desirable to maintain the operating temperature within a safe range, typically around 25–40 °C [126]. Although numerous charging strategies have been studied, the problem considered here is characterised by a specific trade-off among model fidelity, computational burden, thermal safety, and charging duration. Electrochemical models can be computationally intensive, while simplified or purely data-driven models may lose accuracy or interpretability under varying operating conditions [127, 128]. In addition, the historical trajectories generated by repeated charging cycles are not explicitly incorporated into many conventional formulations. This motivates the development of a self-learning MPC method that exploits previously observed state trajectories to refine charging decisions while maintaining tractability and explicit constraint satisfaction.

The model predictive control (MPC) framework is well-suited to this charging setting because it can explicitly enforce state and input constraints over a finite prediction horizon [64, 129]. Nevertheless, its performance depends strongly on model fidelity and can degrade under uncertainty or parameter variation. Iterative learning control (ILC) is able to exploit historical error information, but it does not natively provide the real-time prediction and constraint handling required for online charging control. Reinforcement learning offers a model-free alternative but typically suffers from poor sample efficiency and high online computational demand.

To address this battery-charging problem, this chapter proposes a historical data-driven self-learning MPC framework that integrates the iterative refinement capability of ILC with the predictive and constraint-handling strengths of MPC. The formulation follows the data-efficient constrained learning idea of Zhou et al. [57], while adapting its value-function approximation and historical convex constraint to the battery-charging setting. The main novelty lies in the development of a memory function and an associated convex mapping constraint derived from historical state trajectories. The memory function quantifies the minimum achievable cumulative future cost inferred from past iterations, while the convex mapping constraint confines the terminal predicted state

to the convex hull of the historical state set. By embedding these constructs into the MPC cost function and constraint set, the proposed framework iteratively improves the charging trajectory while preserving computational tractability and explicit thermal safety constraints.

The specific contributions of this chapter are as follows [27]:

1. A historical state convex mapping constraint is formulated from previous charging trajectories to narrow the feasible solution space and evaluate its effect under the model-uncertainty conditions considered.

2. A memory function-enhanced cost function is proposed to incorporate historical state–input information into the current MPC objective and refine the charging trajectory across iterations.

3. The method is instantiated using a cascading linearised thermoelectric battery model that provides a tractable approximation of the coupled electrical–thermal dynamics for the charging case study.

The remainder of this chapter is organised as follows. Section 4.2 establishes the general MPC problem formulation for linear discrete-time systems. Section 4.3 details the representation of historical data and the derivation of convex mapping constraints. Section 4.4 presents the memory function design and the augmented cost function. Section 4.5 describes the complete self-learning iterative MPC framework and the associated general algorithm. Section 4.6 provides the battery charging case study, including the thermoelectric battery model, its instantiation within the proposed framework, and the experimental results. Section 4.7 summarises the chapter contributions.

4.2 General MPC Problem Formulation

4.2.1 General System Model

Consider a general discrete-time linear system with an additive time-varying offset, describing the repetitive dynamical process to be controlled [130]:

$$x(t+1) = A\,x(t) + B\,u(t) + C(t), \tag{4.1}$$

where $x(t) \in \mathbb{R}^n$ is the system state vector, $u(t) \in \mathbb{R}^m$ is the control input, $C(t) \in \mathbb{R}^n$ is a known time-varying offset (capturing exogenous inputs or operating-point corrections), and $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$ are the system matrices. It is assumed that the system in equation 4.1 is stabilisable and that the state $x(t)$ is fully measurable at each time step.

The framework accommodates systems whose dynamics are either directly linear or can be rendered linear via a suitable input substitution. In the case study presented in Section 4.6, the nonlinear thermal dynamics of a lithium-ion battery are linearised through a variable transformation, resulting in a model of the form in equation 4.1. This structure is general enough to encompass a broad class of repetitive engineering processes, including thermal regulation, chemical batch processing, and robotic motion control.

4.2.2 Standard MPC Problem Formulation

Model predictive control is widely applied in practical engineering systems, where a predictive model is used to estimate future system behaviour and evaluate a cost function over a finite horizon. Based on these predictions, the MPC computes the optimal control inputs to achieve the desired performance.

For the purposes of the self-learning MPC formulation, the general system model in equation 4.1 is adopted as the prediction model. The predicted state $\hat{x}_k(j \mid t)$ and predicted control input $u_k(j \mid t)$ satisfy:

$$\hat{x}_k(j+1 \mid t) = A\,\hat{x}_k(j \mid t) + B\,u_k(j \mid t) + C(t+j), \tag{4.2}$$

where $j = 0, 1, \ldots, N_p - 1$, N_p denotes the prediction horizon, and k is the iteration index. The initial predicted state is $\hat{x}_k(0 \mid t) = x_k(t)$.

The control objective is to regulate the system state to track a desired reference trajectory $x_r(t)$ while penalising excessive control increments. The stage cost function is formulated as:

$$\begin{aligned} J_k(t) &= \sum_{j=0}^{N_p-1} L_k\big(\hat{x}_k(j+1 \mid t) - x_r(t+j),\, \delta u_k(j \mid t)\big) \\ &= \sum_{j=0}^{N_p-1} \Big[\big(\hat{x}_k(j+1 \mid t) - x_r(t+j)\big)^\top Q \big(\hat{x}_k(j+1 \mid t) - x_r(t+j)\big) \\ &\quad + \delta u_k^\top(j \mid t) H\, \delta u_k(j \mid t) \Big], \end{aligned} \tag{4.3}$$

where Q and H are positive-definite weighting matrices balancing tracking performance and input increment cost; x_r denotes the reference state trajectory; and $\delta u_k(j \mid t) = u_k(j \mid t) - u_k(j-1 \mid t)$ is the predicted control increment sequence.

Constraints are inherent in practical engineering systems, with their permissible ranges defined by physical limitations and safety regulations. Constraints on both the system states and control inputs are imposed as:

$$u_{\min} \le u_k(t) \le u_{\max}, \qquad x_{\min} \le x_k(t) \le x_{\max}. \tag{4.4}$$

These can be rewritten in a general compact form as:

$$\Gamma^u u_k(t) \le \Lambda_k^u(t), \qquad \Gamma^x x_k(t) \le \Lambda_k^x(t), \tag{4.5}$$

where

$$\Gamma^u = \begin{bmatrix} I_m \\ -I_m \end{bmatrix}, \qquad \Gamma^x = \begin{bmatrix} I_n \\ -I_n \end{bmatrix}. \tag{4.6}$$

The resulting standard constrained MPC optimisation problem is:

$$\begin{aligned} \min_{\mathbf{u}_k(t)} \; & \sum_{j=0}^{N_p-1} \Big[(\hat{x}_k(j+1 \mid t) - x_r(t+j))^\top Q (\hat{x}_k(j+1 \mid t) - x_r(t+j)) \\ & \quad + \delta u_k^\top(j \mid t) H\, \delta u_k(j \mid t) \Big] \\ \text{s.t. } & \hat{x}_k(j+1 \mid t) = A\hat{x}_k(j \mid t) + B u_k(j \mid t) + C(t+j), \\ & \Gamma^u u_k(t) \le \Lambda_k^u(t), \quad \Gamma^x x_k(t) \le \Lambda_k^x(t). \end{aligned} \tag{4.7}$$

Solving the optimisation problem in equation 4.7 based solely on the predictive model and system constraints may lead to unreliable outcomes, particularly under model uncertainty or when the solution space is restricted by stringent constraints. The following sections describe how historical data is systematically leveraged to augment this standard formulation.

4.3 Historical Data Representation and Convex Mapping Constraints

4.3.1 Motivation for Historical Data Integration

The target class of repetitive processes shares a defining characteristic: each execution proceeds from a fixed initial condition to a target final state under the same physical dynamics and operational constraints. Consequently, the state trajectories and control inputs from previous iterations carry significant and directly applicable information for improving future control decisions. Rather than discarding this information, the proposed framework explicitly harnesses historical operational data by constructing data-driven constraints that refine the feasible solution space at each iteration.

The key idea is to represent the historical state trajectories as a convex set and use this set to constrain the terminal predicted state of the current optimisation. By enforcing the terminal predicted state to lie within the convex hull of historical states, the optimisation is guided toward regions of the state space that have been previously explored, thereby promoting consistent and reliable convergence.

4.3.2 Historical State and Input Datasets

The historical system states and control inputs accumulated over previous iterations are organised into structured datasets. Leveraging the data from the $(k-1)$-th iteration, the historical state set and memory function set are constructed as:

$$\mathbf{X}_{k-1} = \{x_{k-1}(0), x_{k-1}(1), \ldots, x_{k-1}(N)\}, \tag{4.8}$$

$$\mathbf{M}_{k-1} = \{M_{k-1}(0), M_{k-1}(1), \ldots, M_{k-1}(N)\}, \tag{4.9}$$

where N is the total control length in one iteration, and $M_{k-1}(t)$ is the memory function value at time step t of the $(k-1)$-th iteration, whose formulation is detailed in Section 4.4.

4.3.3 Convex Hull Formulation

The theoretical basis for the convex mapping constraint is the notion of a convex hull. A convex combination of a set of points $\{x_j\}$ is a weighted sum of the form:

$$y = \sum_{j=1}^{r} \theta_j x_j, \qquad \sum_{j=1}^{r} \theta_j = 1, \quad \theta_j \geq 0, \quad j = 1, \ldots, r, \tag{4.10}$$

where θ_j are the convex combination coefficients. The convex hull of a set $\mathbf{C}$, denoted $\mathbf{conv}\,\mathbf{C}$, is defined as the smallest convex set containing all elements of $\mathbf{C}$ [131]:

$$\mathbf{conv}\,\mathbf{C} = \left\{ \sum_{j=1}^{r} \theta_j x_j \;\middle|\; x_j \in \mathbf{C}, \theta_j \geq 0, j = 1, \ldots, r, \sum_{j=1}^{r} \theta_j = 1 \right\}. \tag{4.11}$$

Building upon this foundation, localised convex hulls of the historical state and memory function datasets are constructed within a sliding window of size d centred at time step t:

$$\mathbf{conv}\,\mathbf{M}_{k-1}(t) = \left\{ \sum_{j=-d}^{d} \theta_{k-1,j} M_{k-1}(t+j) \right\}, \tag{4.12}$$

$$\mathbf{conv}\,\mathbf{X}_{k-1}(t) = \left\{ \sum_{j=-d}^{d} \theta_{k-1,j} X_{k-1}(t+j) \right\}, \tag{4.13}$$

where d is the window size of the historical data to be employed, and the non-negative convex combination coefficients satisfy:

$$\theta_{k-1,j} \geq 0, \qquad \sum_{j=-d}^{d} \theta_{k-1,j} = 1. \tag{4.14}$$

4.3.4 Convex Mapping Constraint on Terminal Predicted State

To integrate the derived convex hulls into the control framework, a convex hull constraint is imposed on the last predicted state $\hat{x}_k(N_p \mid t)$. This constraint ensures that the terminal

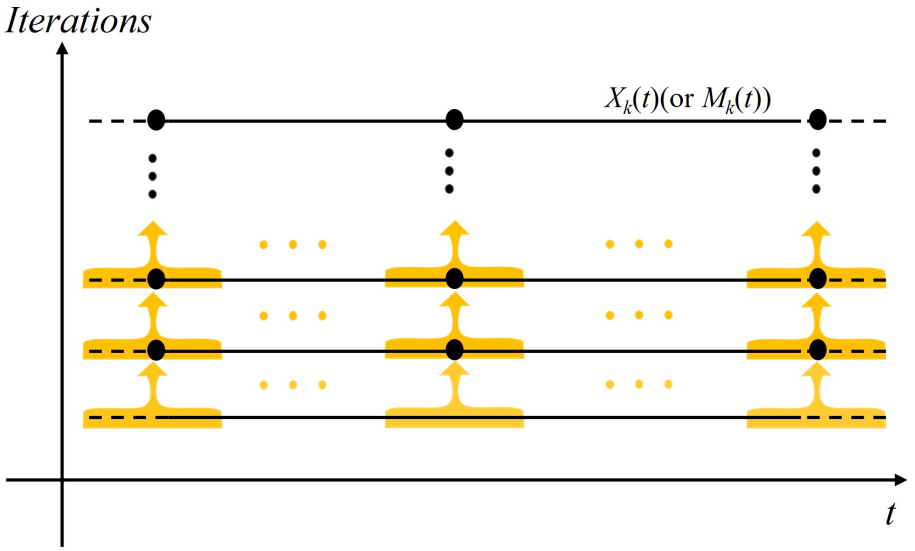


Figure 4.1: Illustration of the historical data convex mapping process.

state in the prediction horizon remains within the convex hull of the historical state set $\mathbf{X}_{k-1}$, thereby contributing to the stability of the closed-loop system:

$$\hat{x}_k(N_p \mid t) = \sum_{j=-d}^{d} \theta_{k-1,j} X_{k-1}(t + N_p + j). \tag{4.15}$$

The physical interpretation of equation 4.15 is illustrated in Figure 4.1: the proposed historical data-driven self-learning control algorithm utilises the convex mapping constraints derived from the historical states and the smallest memory function convex hull within the d-neighbourhood $[t-d, t+d]$ at time step t of the previous iteration to optimise the control input in the current iteration. The convex constraint effectively narrows the feasible search space for the optimisation problem, promoting computational stability and consistent convergence.

To ensure consistency and stability of the proposed algorithm at the boundary time steps, the historical state data and memory function values at the initial and terminal sampling steps are augmented as follows:

$$\begin{cases} X_{k-1}(0+j) = X_{k-1}(0), & -d \le j \le 0, \\ M_{k-1}(0+j) = M_{k-1}(0), & -d \le j \le 0, \\ X_{k-1}(N+j) = X_{k-1}(N), & 0 \le j \le d, \\ M_{k-1}(N+j) = M_{k-1}(N), & 0 \le j \le d. \end{cases} \tag{4.16}$$

This boundary augmentation prevents the convex hull from being ill-defined at the edges of the control horizon, ensuring the constraint in equation 4.15 remains feasible throughout the entire process.

4.4 Memory-Enhanced Objective Design for Self-Learning Control

4.4.1 Design of the Memory Function

To enhance the evaluation of the potential contribution of historical state information and input data in optimising future control input trajectories, a *memory function* is introduced. This function leverages historical system states $x_k(t)$ and input increments $\delta u_k(t)$ to estimate the minimum achievable cumulative cost associated with corresponding state-input pairs.

The memory function is initialised at the terminal time step N of the k-th iteration as:

$$M_k(N) = L_k\big(x_k(N+1), \delta u_k(N)\big) = x_k(N+1)^\top Q\, x_k(N+1) + \delta u_k(N)^\top H\, \delta u_k(N), \quad (4.17)$$

and is then evaluated backward recursively for all preceding time steps:

$$M_k(t) = L_k\big(x_k(t+1), \delta u_k(t)\big) + M_k(t+1), \qquad t < N. \quad (4.18)$$

The backward recursive structure of equations 4.17 and 4.18 implies that the memory function at each time step accumulates the stage costs from that step to the end of the iteration, yielding the total cumulative cost-to-go from that point onward. This structure is conceptually analogous to the cost-to-go function in dynamic programming, but it is computed directly from observed historical data rather than requiring explicit knowledge of the value function.

4.4.2 Memory Function-Augmented Cost Function

The memory function-augmented cost function for the current k-th iteration is obtained by appending the smallest convex hull of the memory function dataset $\mathbf{conv}\, M_{k-1}(t+N_p)$ to the standard MPC cost $J_k(t)$:

$$J'_k(t) = J_k(t) + \sum_{j=-d}^{d} \theta_{k-1,j} M_{k-1}(t+N_p+j), \tag{4.19}$$

where all non-negative coefficients $\theta_{k-1,j}$ are the same as those used in the convex hull of historical states in equation 4.15. This formulation ensures that the convex mapping of the historical state and the augmented memory function term are jointly optimised, maintaining the coupling between the terminal state constraint and the cost augmentation.

The core idea of the memory function-enhanced cost formulation is to evaluate the contribution of the historical dataset and to constrain the terminal predicted state within the resulting convex region using an optimised coefficient. This guarantees that the terminal predicted state always lies within the convex hull of the historical state set, enabling continuous refinement of the control input trajectory. Importantly, the memory function is integrated in a way that preserves the standard quadratic MPC cost structure and maintains the generalisability of the loss function. Furthermore, the memory function term is constructed from stored historical data that is known at each computation step and does not introduce additional uncertainty into the optimisation.

4.4.3 Self-Learning Mechanism

The iterative self-learning mechanism operates as follows. At each iteration k and time step t, the augmented optimisation problem in Section 4.5 is solved to obtain the optimal control input. The resulting state trajectory is stored and used to update the historical state set $\mathbf{X}_k$ and the memory function set $\mathbf{M}_k$ for the next iteration. As k increases, the historical data progressively captures more refined and representative state trajectories, and the memory function provides increasingly accurate estimates of the cumulative

future cost. This iterative feedback mechanism enables continuous refinement of the control policy and drives stable convergence of the process toward the optimal solution.

4.5 Self-Learning Iterative MPC: General Framework and Algorithm

4.5.1 Complete Augmented Optimisation Problem

Building upon the standard MPC formulation in equation 4.7 and the historical data-driven constructs developed in Sections 4.3 and 4.4, the complete memory function-enhanced self-learning optimisation problem is formulated. At each time step t of the k-th iteration, the optimisation variables are the predicted control input sequence $\mathbf{u}_k(t)$ and the convex combination coefficients $\theta_{k-1,j}$:

$$\begin{aligned}
\min_{\theta_{k-1,j},\,\mathbf{u}_k(t)} \; & J_k(t) + \sum_{j=-d}^{d} \theta_{k-1,j} M_{k-1}(t+N_p+j) \\
\text{s.t. } & \hat{x}_k(j+1 \mid t) = A\hat{x}_k(j \mid t) + Bu_k(j \mid t) + C(t+j), \\
& \Gamma^u u_k(t) \le \Lambda_k^u(t), \quad \Gamma^x x_k(t) \le \Lambda_k^x(t), \\
& \theta_{k-1,j} \ge 0, \quad \sum_{j=-d}^{d} \theta_{k-1,j} = 1, \\
& \hat{x}_k(N_p \mid t) = \sum_{j=-d}^{d} \theta_{k-1,j} X_{k-1}(t+N_p+j).
\end{aligned} \tag{4.20}$$

The optimal control input sequence $\mathbf{u}_k^*(t)$ is obtained by solving equation 4.20. Following the receding-horizon principle of MPC, only the first element of this sequence is applied to the system:

$$u_k^*(t) = [1, 0, \ldots, 0]\,\mathbf{u}_k^*(t). \tag{4.21}$$

The augmented problem in equation 4.20 is a convex quadratic programme (QP) because: (i) the objective function is quadratic in $\mathbf{u}_k(t)$ and the convex memory term is linear in $\theta_{k-1,j}$; (ii) all constraints are affine. The incorporation of the convex mapping constraint effectively narrows the feasible solution space without altering the convex

Algorithm 1 Historical Data-Driven Self-Learning MPC for Repetitive Processes

Require: K: number of iterations;
N: control length in one iteration;
N_p: MPC prediction horizon;
$(u_0, u_1, \ldots, u_N)$: initial control input sequence;
$(x_0, x_1, \ldots, x_N, x_{N+1})$: initial state sequence;
Q, H: cost function weighting matrices;
d: convex hull window size.
for $k = 1$ **to** K **do**
    $x_k(0) \leftarrow x_0$; $u_k(0) \leftarrow u_0$
    if $k = 1$ **then**
        Compute initial memory function set $\mathbf{M}_0$ and state set $\mathbf{X}_0$ from the initial state sequence using equations 4.17 and 4.18.
    end if
    for $t = 1$ **to** N **do**
        Solve the augmented optimisation problem 4.20 to obtain $\mathbf{u}_k^*(t)$.
        Apply $u_k^*(t)$ to the system model 4.1 to obtain $x_k(t+1)$.
        Collect updated state $x_k(t+1)$.
    end for
    Compute terminal memory value: $M_k(N) \leftarrow x_k(N+1)^\top Q\, x_k(N+1) + \delta u_k(N)^\top H\, \delta u_k(N)$.
    Backward recursion: $M_k(t) \leftarrow L_k(x_k(t+1), \delta u_k(t)) + M_k(t+1),\ t < N$.
    Update historical sets:
        $\mathbf{X}_k \leftarrow \{x_k(0), x_k(1), \ldots, x_k(N)\}$
        $\mathbf{M}_k \leftarrow \{M_k(0), M_k(1), \ldots, M_k(N)\}$
end for

structure of the problem, thereby reducing the worst-case solve time variability compared to standard MPC. Furthermore, since the memory function term is computed from previously observed data and the convex hull is constructed from a finite window of historical states, the computational overhead per optimisation step relative to standard MPC is negligible.

4.5.2 General Self-Learning Algorithm

Algorithm 1 summarises the complete procedure of the proposed historical data-driven self-learning MPC framework for a general repetitive process.

The framework jointly optimises the control input sequence $\mathbf{u}_k(t)$ and the convex

combination coefficients $\theta_{k-1,j}$ within a single quadratic programme at each time step. Application of the general framework to a specific repetitive process requires: (i) identifying a suitable system model of the form in equation 4.1; (ii) defining appropriate state and input constraints; and (iii) specifying an initial feasible state and input sequence. Section 4.6 demonstrates this instantiation in full detail for the battery charging problem.

4.6 Case Study: Iteratively Optimised Battery Charging under Thermal Constraints

This section instantiates the proposed self-learning MPC framework on the battery charging problem. The section is organised as follows. Section 4.6.1 develops the cascading linearised thermoelectric battery model that serves as the prediction model within the framework. Section 4.6.2 then describes how the general algorithm is specialised for the battery charging problem, presenting the complete control architecture and the battery-specific algorithm. The remaining subsections present the simulation results and analysis.

4.6.1 Cascading Linearised Thermoelectric Battery Model

The battery charging process is inherently complex, involving coupled interactions among several key variables, including the state of charge (SOC), terminal voltage, and internal temperature. A major challenge in developing model-based charging control strategies lies in the strong nonlinear coupling between the electrical and thermal dynamics. To address this issue, a cascading linearised thermoelectric battery model is developed to decouple the electrical and thermal sub-models while retaining the essential physical relationships required for accurate constraint enforcement.

4.6.1.1 Battery Electrical Sub-Model

Models for $LiFePO_4$ battery cells can be categorised into electrochemical models, equivalent circuit models, and data-driven neural models. While electrochemical models such as the pseudo-two-dimension (P2D) and single particle model (SPM) offer high accuracy, their computational complexity makes them unsuitable for real-time control [132]. Neural network-based models offer computational efficiency but limited interpretability. In contrast, the equivalent circuit model offers a simple structure and straightforward parameters identification while retaining comparable accuracy [86].

In this study, a first-order Thevenin equivalent circuit model is adopted. As illustrated in Figure 4.2, U_{ocv} denotes the open-circuit voltage of the battery when no external load is applied. V and i represent the terminal voltage and charging current, respectively. R_0 is the internal resistance. The RC sub-circuit characterises the battery relaxation effect. Based on this circuit model, the corresponding discrete-time formulations are:

$$\text{soc}(t) = \text{soc}(t-1) - \frac{\Delta t}{C_n} i(t-1) = \text{soc}(t_0) - \frac{\Delta t}{C_n} \sum_{j=t_0}^{t} i(j), \tag{4.22}$$

$$V(t) = U_{ocv} - V_1 - i(t)R_0, \tag{4.23}$$

where Δt is the sampling interval in seconds, C_n is the nominal capacity in ampere-seconds, and t_0 is the initial time step. The charging current $i(t)$ is defined as negative in this study. The open-circuit voltage U_{ocv} is modelled as a function of SOC:

$$U_{ocv}(t) = f_{ocv}(\text{soc}(t)), \tag{4.24}$$

and the internal resistance R_0 is represented as a function of the internal battery temperature T_{in}:

$$R_0 = f_R(T_{in}(t)), \tag{4.25}$$

where f_{ocv} and f_R are implemented using look-up tables as detailed in Section 4.6.3.

A first-order RC sub-circuit is employed to represent the slow-varying diffusion voltage within the battery cell, under the assumption that the load current remains constant over each sampling interval. The voltage V_1 is governed by:

$$V_1(t) = a_1 V_1(t-1) + b_1 i(t-1), \tag{4.26}$$

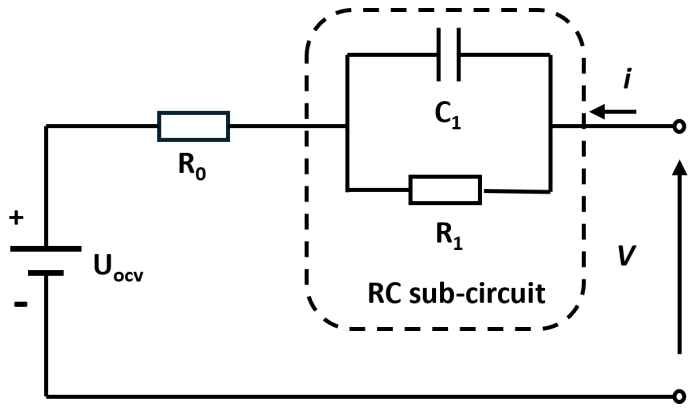


Figure 4.2: First-order Thevenin equivalent circuit model of the lithium-ion battery.

where

$$a_1 = e^{-\Delta t/(R_1 C_1)}, \qquad b_1 = R_1\left(1 - e^{-\Delta t/(R_1 C_1)}\right). \tag{4.27}$$

Combining the above equations, the complete discrete-time electric sub-model is:

$$\begin{cases} \text{soc}(t) = \text{soc}(t-1) - \dfrac{\Delta t}{C_n} i(t-1), \\ V_1(t) = a_1 V_1(t-1) + b_1 i(t-1), \\ V(t) = U_{ocv} - i(t) f_R(T_{in}(t)) - V_1(t). \end{cases} \tag{4.28}$$

4.6.1.2 Battery Thermal Sub-Model

During battery operation, heat is generated within the cell and conducted to the surface before dissipating into the surrounding environment. Temperature is a critical factor affecting both battery performance and degradation.

The dominant heat source within the operating range is ohmic heating from the internal resistance:

$$Q_J = i(t)^2 R_0. \tag{4.29}$$

The contributions from electrolyte decomposition and side reactions are negligible when the battery operates within safe voltage and temperature ranges [133].

A simplified lumped thermal model is adopted, assuming uniform temperature across the internal and surface regions and a spatially uniform heat generation rate. Based on

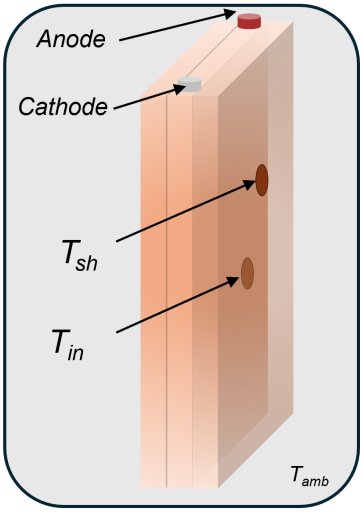


Figure 4.3: Lumped two-node battery thermal model.

the energy balance equation, the continuous-time thermal dynamics are:

$$\begin{cases} C_{p1}\dfrac{dT_{in}}{dt} = Q_J + k_1(T_{sf} - T_{in}), \\ C_{p2}\dfrac{dT_{sf}}{dt} = k_1(T_{in} - T_{sf}) + k_2(T_a - T_{sf}), \end{cases} \tag{4.30}$$

where T_{in} and T_{sf} denote the battery internal and surface temperatures, respectively; T_a is the ambient temperature; C_{p1} and C_{p2} are the heat capacities of the battery interior and shell; k_1 is the heat transfer coefficient between the battery and the shell; and k_2 is the heat transfer coefficient between the battery shell and the environment. The thermal model is illustrated in Figure 4.3.

Discretising using the forward Euler method, the discrete-time thermal dynamics are:

$$\begin{cases} T_{in}(t+1) = \left(1 - \Delta t\dfrac{k_1}{C_{p1}}\right)T_{in}(t) + \Delta t\dfrac{k_1}{C_{p1}}T_{sf}(t) + \dfrac{\Delta t}{C_{p1}}i(t)^2R_0, \\ T_{sf}(t+1) = \Delta t\dfrac{k_1}{C_{p2}}T_{in}(t) + \left(1 - \Delta t\dfrac{k_1+k_2}{C_{p2}}\right)T_{sf}(t) + k_2T_a\dfrac{\Delta t}{C_{p2}}. \end{cases} \tag{4.31}$$

4.6.1.3 Cascading Linearised Integration

In the proposed cascaded linearised thermoelectric model, the nonlinear coupling between the thermal dynamics and the electrical subsystem is captured by the substitution $u(t) = i(t)^2$, where $u'(t) = i(t)$ denotes the actual charging current. This transformation renders

the thermal dynamics linear with respect to the new input $u(t)$. The two linearised sub-models are then hierarchically integrated within the control framework.

Thermal sub-model (primary control loop):

$$x_T(t+1) = A_T\, x_T(t) + B_T\, u(t) + C_T(t), \tag{4.32}$$

where $x_T(t) = [T_{in}(t),\, T_{sf}(t)]^\top$, $u(t) = i(t)^2$, $C_T = [0,\, k_2 T_a \Delta t / C_{p2}]^\top$, and

$$A_T = \begin{bmatrix} 1 - \Delta t \dfrac{k_1}{C_{p1}} & \Delta t \dfrac{k_1}{C_{p1}} \\ \Delta t \dfrac{k_1}{C_{p2}} & 1 - \Delta t \dfrac{k_1 + k_2}{C_{p2}} \end{bmatrix}, \qquad B_T = \begin{bmatrix} \dfrac{\Delta t R_0}{C_{p1}} \\ 0 \end{bmatrix}. \tag{4.33}$$

Electric sub-model (subordinate loop):

$$\begin{cases} x_E(t+1) = A_E\, x_E(t) + B_E\, u'(t), \\ V(t) = f_{ocv}(\mathrm{soc}(t)) - i(t)\, f_R(T_{in}(t)) - V_1(t), \end{cases} \tag{4.34}$$

where $x_E(t) = [\mathrm{soc}(t),\, V_1(t)]^\top$, $u'(t) = i(t)$, and

$$A_E = \begin{bmatrix} 1 & 0 \\ 0 & a_1 \end{bmatrix}, \qquad B_E = \begin{bmatrix} -\Delta t / C_n \\ b_1 \end{bmatrix}. \tag{4.35}$$

The cascaded relationship $u(t) = [u'(t)]^2 = i(t)^2$ links the two sub-models, ensuring that the internal temperature and electrical states are simultaneously regulated while maintaining convex tractability in the optimisation problem. The thermal sub-model 4.32 is of exactly the form required by the general system model in equation 4.1, with $A = A_T$, $B = B_T$, and $C(t) = C_T(t)$.

4.6.2 Instantiation of the Self-Learning MPC Framework for Battery Charging

With the thermoelectric battery model established, the general self-learning MPC framework in Section 4.5 is specialised to the battery charging problem. The instantiation proceeds in two steps: the control architecture is defined, and the general algorithm is adapted to account for the cascaded model structure.

Control Framework Architecture. Figure 4.4 intuitively illustrates the overall control framework. The battery thermal sub-model serves as the primary model within the self-learning MPC framework, while the electric sub-model functions as a subordinate model for monitoring the battery electrical states. Accordingly, the thermal sub-model states are directly formulated into the cost function of the MPC framework. In contrast, constraints from the electric sub-model — specifically those on terminal voltage and SOC — are enforced by converting the state constraints into equivalent constraints on the charging current, thereby regulating the direct control input $u_k(t)$ and ensuring safe and efficient charging operation across both thermal and electrical domains.

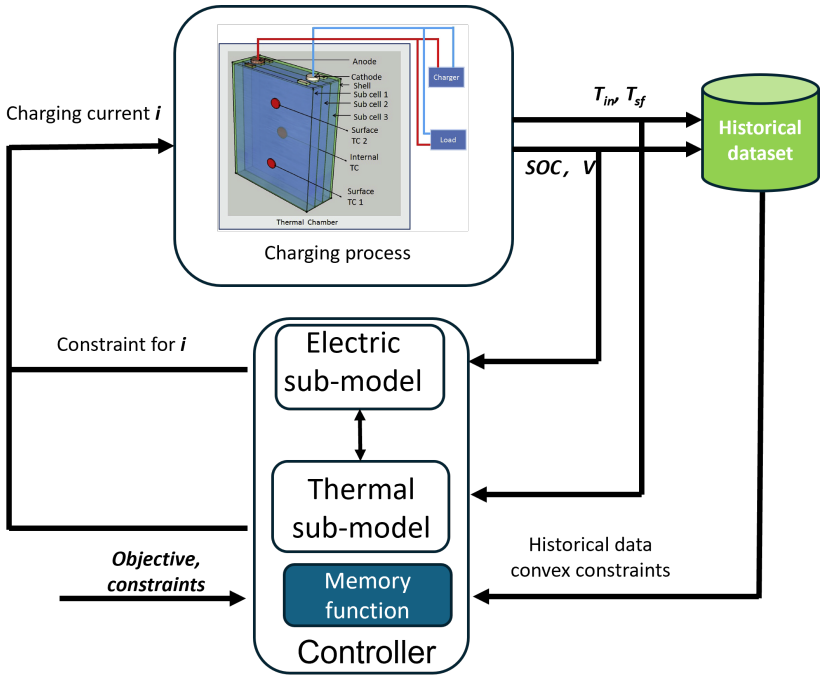


Figure 4.4: Overall control framework of the proposed historical data-driven self-learning MPC for battery charging.

Since the thermal sub-model is adopted as the primary control model and the optimisation variable is defined as $u(t) = i(t)^2$, the corresponding actual charging current is recovered as:

$$u_k'^*(t) = i(t) = -\sqrt{u_k^*(t)}, \tag{4.36}$$

where the negative sign reflects the convention that charging current is negative in this study. This recovered current is then applied to the electric sub-model to update the

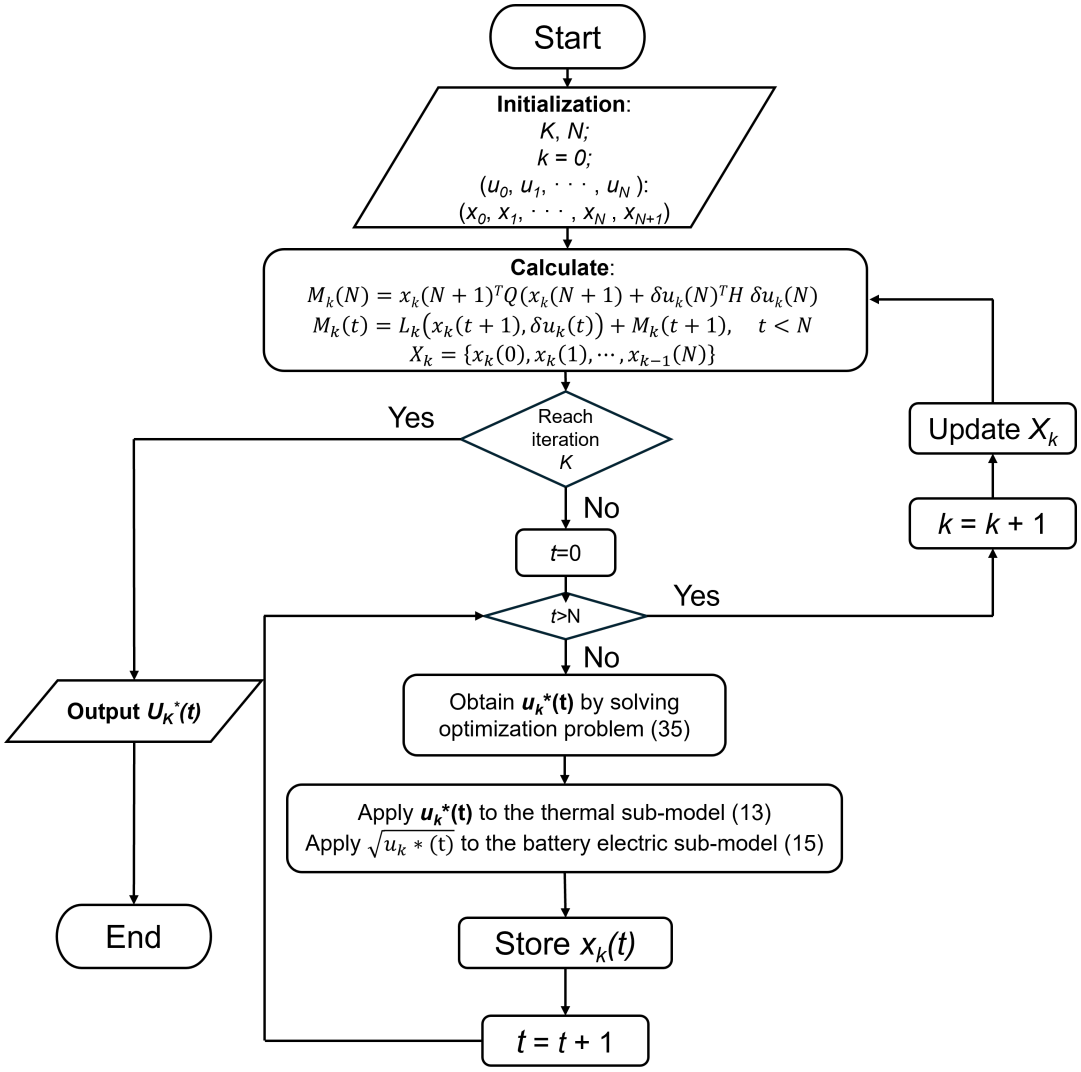


Figure 4.5: Flowchart of the proposed historical data-driven self-learning control framework applied to battery charging.

electrical states and enforce the voltage constraint.

Battery Charging Algorithm. Figure 4.5 presents the flowchart of the proposed historical data-driven self-learning control framework instantiated for battery charging. At each iteration, the control input is updated based on the historical state information and the resulting system response, which is subsequently stored and fed back to initialise the next iteration.

4.6.3 Experimental Setup and Battery Model Parameters

The parameters used in the thermoelectric battery model were adopted from [134, 135], in which identical electrical and thermal sub-model structures were employed. In those works, a comprehensive parameter identification was experimentally conducted on a prismatic $LiFePO_4$-Graphite battery with a nominal capacity of 10 Ah and a nominal voltage of 3.2V. Since embedding micro-sensors within the cell core is experimentally challenging, internal temperature measurements were approximated using a three-sub-cell configuration in which both central and surface temperatures were instrumented. The identified parameters were validated through direct comparison between model simulations and experimental measurements, achieving a maximum voltage RMSE of approximately 3.4 mV and an internal temperature RMSE of approximately 0.82°C. The complete experimental setup and measurement procedures are documented in detail in [134].

The identified model parameters are summarised in Table 4.1. The functions f_{ocv} and f_R are obtained from look-up tables using linear interpolation, with reference to Tables 4.2 and 4.3. In all simulations, the ambient temperature was maintained at 27°C, while both the initial internal and surface temperatures were set to 29°C.

Table 4.1: Identified thermoelectric battery model parameters.

Parameter	Value	Unit
a_1	0.981	—
b_1	1.8 e-4	—
k_1	1.264	$\mathrm{Wm^{-1}K^{-1}}$
k_2	0.33	$\mathrm{Wm^{-1}K^{-1}}$
C_{p1}	263.8	$\mathrm{Jkg^{-1}K^{-1}}$
C_{p2}	31.2	$\mathrm{Jkg^{-1}K^{-1}}$
C_n	36000	As

Table 4.2: SOC–U_{ocv} look-up table.

SOC	0.900	0.798	0.695	0.593	0.491	0.389	0.287	0.186	0.085
U_{ocv} (V)	3.330	3.325	3.299	3.292	3.290	3.278	3.251	3.215	3.057

Table 4.3: T_{in}–R_0 look-up table.

T_{in} (°C)	−10	0	10	23	32	39	52
R_0 (Ω)	0.0259	0.0180	0.0164	0.0152	0.0125	0.0124	0.0120

4.6.4 Constraint Formulation for Battery Charging

According to the safe operating limits of $LiFePO_4$ batteries, constraints on the charging current, terminal voltage, and SOC are imposed throughout the charging process, with the SOC required to increase from 0.1 to 0.9. The maximum charging current is limited to 30 A (3C rate). Since the optimisation variable is defined as $u(t) = i(t)^2$ in the thermal sub-model, the original bound $-30\,\mathrm{A} \le i(t) \le 0\,\mathrm{A}$ is equivalently expressed as:

$$0 \le u(t) \le 900. \tag{4.37}$$

The terminal voltage constraint $2.6\,\mathrm{V} \le V(t) \le 3.65\,\mathrm{V}$ is reformulated into an input-dependent constraint through the electric sub-model. Through the relation equation 4.28, the upper voltage limit $V(t) \le 3.65\,\mathrm{V}$ yields:

$$i(t) \ge \frac{U_{ocv}(\mathrm{soc}(t)) - V_1(t) - 3.65}{f_R(T_{in}(t))}. \tag{4.38}$$

Since $i(t) = -\sqrt{u(t)}$, this is converted into an admissible bound on $u(t)$:

$$u(t) \le \left(\frac{U_{ocv}(\mathrm{soc}(t)) - V_1(t) - 3.65}{f_R(T_{in}(t))}\right)^2. \tag{4.39}$$

All quantities on the right-hand side are known at the current time step, so the resulting constraint remains linear with respect to the optimisation variable and can be directly enforced within the MPC framework.

The SOC constraint is defined as:

$$0.1 \le \mathrm{soc}(t) \le 0.9. \tag{4.40}$$

By reformulating the current and voltage limits into input-compatible expressions, all safety requirements are enforced explicitly within the MPC solver without introducing nonlinear state constraints, thereby maintaining both computational tractability and operational safety throughout the charging process.

4.6.5 Validation of Cascaded Model Constraints

To evaluate the impact of state constraints from the cascaded sub-model, a comparative study was performed with and without enforcing the terminal-voltage constraint. The controller was configured with $K = 10$ iterations, prediction horizon $N_p = 15$, weighting factors $Q = \text{diag}[10, 0]$ and $H = 20$, and an internal temperature constraint of 38 °C. The results are shown in Figure 4.6.

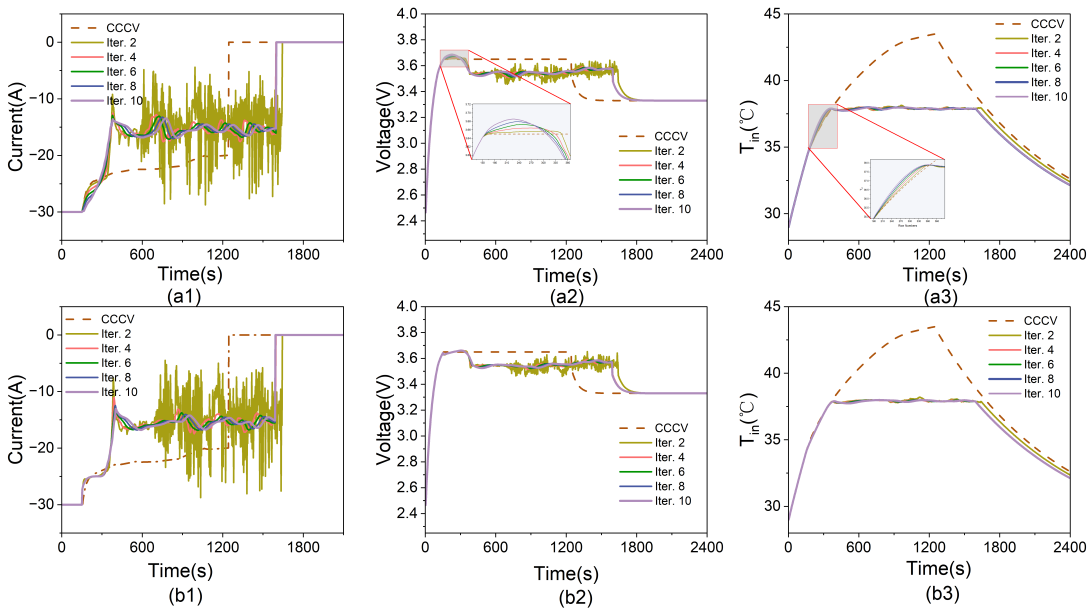


Figure 4.6: Comparative results: (a) without terminal voltage constraint; (b) with terminal voltage constraint. ($N_p = 15$, $H = 20$, $T_{in}^{\max} = 38\,^{\circ}\text{C}$)

When the terminal-voltage constraint is not applied (Figure 4.6(a)), the charging current trajectories become progressively smoother across iterations, demonstrating the self-learning capability of the proposed framework. However, the terminal voltage exceeds the allowable limit (Figure 4.6(a2)) because the controller prioritises rapid charging. In contrast, when the voltage constraint is enforced (Figure 4.6(b)), the terminal voltage remains consistently below the prescribed upper bound, while the internal temperature stabilises upon reaching the threshold.

These results confirm both the effectiveness and necessity of incorporating state constraints from the cascaded electrical sub-model, and demonstrate the stable convergence

and strong self-learning capability of the proposed historical data-driven convex-mapping strategy.

4.6.6 Effect of Prediction Horizon

The prediction horizon N_p plays a crucial role in balancing control performance and computational burden. Figure 4.7 displays the system response under $N_p \in \{15, 30, 45\}$ in a matrix format, where rows (a)–(c) correspond to the three horizon values and columns (1)–(3) represent the optimised charging current trajectory, terminal voltage, and internal temperature evolution over 10 iterations.

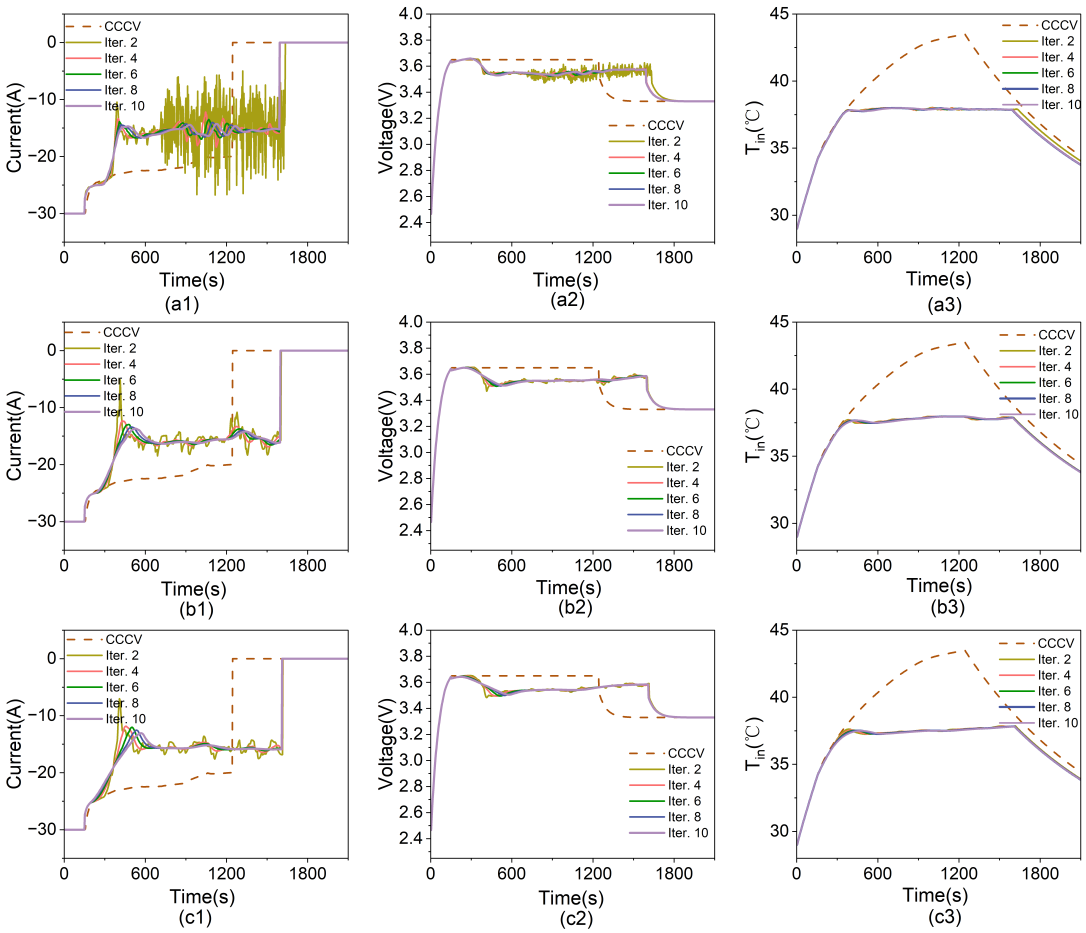


Figure 4.7: Effect of different prediction horizons: (a) $N_p = 15$; (b) $N_p = 30$; (c) $N_p = 45$. ($H = 10$, $T_{in}^{\max} = 38\,^{\circ}\mathrm{C}$)

Increasing N_p noticeably reduces the fluctuation of the optimised charging current, leading to improved control stability. At $N_p = 45$, the charging current trajectories exhibit considerable stability even in the early stages of the iteration process, achieving faster convergence compared to $N_p = 15$. Due to the constraints imposed by the cascaded electric sub-model, all terminal voltage outputs are consistently maintained below 3.65V across all horizon settings. While larger horizons enhance stability and convergence, they also increase computational cost. Therefore, an appropriate balance between control performance and computational efficiency should be determined according to the specific application requirements.

4.6.7 Analysis of Weighting Factors

The cost function is formulated to balance control performance in regulating the internal temperature and optimising charging efficiency. A comparative analysis is conducted using three configurations: $H \in \{10, 20, 40\}$, with fixed $N_p = 15$ and $T_{in}^{\max} = 40\,^{\circ}\mathrm{C}$. Results are presented in Figure 4.8.

As the weighting factor H for the input increment cost increases, the algorithm increasingly prioritises charging stability during the optimisation of the charging current trajectory. A larger H leads to a more stable optimised charging current trajectory due to the increased penalty on δu in the cost function. These results demonstrate the adaptability of the proposed historical data-driven self-learning control algorithm in balancing various control objectives.

4.6.8 Robustness under Parameter Uncertainty

To evaluate the robustness of the proposed method against battery parameter uncertainties, two levels of stochastic perturbations using Beta distributions were applied to the internal resistance and capacity parameters to account for ageing and inter-cell variability:

$$C' \sim C_n \cdot (1 - \mathrm{Beta}(2,5) \cdot k_\beta), \tag{4.41}$$

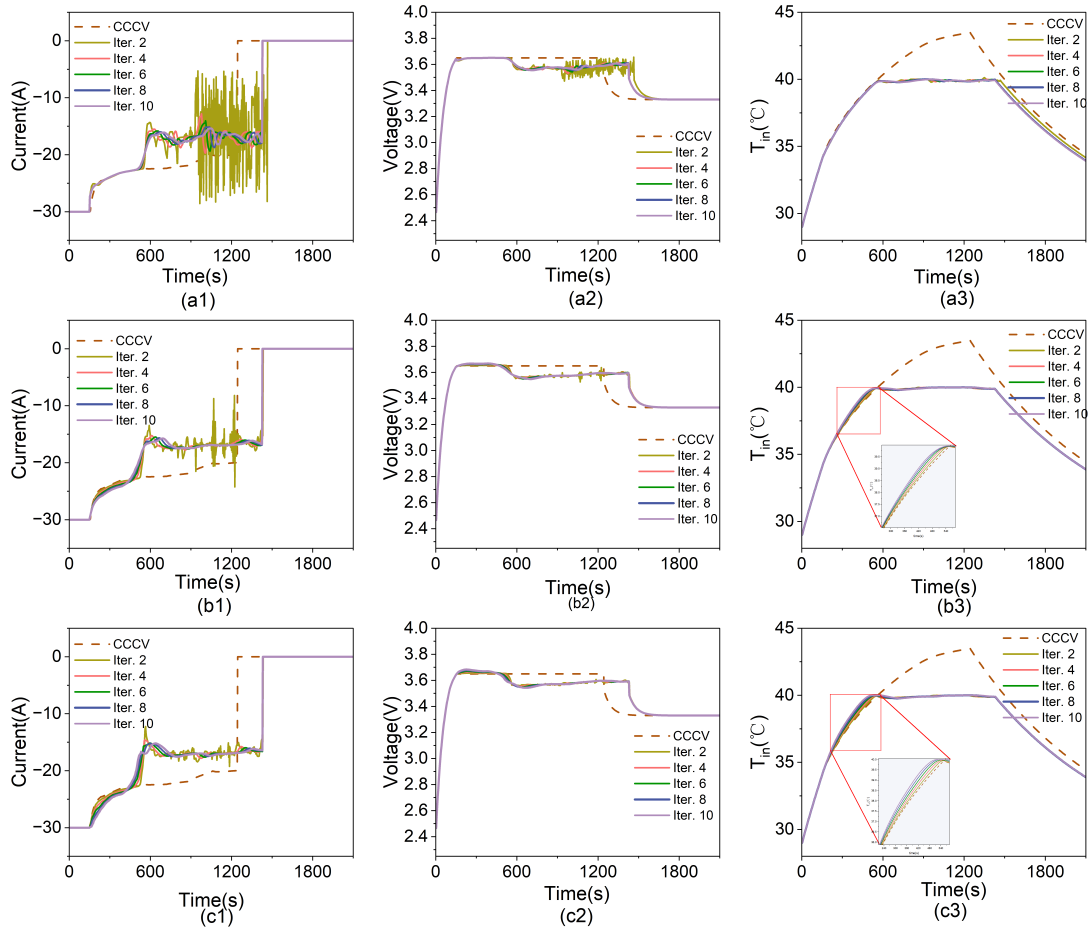


Figure 4.8: Effect of weighting factor on control performance: (a) $H = 10$; (b) $H = 20$; (c) $H = 40$. ($N_p = 15$, $T_{in}^{\max} = 40\,^{\circ}\mathrm{C}$)

$$R_0' \sim R_0 \cdot (1 + \mathrm{Beta}(2,5) \cdot k_\beta), \tag{4.42}$$

where $k_\beta \in \{5\%, 10\%\}$ denotes the uncertainty level. The Beta distribution naturally aligns with the physical constraints of battery parameters, such as bounded internal resistance and capacity variations within finite ranges [64].

As shown in Figure 4.9, the internal temperature remains strictly below the specified threshold of 37 °C under both uncertainty levels, demonstrating consistent thermal management performance. The voltage responses confirm that the algorithm effectively maintains the terminal voltage within the safe upper bound of 3.65 V. Although the current profiles under 10% noise exhibit more pronounced fluctuations during early

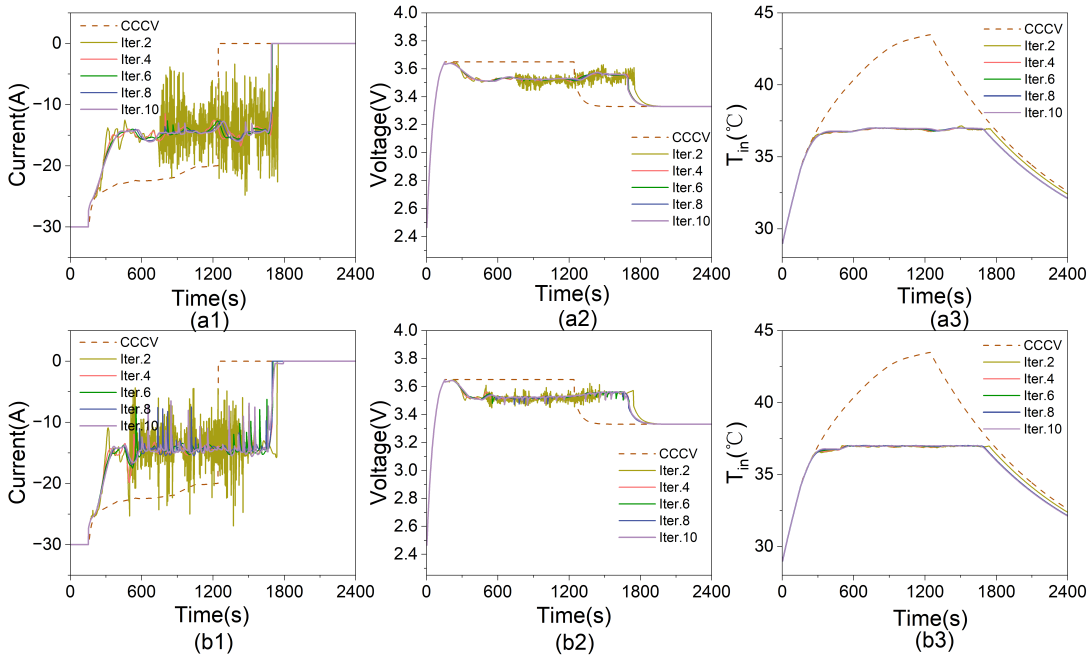


Figure 4.9: Robustness under parameter uncertainty: (a) 5% Beta uncertainty; (b) 10% Beta uncertainty. ($T_{in}^{\max} = 37\,^{\circ}\mathrm{C}$)

iterations, both scenarios ultimately converge to stable and smooth current trajectories. These results verify the inherent robustness of the proposed approach to significant parameter uncertainties.

4.6.9 Sensitivity to Initial Solutions

To investigate the sensitivity of the proposed method to different initial solutions, a comparative study was conducted under a stringent thermal constraint of 34 °C, representing only a 5 °C margin above the initial battery temperature of 29 °C. Two distinct charging profiles were adopted as initial references: the conventional CC-CV charging profile and a pure MPC-solved profile.

The results in Figure 4.10 reveal notable differences between the two initialisation strategies. When initialised with the CC-CV profile, the method experiences mild convergence difficulties during early iterations, as reflected by fluctuations in the charging current and insufficient temperature regulation. Nevertheless, the algorithm gradually

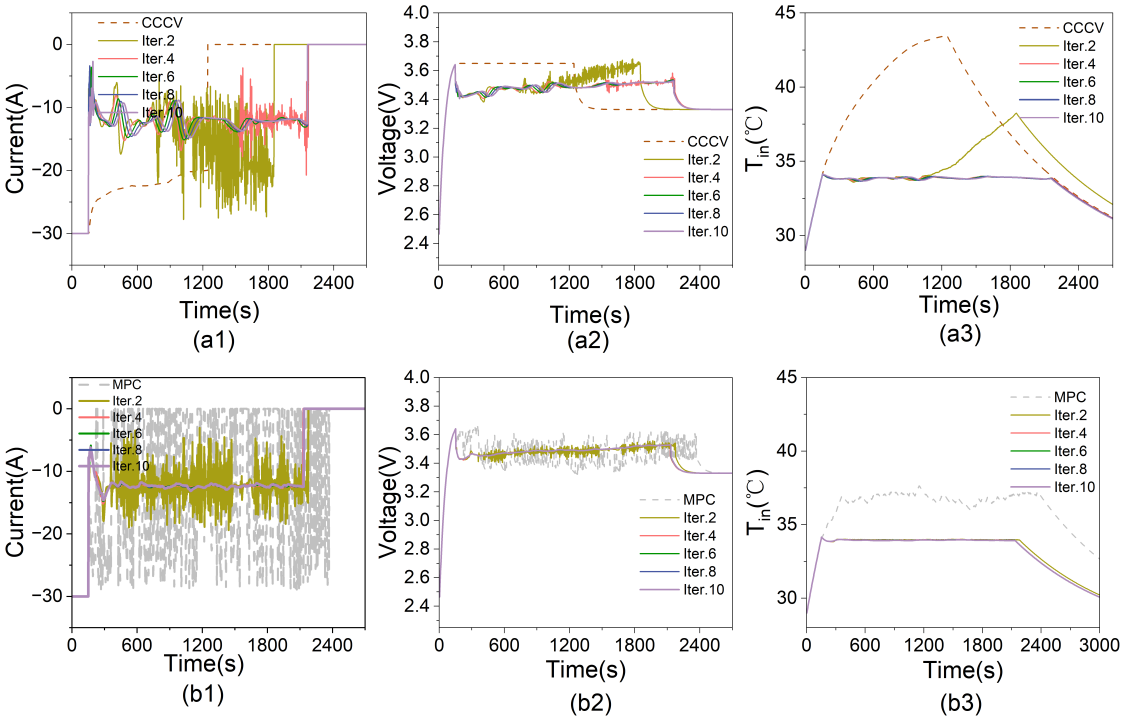


Figure 4.10: Sensitivity analysis of initial solutions: (a) CC-CV initialisation; (b) standard MPC initialisation. ($T_{in}^{\max} = 34\,^{\circ}\mathrm{C}$)

stabilises and achieves effective temperature control in subsequent iterations. When initialised with the MPC profile, the proposed method quickly converges toward a smooth and stable charging trajectory, successfully limiting the internal temperature at the specified threshold.

This comparative analysis confirms that the initialisation strategy plays a critical role in determining convergence performance, especially when the feasible region is severely restricted by stringent thermal limits. An initial solution that provides a more balanced trade-off between charging intensity and thermal safety is preferable, as it facilitates more robust convergence and ensures effective satisfaction of the imposed constraints.

Table 4.4: Computation time statistics for different prediction horizons (unit: seconds).

Horizon Length	Standard MPC				Proposed Method			
	Mean	Std	Max	Min	Mean	Std	Max	Min
5	0.094	0.015	0.140	0.081	0.086	**0.004 (↓73%)**	0.096	0.081
10	0.110	0.017	0.158	0.097	0.105	**0.003 (↓82%)**	0.120	0.099
20	0.143	0.019	0.202	0.129	0.138	**0.004 (↓79%)**	0.156	0.131
40	0.199	0.025	0.291	0.179	0.193	**0.005 (↓80%)**	0.221	0.184

Table 4.5: Computation time standard deviation comparison at $N_p = 10$.

Method	Standard Deviation (s)
SMPC [64]	1.346
DASC [64]	**0.003**
Proposed	**0.003**

4.6.10 Computational Efficiency Analysis

The computational efficiency of the proposed control framework was evaluated across multiple prediction horizon lengths and benchmarked against a standard MPC approach. Table 4.4 provides detailed statistics of the computation time required to determine the optimal action per step in one iteration.

Across all horizon settings, the proposed method achieves mean computation times that are comparable to or slightly lower than those of standard MPC. Notably, the standard deviation of computation time is reduced by approximately 80% compared to standard MPC, indicating a significantly more stable and predictable computational burden. This improvement is attributed to the convex mapping constraint, which effectively narrows and refines the feasible solution space, ensuring more stable and consistent solving time costs even at the longest horizon of 40 steps.

Table 4.5 compares the computation time standard deviations of the proposed method with two benchmark strategies: the stochastic MPC (SMPC) and the learning-driven DAgger-Based Stochastic Control (DASC) method [64]. The proposed method achieves a computation-time standard deviation of approximately 0.003 with a prediction horizon of 10, which is comparable to DASC but significantly lower than SMPC. These results

confirm the computational stability of the proposed approach and its suitability for real-time battery charging applications.

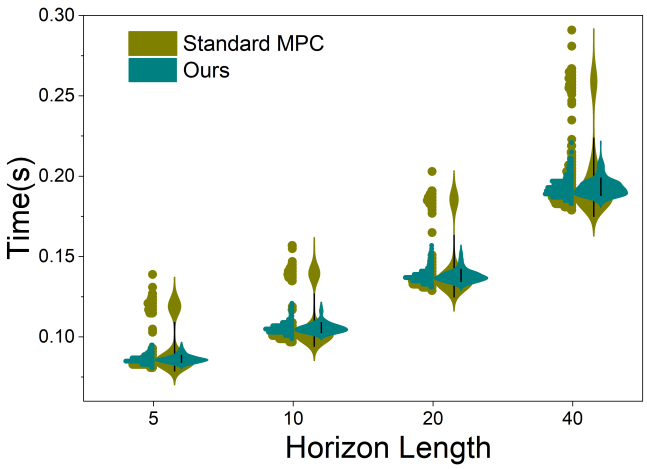


Figure 4.11: Violin plot of computation time distributions across different prediction horizons.

Figure 4.11 visualises the full distribution of computation times using violin plots. The proposed method exhibits tighter and more concentrated distributions across all horizon lengths, confirming the stable computational behaviour. In contrast, standard MPC exhibits heavier upper tails, particularly at longer horizons, indicating a greater chance of occasional prolonged computations. The consistent compact shape of the computation time distributions of the proposed strategy demonstrates improved computation determinism, an essential property for reliable real-time implementation.

4.6.11 Continuous Optimisation Performance

Table 4.6 summarises the evolution of the mean and standard deviation of the charging current across 10 iterations under different internal temperature limitations. To further verify the robustness of the iterative refinement process, results obtained using both CC-CV and standard MPC charging profiles as initial conditions were compared under a stringent temperature limit of 34 °C.

Table 4.6: Mean and standard deviation of charging current under different internal temperature limits across 10 iterations.

Iter.	Mean of Charging Currents (A)						Std of Charging Currents (A)					
	40°	39°	38°	37°	$34^{(a)}$	$34^{(b)}$	40°	39°	38°	37°	$34^{(a)}$	$34^{(b)}$
1	−18.32	−18.10	−16.53	−15.15	−18.19	−13.40	8.35	7.22	7.62	11.12	7.54	**5.19**
2	−19.69	−19.00	−18.00	−15.80	−15.58	−13.24	6.06	5.44	5.12	9.77	6.38	**5.35**
3	−20.19	−19.21	−18.05	−16.68	−13.89	−13.55	4.73	4.84	4.99	5.82	5.43	**4.61**
4	−20.19	−19.22	−18.06	−17.03	−13.30	−13.55	4.69	4.83	4.94	4.95	4.88	**4.61**
5	−20.20	−19.22	−18.07	−17.05	−13.36	−13.54	4.66	4.83	4.92	4.88	4.76	**4.61**
6	−20.21	−19.22	−18.07	−17.06	−13.35	−13.35	4.65	4.83	4.90	4.86	4.75	**4.61**
7	−20.22	−19.22	−18.07	−17.07	−13.35	−13.53	4.63	4.84	4.88	4.85	4.75	**4.61**
8	−20.22	−19.22	−18.07	−17.07	−13.35	−13.52	4.62	4.84	4.87	4.84	4.73	**4.61**
9	−20.23	−19.22	−18.07	−17.09	−13.35	−13.51	4.61	4.84	4.85	4.84	4.73	**4.61**
10	−20.22	−19.22	−18.06	−17.09	−13.35	−13.50	4.60	4.84	4.84	4.84	4.71	**4.61**

$34°C^{(a)}$: result with CC-CV charging profile as initial solution.
$34°C^{(b)}$: result with standard MPC charging profile as initial solution.

The mean charging current exhibits stable convergence trends across all temperature thresholds. The values gradually increase during the initial iterations to reduce the overall charging duration and then stabilise once convergence is achieved. The reduction in mean current when decreasing the temperature limit further demonstrates the controller's ability to adaptively regulate the charging rate in response to stricter thermal constraints. Comparisons between the two initialisation strategies at 34 °C confirm that the proposed method effectively mitigates the influence of the initial solution; regardless of whether the process starts from a CC-CV profile or a standard MPC profile, the mean charging current converges toward nearly identical steady-state values.

The evolution of the standard deviation confirms that the controller progressively suppresses fluctuations in the charging current. Notably, in the $34°C^{(b)}$ case, the initial current exhibits noticeable fluctuation due to the stringent thermal constraint; however, after several iterations, the variability is reduced to a comparable level (approximately 4.61–4.62 A). These results collectively confirm the robustness and repeatability of the continuous optimisation mechanism irrespective of the starting solution.

Figure 4.12 illustrates the evolution of the memory function across iterations. The magnitude of the memory function decreases consistently with the iterations, reflect-

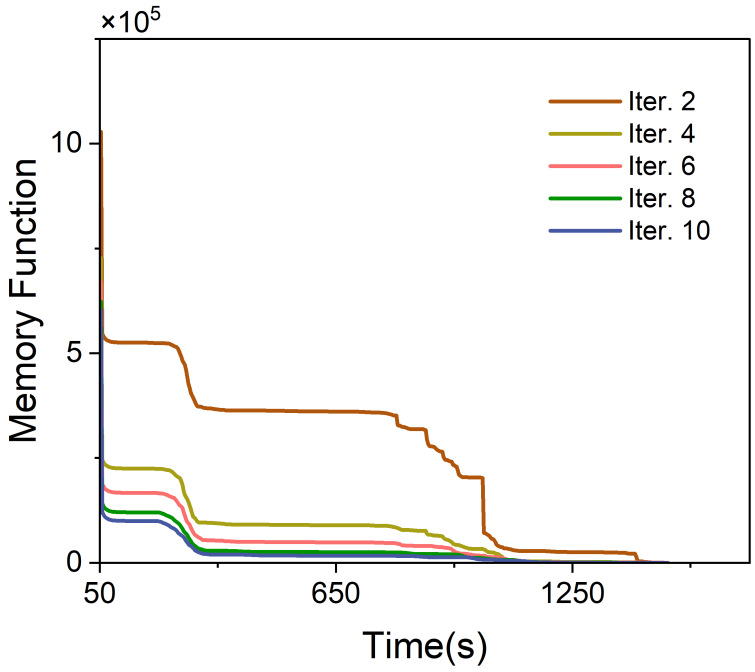


Figure 4.12: Memory function evolution across iterations. ($N_p = 20$, $T_{in}^{\max} = 40\,^{\circ}\mathrm{C}$)

ing the progressive refinement of the internal representation of historical information. This reduction indicates that the proposed historical data-driven self-learning control framework continually improves the control performance and corrective capability, ultimately driving the control process toward stable convergence and effective achievement of charging objectives.

4.6.12 Charging Duration Comparison

Table 4.7 compares the charging durations achieved by the proposed control strategy with those of an existing constrained generalised predictive control (GPC) method [135] under different internal temperature thresholds. Across all comparable temperature limits, the proposed strategy consistently achieves shorter charging durations while satisfying the same thermal constraints. For instance, at 37 °C, the charging duration is reduced by 100 s compared to GPC, with similar improvements observed at other temperature thresholds. These results highlight the enhanced charging efficiency delivered by the

proposed method and confirm its adaptability in optimising the charging duration across a range of thermal limitations.

Table 4.7: Charging duration comparison against the reference GPC strategy.

	40°C	39°C	38°C	37°C	34°C
GPC [135] (s)	1498	1579	1674	1786	—
Proposed (s)	1428	1500	1590	1686	2133
Improvement (s)	70	79	84	100	—

4.7 Chapter Summary

This chapter has developed a historical data-driven self-learning MPC framework for repetitive battery charging under explicit thermal constraints [27]. The work addresses the problem of incorporating historical charging trajectories into the control formulation while maintaining computational tractability and constraint satisfaction.

The main conclusions of this chapter are as follows:

1. A convex mapping constraint was formulated from historical charging trajectories. The constraint narrows the feasible solution space and provides a systematic mechanism for incorporating past experience into the current optimisation. In the reported experiments, it reduces MPC solve-time variability by approximately 80% compared to standard MPC.

2. A memory function-enhanced cost function was introduced to incorporate historical state–input information into the current quadratic MPC objective. The resulting formulation enables iterative refinement of the charging trajectory while preserving optimisation tractability; the reported results demonstrate progressive improvement across charging iterations rather than a general guarantee of global optimality.

3. A cascading linearised thermoelectric battery model was developed in the case study to capture the coupled electrical and thermal dynamics of the battery, includ-

ing both internal and surface temperature dynamics. By reformulating the thermal dynamics as linear with respect to $u(t) = i(t)^2$, the model enables the application of convex optimisation within the general MPC framework while preserving the nonlinear battery characteristics. The model was validated against experimental data with a voltage RMSE of approximately 3.4 mV and an internal temperature RMSE of approximately 0.82 °C.

4. Experimental studies demonstrated that the proposed control strategy maintains the internal battery temperature within the tested safety limits and reduces charging duration by 70–100 s compared to an existing GPC strategy across multiple temperature thresholds. It also maintains satisfactory performance under the battery-parameter uncertainty levels evaluated, up to 10%.

5. The convex mapping constraint improves computational reliability in the reported experiments by narrowing the feasible solution space. The computation-time standard deviation is reduced by approximately 80% compared to standard MPC. The comparison with DASC is retained as an experimental benchmark only [64].

The proposed formulation is expressed using general linear-system notation, although it is validated here for repetitive battery charging. Since the convex mapping constraint and memory function are derived from historical state and input trajectories, the approach may be transferable to other repetitive constrained processes when comparable data and a suitable predictive model are available.

However, this generality is predicated on a separability assumption: a single, pre-identified linearised model can represent the controlled system, so modelling and control can be treated as sequential, decoupled stages. This assumption fails once the system comprises multiple interacting subsystems with heterogeneous parameters and markedly different timescales. The multi-cell battery pack, with its cell-to-cell heterogeneity and disparate electrical and thermal time constants, exemplifies this challenge. This motivates the following chapter, which develops a unified modelling-and-control framework, pairing a parallel-inference neural model for heterogeneous cells with a 2D cost map-

guided predictive controller, to jointly achieve SOC equalisation, thermal regulation, and voltage stabilisation across the pack.

CHAPTER 5

2D COST MAP-GUIDED PARALLEL-INFERENCE NEURAL MODELLING AND PREDICTIVE CONTROL FRAMEWORK

5.1 Introduction

For the converter-based modular battery pack studied in this chapter, the balancing task is characterised by strongly coupled electrical and thermal dynamics, heterogeneous cell behaviour, and state variables that respond on markedly different timescales. A model-based solution therefore requires not only a description of the coupled dynamics, but also a control architecture that can quantify the relative urgency of fast electrical states and slow thermal states under candidate balancing actions. The pack-level problem is further complicated by cell-to-cell heterogeneity and by the fact that some internal states are not directly measurable [136]. Data-driven neural models can reduce the need for explicit physical modelling, but a one-dimensional representation may not make the cross-cell and temporal dependencies explicit [137]. In addition, integrating a neural predictor into real-time predictive control requires an efficient way to evaluate candidate actions without repeatedly computing network gradients.

To address this specific pack-balancing problem, this chapter proposes a 2D Cost

Map-Guided Parallel-Inference Neural modelling and Predictive Control framework. The framework is tailored to the coupled electrical and thermal dynamics of the battery pack and rests on two principal innovations:

1. A unified multi-task state-action neural model that encodes the historical state sequence of the battery cells and current control inputs in a two-dimensional (2D) image-like matrix, enabling simultaneous prediction of the coupled states considered under different candidate balancing actions. The architecture employs decomposed prediction subnetworks for the fast electrical and slow thermal dynamics, together with a recursive closed-loop mechanism for multi-step prediction.

2. An inertia-aware 2D cost-map guided multi-objective control framework that represents the different response rates of the electrical and thermal states across the prediction horizon. State-specific exponential coefficients provide a physics-motivated weighting of prediction-horizon penalties. A Gaussian-based parallel sampling strategy generates candidate balancing trajectories, which are evaluated by the neural model and combined through a softmax-weighted mechanism without gradient computation through the network.

The framework is applied to the multi-objective active balancing of converter-based modular battery packs as a representative and practically significant case study. In this application, the fast-dynamics states correspond to electrochemical quantities (state of charge and output voltage), while the slow-dynamics states capture thermal behaviour (cell temperatures). Simultaneous balancing of these coupled states under time-varying load profiles constitutes a challenging multi-objective control problem that exercises all aspects of the proposed framework.

The specific contributions of this chapter are as follows:

1. A control-oriented, multi-task deep neural model is developed for the coupled battery-pack dynamics considered here. The 2D state-action input formulation separates historical system dynamics from current control actions and distinguishes

fast electrical states, slow thermal states, and system-level outputs. This formulation supports closed-loop multi-step forecasting under the evaluated operating conditions without requiring an explicit pack-level state-space model.

2. A decomposed multi-rate prediction architecture is proposed for the electrical and thermal state classes in the battery pack: MLP subnetworks capture fast single-step transitions, while TimesNet blocks extract multi-scale temporal dependencies in the slower thermal states. An ablation study confirms the complementary contributions of both components.

3. An inertia-aware 2D cost map is introduced as a physics-motivated multi-objective evaluation metric. State-specific exponential coefficients are assigned according to the evaluated temporal response characteristics, with decaying weights for fast states and amplifying weights for slow states. The resulting 2D visualisation provides an interpretable and adjustable basis for action selection.

4. A parallel sampling-based neural MPC (NMPC) framework is proposed that exploits batch processing to evaluate candidate action trajectories without gradient computation. Applied to the battery-pack balancing case study under three standard automotive driving cycles, the framework achieves simultaneous regulation of temperature difference below 5°C, SOC non-uniformity below 1%, and output voltage within a ±10% error band.

The remainder of this chapter is organised as follows. Section 5.2 presents the general formulation of multi-state coupled control problems. Section 5.3 details the unified multi-task state-action neural model. Section 5.4 introduces the inertia-aware 2D cost-map guided control framework. Section 5.5 presents the comprehensive battery pack balancing case study. Section 5.6 summarises the chapter.

5.2 General Formulation of Multi-State Coupled Control Problems

5.2.1 Multi-Subsystem Architecture and State Representation

Consider a system comprising N interacting subsystems, each equipped with a dedicated actuator that can be independently commanded. The state variables of each subsystem $i \in \{1, \ldots, N\}$ are characterised by their temporal response to control actions and can be partitioned into two categories based on their physical time constants:

- **Fast-dynamics states** $\mathbf{x}_i^{\mathrm{f}} \in \mathbb{R}^{n_f}$: quantities that respond rapidly to changes in the control input, exhibiting short response time constants and relatively weak long-range temporal dependencies. These states are, in principle, correctable within a short prediction horizon.
- **Slow-dynamics states** $\mathbf{x}_i^{\mathrm{s}} \in \mathbb{R}^{n_s}$: quantities that evolve cumulatively through dissipative, diffusive, or integrative processes, exhibiting long time constants and significant inertia. Deviations in these states accumulate over time and require anticipatory control actions over an extended horizon.

In addition to per-subsystem states, the system may exhibit one or more system-level outputs $\mathbf{y} \in \mathbb{R}^{n_y}$ that aggregate contributions from all subsystems and are subject to regulation constraints. The control input to subsystem i is denoted $u_i \in \mathbb{R}$, and the collective control vector is $\mathbf{u} = [u_1, \ldots, u_N]^\top \in \mathbb{R}^N$.

The system-wide aggregations of fast and slow states across all subsystems are:

$$\mathbf{X}^{\mathrm{f}} = \left[\mathbf{x}_1^{\mathrm{f}}, \ldots, \mathbf{x}_N^{\mathrm{f}}\right] \in \mathbb{R}^{N \times n_f}, \tag{5.1}$$

$$\mathbf{X}^{\mathrm{s}} = \left[\mathbf{x}_1^{\mathrm{s}}, \ldots, \mathbf{x}_N^{\mathrm{s}}\right] \in \mathbb{R}^{N \times n_s}. \tag{5.2}$$

The degree of non-uniformity across subsystems for the l-th fast-dynamics state variable and the m-th slow-dynamics state variable is quantified by the respective

standard deviations:

$$SD\left(X_l^{\mathrm{f}}\right) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(x_{i,l}^{\mathrm{f}} - \bar{x}_l^{\mathrm{f}}\right)^2}, \tag{5.3}$$

$$SD\left(X_m^{\mathrm{s}}\right) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(x_{i,m}^{\mathrm{s}} - \bar{x}_m^{\mathrm{s}}\right)^2}, \tag{5.4}$$

where $\bar{x}_l^{\mathrm{f}}$ and $\bar{x}_m^{\mathrm{s}}$ denote the respective mean values across all N subsystems.

5.2.2 Multi-Objective Control Problem and Key Challenges

The multi-objective control problem addressed in this chapter is to simultaneously achieve the following objectives by selecting the collective control input $\mathbf{u}$:

1. **Fast-state uniformity**: minimise $SD(X_l^{\mathrm{f}})$ for $l = 1,\ldots,n_f$, equalising the fast-dynamics states across all subsystems to prevent localised overloading or under-utilisation;

2. **Slow-state uniformity**: limit $SD(X_m^{\mathrm{s}})$ for $m = 1,\ldots,n_s$, managing the spatially heterogeneous evolution of slowly varying states to avoid accumulated divergence;

3. **System-level output regulation**: maintain the aggregate system output $\mathbf{y}$ within a desired operating range or close to a setpoint $\mathbf{y}_d$.

Addressing this problem using conventional model-based approaches faces three fundamental challenges.

Challenge 1: Coupled nonlinear multi-state dynamics. The fast and slow states are not independent: the evolution of slow-dynamics states is driven in part by the currents, heat fluxes, or other quantities associated with fast-dynamics transitions. This coupling makes it infeasible to predict or control each state class in isolation. Constructing an accurate physics-based model of the coupled dynamics typically requires extensive per-subsystem parameter identification and is further complicated by parameter drift under ageing or varying operating conditions.

Challenge 2: State inertia heterogeneity. Fast and slow states exhibit fundamentally different physical time constants. Applying a uniform cost penalty across all state types

and all prediction steps is therefore physically inconsistent: a slow-state deviation P steps ahead carries greater practical significance than a fast-state deviation at the same horizon, because the former has accumulated cumulatively and cannot be reversed quickly, whereas the latter can be corrected by near-future control actions. A principled treatment of the prediction horizon must reflect these differences.

Challenge 3: Gradient-based optimisation infeasibility. Multi-objective optimisation over a deep neural model is non-convex, and computing analytical gradients through the network at each real-time control step is computationally prohibitive for embedded deployment. An alternative optimisation strategy that avoids gradient computation while exploiting the parallelism available on modern hardware is needed.

The following two sections address these three challenges through the proposed unified neural modelling (Section 5.3) and parallel sampling-based control framework (Section 5.4).

5.3 Unified Multi-Task State-Action Neural Modelling

5.3.1 Architecture Overview

Conventional neural prediction models for multi-component systems typically predict a single state variable from a one-dimensional input representation, which fails to capture spatial correlations across subsystems, temporal evolution of coupled states of multiple types, or the influence of control actions on future state trajectories. Integrating such models into a predictive control loop is therefore problematic: without explicit conditioning on candidate control actions, the model cannot distinguish the system's response to different control decisions.

To address these limitations, a *control-oriented, multi-task deep neural model* is developed. Model inputs are divided into a 2D historical state matrix and a current action vector, enabling simultaneous prediction of multiple state variables across all subsystems in response to different control actions. The overall architecture is illustrated in Figure 5.1.

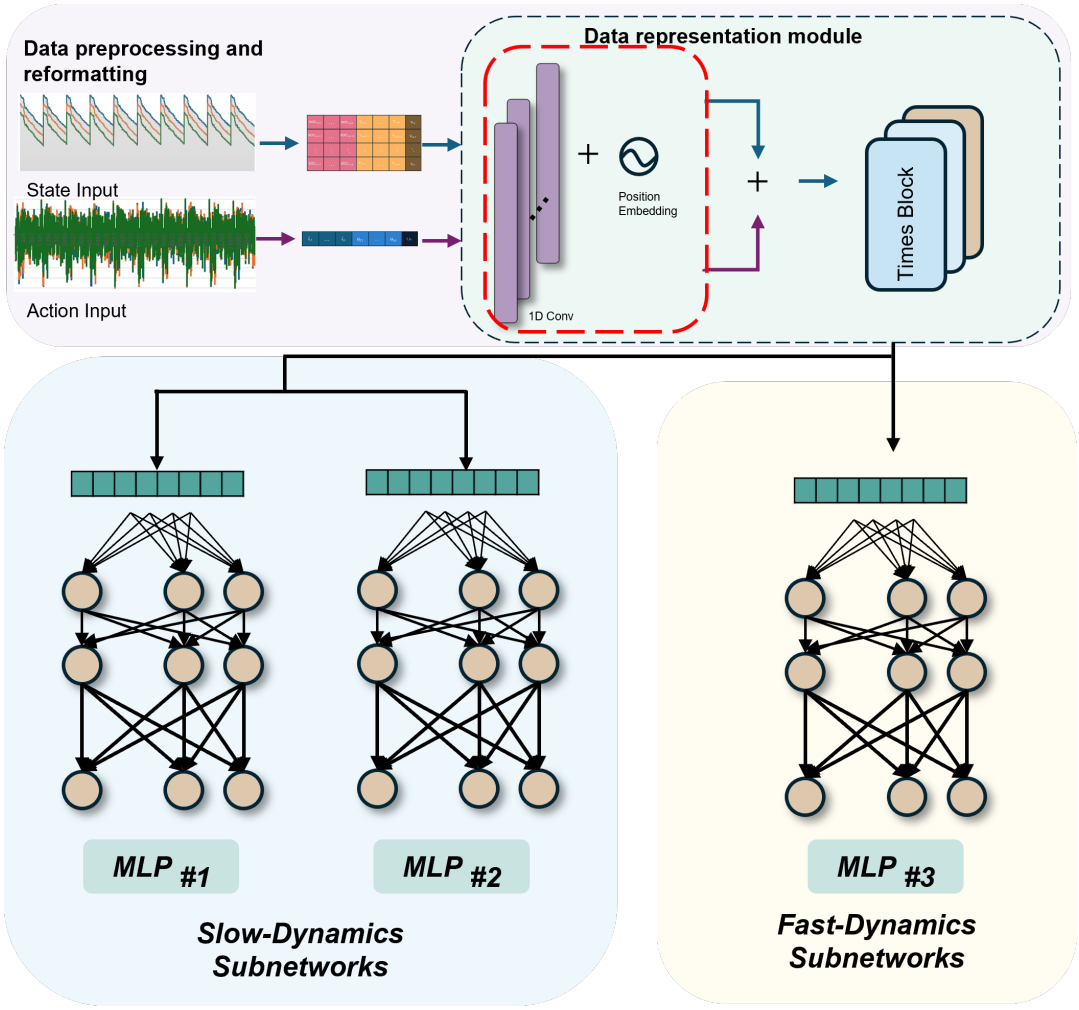


Figure 5.1: Overall architecture of the proposed control-oriented multi-task neural model for multi-state coupled systems.

The proposed model comprises four main components: (i) a 2D input formulation that encodes historical multi-subsystem state sequences into an image-like matrix, capturing spatial-temporal dependencies across subsystems and state types; (ii) a time-series embedding module that projects heterogeneous state and action inputs into a unified higher-dimensional representation space; (iii) decomposed fast-dynamics and slow-dynamics prediction subnetworks whose architectural complexity is matched to the physical time constants of each state class; and (iv) a recursive closed-loop deployment mechanism for multi-step prediction during control execution.

5.3.2 2D State-Action Input Formulation

The multi-subsystem system exhibits coupled dynamics across state variables of different types, evolving over different timescales. Representing these states as a simple one-dimensional vector is insufficient to capture their joint temporal and cross-variable correlations. Instead, state data are formulated into a 2D image-like structure integrating historical time-series information and multi-variable dependencies, as illustrated in Figure 5.2.

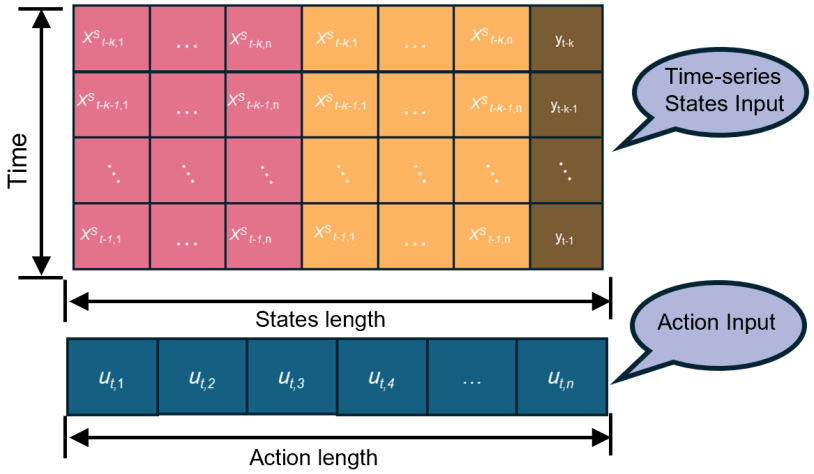


Figure 5.2: Structure of the 2D state input matrix and associated action input vector.

A sliding window of length K is employed. For each time step t, the past K system states are arranged vertically along the temporal dimension, with the concatenated state variables from all N subsystems forming the feature dimension. The resulting 2D state input matrix is:

$$\mathbf{X}_t \in \mathbb{R}^{K \times d_s}, \tag{5.5}$$

where $d_s = N(n_f + n_s) + n_y$ is the total feature dimension per time step, and row k of $\mathbf{X}_t$ contains:

$$\left[\mathbf{x}^{\mathrm{f}}_{t-k,1}, \ldots, \mathbf{x}^{\mathrm{f}}_{t-k,N}, \mathbf{x}^{\mathrm{s}}_{t-k,1}, \ldots, \mathbf{x}^{\mathrm{s}}_{t-k,N}, \mathbf{y}_{t-k}\right]. \tag{5.6}$$

This image-like formulation enables the network to extract richer spatial-temporal features, simultaneously capturing correlations across subsystems, across state variable types, and across time.

The action input at time step t encodes the current control command together with any exogenous inputs required to characterise the operating condition:

$$\mathbf{u}_t \in \mathbb{R}^{d_u}, \tag{5.7}$$

where d_u includes the primary actuator signals $[u_{t,1}, \ldots, u_{t,N}]^\top$ as well as any derived or exogenous variables (e.g., load current, reference setpoints) needed to fully specify the system's input at time t.

Separating the input into a 2D state matrix and an action vector offers two key advantages: it enables the model to distinguish intrinsic system dynamics from externally imposed control actions, improving feature extraction and generalisation; and it establishes a control-oriented prediction structure under which the model can accurately forecast system responses to different candidate control actions, as required by the NMPC framework in Section 5.4.

5.3.3 Time-Series Embedding and Unified Representation

To address the dimension mismatch between the historical state sequence and the single-step action input, a dedicated time-series embedding module is introduced, motivated by the positional encoding principles of large language models [138]. Both inputs are projected through a 1D convolutional layer to a unified higher-dimensional embedding space. For each time step k, the value embedding of an input vector $x_k^a \in \mathbb{R}^a$ into a d-dimensional latent representation $h_k^d \in \mathbb{R}^d$ is:

$$h_k^d = \mathrm{conv}_{1\mathrm{D}}(x_k^a). \tag{5.8}$$

To preserve temporal order information, a sinusoidal positional embedding is applied. For dimension index $m = 0, \ldots, \lfloor d/2 \rfloor$, the positional embedding $p_k^d \in \mathbb{R}^d$ is:

$$p_k^{2m} = \sin\!\left(k / 10000^{2m/d}\right), \qquad p_k^{2m+1} = \cos\!\left(k / 10000^{(2m+1)/d}\right). \tag{5.9}$$

The convolutional feature and positional embedding are fused by element-wise addition:

$$r_k^d = h_k^d \oplus p_k^d. \tag{5.10}$$

The embeddings for the state matrix and action vector are denoted $r_k^{d,s}$ and $r^{d,u}$, respectively. Since the action input spans only one time step while the state matrix spans K steps, a temporal padding module aligns the action embedding to the same temporal dimension. The final time-aligned, unified latent representation is:

$$\mathbf{R}_k^d = r_k^{d,s} \oplus \text{Padding}\big(r^{d,u}\big), \tag{5.11}$$

which provides a complete and dimensionally consistent representation for the downstream prediction subnetworks.

5.3.4 Decomposed Multi-Rate Prediction Subnetworks

To accurately capture the coupled dynamics of a multi-state system, the prediction task is decomposed into two dedicated subnetworks whose architectural complexity is matched to the physical timescale of each state class.

5.3.4.1 Fast-Dynamics Prediction Subnetwork

Fast-dynamics states—and system-level outputs that respond promptly to control inputs—exhibit short temporal dependencies primarily driven by the instantaneous control command, with limited dependence on remote history. A multilayer perceptron (MLP) is therefore sufficient to approximate the one-step-ahead transition of these quantities. Using the unified embedding $\mathbf{R}_k^d$, the fast-dynamics predictions are:

$$\hat{\mathbf{x}}_{k+1}^{\text{f}} = \text{MLP}_{\text{f}}\big(\mathbf{R}_k^d\big) \in \mathbb{R}^{Nn_f+n_y}, \tag{5.12}$$

where the output concatenates the predicted fast-dynamics state variables for all N subsystems and the system-level output(s). The MLP's feedforward structure is well-suited to capturing the instantaneous, algebraically-dominated transitions characteristic of fast-responding quantities.

5.3.4.2 Slow-Dynamics Prediction Subnetwork

Slow-dynamics states evolve through dissipative, diffusive, or integrative processes involving multi-scale temporal patterns that require richer temporal modelling. A Times-

Net block [139] is adopted, which leverages the Fast Fourier Transform (FFT) to extract multi-period temporal structures from the unified embedding:

$$\mathbf{H}^{\mathrm{s}} = \mathrm{TimesBlock}\big(\mathbf{R}_k^d\big), \tag{5.13}$$

where $\mathbf{H}^{\mathrm{s}} \in \mathbb{R}^d$ encodes both short-term and long-term temporal dependencies relevant to slowly evolving states. The refined representation is passed to a task-specific MLP decoder:

$$\hat{\mathbf{x}}_{k+1}^{\mathrm{s}} = \mathrm{MLP}_{\mathrm{s}}\big(\mathbf{H}^{\mathrm{s}}\big) \in \mathbb{R}^{Nn_s}. \tag{5.14}$$

5.3.4.3 Integrated Prediction and Recursive Closed-Loop Deployment

The outputs of the two subnetworks are concatenated to form the full predicted state vector:

$$\hat{\mathbf{x}}_{k+1} = \begin{bmatrix} \hat{\mathbf{x}}_{k+1}^{\mathrm{f}} \\ \hat{\mathbf{x}}_{k+1}^{\mathrm{s}} \end{bmatrix}, \tag{5.15}$$

containing the predictions of all fast-dynamics states, system-level outputs, and slow-dynamics states across all subsystems.

During *training*, the state input is constructed from historical state sequences and the action input corresponds to the recorded control command; the model is trained end-to-end with a weighted multi-task regression loss. During *control deployment*, the predicted state $\hat{\mathbf{x}}_{k+1}$ is recursively fed back to update the sliding state window, enabling closed-loop multi-step forecasting under any candidate action trajectory, as required by the NMPC framework described in Section 5.4.

The decomposed architecture reflects a principled alignment between model complexity and the physical time constants of each state variable class. The validity of this design is confirmed by the ablation study presented in Section 5.5.4.

5.4 Inertia-Aware 2D Cost-Map Guided Parallel-Inference Control Framework

5.4.1 Parallel Sampling-Based Action Generation

Although deep neural models provide powerful nonlinear approximation capabilities, incorporating them into gradient-based optimisation frameworks remains challenging due to the intrinsic non-convexity of neural networks and the absence of tractable analytical gradients for control-oriented objectives. A *sampling-based neural MPC framework* is therefore proposed, enabling efficient parallel evaluation of candidate actions and seamless embedding of the neural predictor into the control loop. The overall control architecture is illustrated in Figure 5.3.

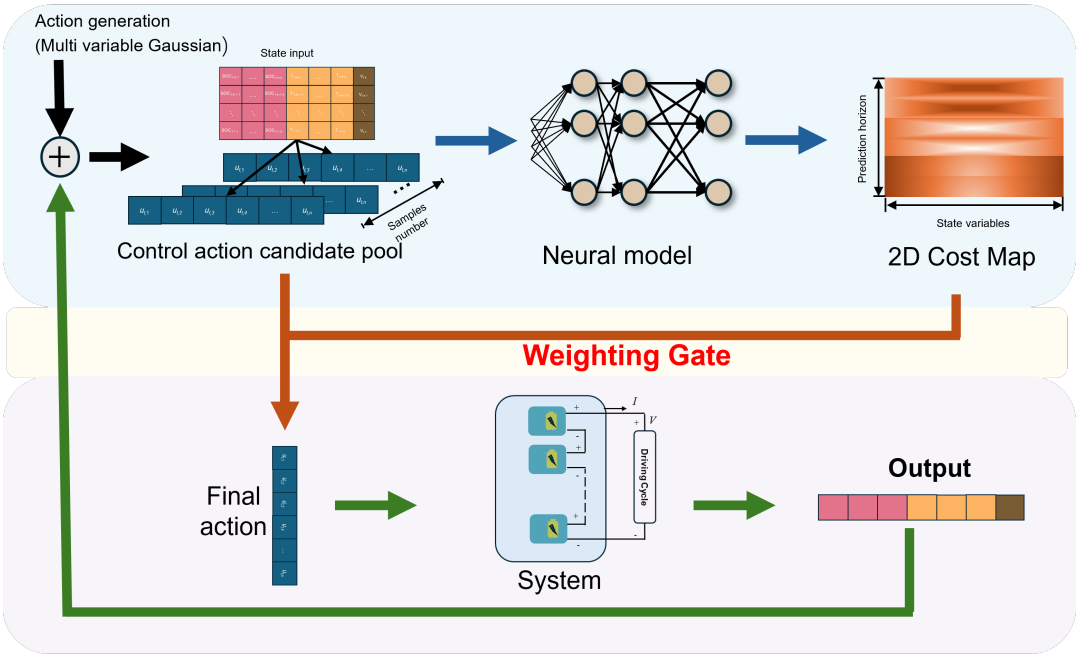


Figure 5.3: Proposed parallel sampling-based neural MPC framework for multi-state coupled systems.

At each control step t, the previous control input vector $\mathbf{u}_{t-1} \in \mathbb{R}^N$ is used as the centre of a multivariate Gaussian distribution from which a batch of B candidate action

trajectories is sampled:

$$\tilde{\mathbf{u}}_t \sim \mathcal{N}\big(\mathbf{u}_{t-1}, \boldsymbol{\Sigma}\big), \tag{5.16}$$

where $\boldsymbol{\Sigma}$ controls the exploration spread. Gaussian sampling centred on the previous action is physically consistent with the continuous, smooth evolution of most engineering state variables, which do not exhibit abrupt jumps between consecutive control steps.

Physical constraints are enforced by passing each sample through a bounding operator applied element-wise:

$$\mathscr{B}(\tilde{\mathbf{u}}_t) = \max\big(\mathbf{u}_{\min}, \min(\tilde{\mathbf{u}}_t, \mathbf{u}_{\max})\big), \tag{5.17}$$

ensuring all control signals remain within their physically admissible ranges $[\mathbf{u}_{\min}, \mathbf{u}_{\max}]$. The resulting bounded candidate pool is:

$$U = [U_1, U_2, \ldots, U_B]^\top \in \mathbb{R}^{B\times P\times N}, \tag{5.18}$$

where each trajectory $U_j \in \mathbb{R}^{P\times N}$ specifies the control actions over the full prediction horizon P.

5.4.2 Inertia-Aware 2D Cost Map

A central challenge in multi-objective control of systems with heterogeneous state dynamics is that a physically consistent cost evaluation must account for the different inertia characteristics of fast and slow states. A fast-state deviation P steps ahead is generally less critical than a slow-state deviation at the same horizon, because the former can be corrected by subsequent near-future actions while the latter has accumulated through a slow cumulative process and is difficult to reverse quickly. A uniform cost penalty across all states and all prediction steps is therefore physically inconsistent.

To address this, an *inertia-aware cost function* is formulated based on a two-dimensional cost map that jointly incorporates (i) the prediction horizon index and (ii) the multi-state non-uniformity and regulation errors. For each subsystem $i = 1,\ldots,N$ and each prediction step $j = 1,\ldots,P$, the instantaneous non-uniformity errors for the fast and slow state variables are:

$$e_l^{\mathrm{f}}(t+j, i) = \hat{x}^{\mathrm{f}}_{t+j,i,l} - \frac{1}{N}\sum_{i'=1}^{N} \hat{x}^{\mathrm{f}}_{t+j,i',l}, \quad l = 1,\ldots,n_f, \tag{5.19}$$

$$e_m^{\mathrm{s}}(t+j,\,i) = \hat{x}_{t+j,i,m}^{\mathrm{s}} - \frac{1}{N}\sum_{i'=1}^{N} \hat{x}_{t+j,i',m}^{\mathrm{s}}, \quad m = 1,\ldots,n_s, \tag{5.20}$$

and the system-level output regulation error is:

$$\mathbf{e}_y(t+j) = \hat{\mathbf{y}}_{t+j} - \mathbf{y}_d. \tag{5.21}$$

Inertia adjustment coefficients α^{f}, α^{s}, and α_y scale each state class's error contribution along the prediction horizon. Setting $\alpha^{\mathrm{f}} \le 1$ emphasises short-horizon errors for fast states, which is appropriate since these quantities can be corrected promptly. Setting $\alpha^{\mathrm{s}} \ge 1$ amplifies long-horizon errors for slow states, reflecting their cumulative nature and the practical importance of anticipating their evolution well in advance. The resulting inertia-aware multi-objective cost function is:

$$\begin{aligned} J(t) = \frac{1}{P}\sum_{j=1}^{P}\Bigg[&\sum_{l=1}^{n_f} \|\mathbf{e}_l^{\mathrm{f}}(t+j)\|_{R_f}^2 \left(\alpha^{\mathrm{f}}\right)^j + \sum_{m=1}^{n_s} \|\mathbf{e}_m^{\mathrm{s}}(t+j)\|_{R_s}^2 \left(\alpha^{\mathrm{s}}\right)^j \\ &+ \|\mathbf{e}_y(t+j)\|_{R_y}^2 \alpha_y^j + \Delta\mathbf{u}_{t+j}^{\top} H\, \Delta\mathbf{u}_{t+j} \Bigg], \end{aligned} \tag{5.22}$$

where $R_f, R_s \succ 0$ and $R_y > 0$ are positive weighting matrices/scalars; $\Delta\mathbf{u}_{t+j} = \mathbf{u}_{t+j} - \mathbf{u}_{t+j-1}$ is the control increment; and $H \succeq 0$ penalises excessive control variation.

The 2D cost map arising from equation (5.22) is a matrix of dimension $P \times (Nn_f + Nn_s + n_y)$, with columns corresponding to per-subsystem fast-state errors, slow-state errors, and system-level output errors, and rows corresponding to prediction steps. The exponential inertia coefficients create distinct colour gradients along the horizon dimension for each state class—decaying for fast states and amplifying for slow states—providing an interpretable and physically grounded basis for action selection and performance analysis.

5.4.3 Weighted Action Selection Mechanism

The B candidate action trajectories in U are evaluated by the multi-task neural model in a single parallel batch-inference pass, producing the corresponding multi-step state predictions and cost values $J(U_j)$ for each $j = 1,\ldots,B$. A softmax-weighted aggregation mechanism is then applied to synthesise the final control decision without any gradient computation.

The weighting gate $\mathbf{w} \in \mathbb{R}^B$ is computed using a normalised softmax formulation that emphasises relative cost differences rather than absolute magnitudes:

$$w(j) = \frac{\exp\big(-\lambda\,\xi(j)\big)}{\sum_{j=1}^{B} \exp\big(-\lambda\,\xi(j)\big)}, \qquad j = 1,\ldots,B, \tag{5.23}$$

where $\lambda > 0$ controls the sharpness of the weighting distribution. The normalised cost $\xi(j)$ removes the influence of the absolute cost magnitude:

$$\xi(j) = \sum_{i=1}^{P} J\big(U_j(i)\big) - \min_{k\in\{1,\ldots,B\}} \sum_{i=1}^{P} J\big(U_k(i)\big). \tag{5.24}$$

The optimal action sequence is the weighted average of all candidates:

$$\mathbf{u}^* = \sum_{j=1}^{B} w(j)\,U_j. \tag{5.25}$$

Following the receding-horizon principle, only the first action vector is extracted and applied to the system:

$$u_t^* = [1, 0, \ldots, 0]\,\mathbf{u}^*, \qquad u_t^* \in \mathbb{R}^N. \tag{5.26}$$

This weighted-gate mechanism produces smooth and stable control outputs, and effectively mitigates the instability typically associated with pure best-candidate selection in sampling-based methods, particularly when the batch size is varied.

5.4.4 Algorithm Summary

Algorithm 2 summarises the complete parallel-inference neural MPC procedure. A sliding-window mechanism maintains a fixed-length state input sequence; after each control step, the window advances by one as new state measurements are received.

The framework avoids gradient computation through the neural model entirely by using sampling-based optimisation. The parallel evaluation of B candidates in a single neural model forward pass fully exploits GPU batch-processing capabilities, enabling real-time deployment even for large candidate pools. The inertia-aware 2D cost map provides interpretable and physically consistent differentiation of state contributions across the horizon, making control decisions transparent and adjustable through the intuitive inertia coefficients α^{f}, α^{s}, and α_y.

Algorithm 2 Parallel Sampling-Based Neural MPC for Multi-State Coupled Systems

Require: K: state window length; N: number of subsystems; P: prediction horizon; B: batch size;
(x_0, u_0): initial state window and action;
R_f, R_s, R_y, H: cost weighting parameters;
$\alpha^{\mathrm{f}}, \alpha^{\mathrm{s}}, \alpha_y$: inertia coefficients; λ: softmax sharpness.
while control task not completed **do**
  Generate candidate pool $U \in \mathbb{R}^{B\times P\times N}$ via equations (5.16) and (5.17).
  for $j = 0$ **to** $P-1$ **do**
    $\hat{\mathbf{x}}_{t+j} \leftarrow \mathscr{F}(U, x_t)$ // *multi-task neural model (Section 5.3)*
  end for
  Compute cost $J(U_j)$ for all $j \in \{1,\dots,B\}$ via equation (5.22).
  $w(j) \leftarrow \exp(-\lambda\,\xi(j))/\sum_{k=1}^{B}\exp(-\lambda\,\xi(k))$ via equations (5.23)–(5.24).
  $\mathbf{u}^* \leftarrow \sum_{j=1}^{B} w(j)U_j$ via equation (5.25).
  $u_t^* \leftarrow [1,0,\dots,0]\mathbf{u}^*$; apply u_t^* to the system.
  $t \leftarrow t+1$; update sliding state window x_t and last action u_t.
end while

5.5 Case Study: Multi-Objective Balancing of Converter-Based Modular Battery Packs

This section applies the general framework developed in Sections 5.3 and 5.4 to the simultaneous SOC balancing, thermal balancing, and output voltage regulation of converter-based modular battery packs. This application is selected as a representative case study because it embodies all three key challenges identified in Section 5.2: strongly coupled electro-thermal dynamics, fundamentally different time constants between electrical and thermal states, and the need for real-time multi-objective control without gradient-based network optimisation.

5.5.1 Battery Pack Architecture and Application-Specific State Formulation

5.5.1.1 System Architecture

Battery packs assembled from numerous series-connected cells inevitably exhibit non-uniform electrical and thermal behaviours arising from manufacturing variability, self-

discharge rate differences, and uneven operating conditions [140]. This non-uniformity is further amplified in second-life battery applications. Heterogeneity in state of charge (SOC) and temperature across cells increases the risk of localised over-charging or over-discharging, accelerates capacity degradation [141], and in extreme cases can trigger thermal runaway. It is recommended that the maximum intra-pack temperature difference be maintained below 5 °C [142], while output voltage regulation is essential to satisfy load requirements in electric vehicle (EV) and grid-connected applications [143, 144].

To enable distributed cell-level balancing, this work adopts a converter-based modular battery architecture. Figure 5.4 illustrates the series-connected modular battery system based on bidirectional DC-DC converters. Each battery cell is paired with a dedicated bidirectional DC-DC converter that regulates the energy flow of the corresponding cell. During balancing, the high-side switch SH_i and low-side switch SL_i in each converter operate in a complementary manner. These cell-converter pairs are referred to as modular batteries (MB). The system delivers the output current I in response to the instantaneous power demand from the load, and the control variable u_i is defined as the duty cycle of the high-side switch SH_i. The pack output voltage is denoted V.

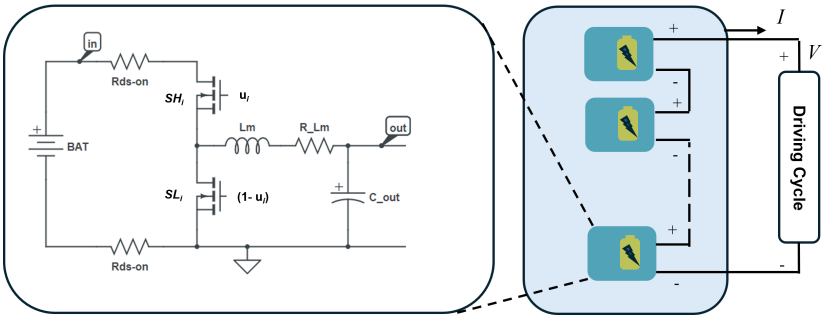


Figure 5.4: Converter-based modular battery structure.

The modular topology offers a distributed control structure particularly suited to multi-objective balancing. By adapting each module's duty cycle according to its electrical and thermal states, the system can flexibly regulate cell-level charge/discharge rates without disturbing the overall pack operation, enabling simultaneous SOC equalisation, temperature imbalance mitigation, and stable output voltage behaviour.

5.5.1.2 Application-Specific State Variables and Control Inputs

In the context of the general formulation (Section 5.2), the battery pack corresponds to a system of N subsystems (cells) with the following state classification:

- **Fast-dynamics states** ($n_f = 1$): the state of charge Q_i of cell i, which responds directly and rapidly to changes in the applied balancing current. The aggregate electrical state vector is $\mathbf{x}^{\mathrm{f}} = [Q_1, \ldots, Q_N]^\top \in \mathbb{R}^N$.
- **Slow-dynamics states** ($n_s = 1$): the internal temperature T_i of cell i, which evolves cumulatively through Joule heating, heat conduction, and ambient exchange, exhibiting significantly longer time constants than the electrical states. The aggregate thermal state vector is $\mathbf{x}^{\mathrm{s}} = [T_1, \ldots, T_N]^\top \in \mathbb{R}^N$.
- **System-level output** ($n_y = 1$): the pack output voltage V, which is an aggregate function of all cell voltages and is subject to regulation. The state representation thus corresponds to $\mathbf{x}^{\mathrm{f}}_{\mathrm{elec}} = [Q_1, \ldots, Q_N, V]^\top \in \mathbb{R}^{N+1}$ for the combined electrical quantities.

The 2D state input matrix $\mathbf{X}_t \in \mathbb{R}^{K \times (2N+1)}$ has rows:

$$\left[Q_{t-k,1}, \ldots, Q_{t-k,N},\, T_{t-k,1}, \ldots, T_{t-k,N},\, V_{t-k}\right], \quad k = 0, \ldots, K-1. \tag{5.27}$$

The action input is:

$$\mathbf{u}_t = \left[\bar{i}_{t,1}, \ldots, \bar{i}_{t,N},\, u_{t,1}, \ldots, u_{t,N},\, I_t\right] \in \mathbb{R}^{2N+1}, \tag{5.28}$$

where $u_{t,n}$ is the duty cycle of module MB_n, I_t is the pack load current, and the per-cell average current is $\bar{i}_{t,n} = I_t\, u_{t,n}$.

5.5.1.3 Multi-Objective Balancing Objectives

The multi-objective balancing problem is to simultaneously minimise:

1. **SOC non-uniformity**: achieve uniform SOC across all cells to maximise usable pack capacity and prevent localised over-charge or over-discharge, quantified by:

$$SD(Q) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(Q_i - \bar{Q})^2}\,; \tag{5.29}$$

2. **Thermal non-uniformity**: limit temperature disparities arising from unequal balancing currents, with a target maximum difference $\Delta T_{\max} \leq 5\,^{\circ}\mathrm{C}$ [142], quantified by:

$$SD(T) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(T_i - \bar{T})^2}\,; \tag{5.30}$$

3. **Output voltage regulation**: maintain the pack output voltage V close to a desired setpoint V_d to satisfy load requirements.

The inertia coefficients from the general cost function (5.22) are instantiated for this application as $\alpha^{\mathrm{f}} = \alpha_Q$ (for SOC), $\alpha_y = \alpha_V$ (for voltage), and $\alpha^{\mathrm{s}} = \alpha_T$ (for temperature), with $\alpha_Q \leq 1$, $\alpha_V \leq 1$, and $\alpha_T \geq 1$ to reflect the fast electrical and slow thermal response characteristics, respectively.

5.5.2 Experimental Setup and Battery Pack Configuration

Both the neural model prediction performance and the closed-loop NMPC balancing performance are evaluated using a coupled MATLAB/Simulink and Python environment. A modular battery pack model with the converter-based active balancing circuit is constructed in Simulink and imported into Python as a callable module, enabling step-wise execution while the neural modelling and control algorithms run in Python.

The case study uses a pack of three series-connected cylindrical cells (Sanyo UR18650W, 2.3 Ah, 3.6 V), each represented by an electro-thermal model. Nominal parameters are listed in Table 5.1. Internal parameter drift is incorporated following [145] to emulate realistic cell behaviour. The ambient temperature is fixed at 25 °C and the initial SOCs are set to [90%, 80%, 70%] to represent an imbalanced starting condition.

Table 5.1: Battery cell parameters.

Property	Value	Unit
Cell number	3	—
Nominal OCV	3.6	V
Nominal capacity	2.3	Ah
Air heat transfer coefficient	10	$\mathrm{W\,m^{-2}\,K^{-1}}$
Cell surface heat transfer coef.	5	$\mathrm{W\,m^{-2}\,K^{-1}}$
Cell width	0.018	m
Cell height	0.065	m
Cell thickness	0.018	m
Ambient temperature	25	°C

Three standard automotive driving cycles, BJDST, FUDS, and US06 [146], are converted into battery working current profiles using the longitudinal vehicle model, covering a broad range of operating conditions. The current profiles and their statistical distributions are shown in Figure 5.5:

1. **BJDST**: Urban driving with frequent speed and acceleration fluctuations.
2. **FUDS**: Low-to-medium speed operation with significant regenerative braking events.
3. **US06**: Aggressive high-speed driving with rapid acceleration; repeated twice to compensate for its shorter duration.

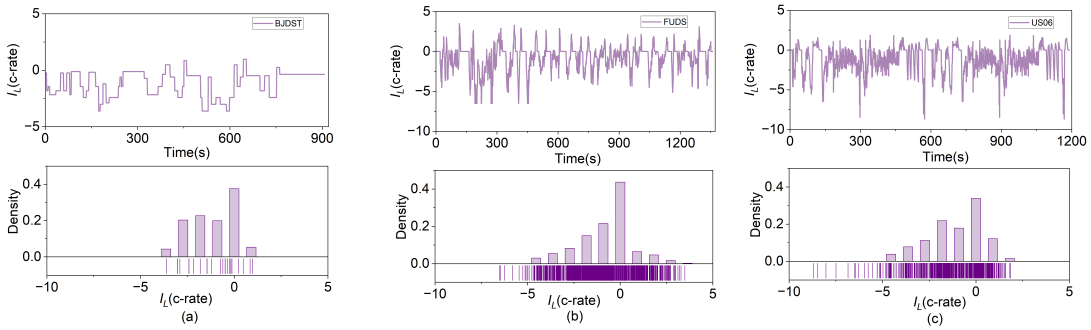


Figure 5.5: Battery working current and value density for the three dynamic current profiles used in the case study: (a) BJDST, (b) FUDS, (c) US06.

5.5.3 Dataset and Implementation Details

State-action pairs $(\mathbf{X}_t, \mathbf{u}_t)$ are collected at each time step from each driving cycle, with the one-step-ahead state $\mathbf{X}_{t+1}$ as the prediction target. The dataset is split 6:2:2 into training, validation, and test sets; sizes are reported in Table 5.2.

Table 5.2: Dataset sizes for training, validation, and testing.

Driving cycle	Training	Validation	Testing
BJDST	6,726	2,242	2,242
FUDS	9,730	3,243	3,243
US06	4,932	1,644	1,644

Neural models are trained end-to-end with MSE loss using the Adam optimiser (learning rate 10^{-3}, batch size 100, maximum 100 epochs) with early stopping based on validation loss. Prediction performance is evaluated using MAE, RMSE, and MAPE. For control deployment, the trained model is embedded in the NMPC framework with: prediction horizon $N_p = 20$, batch size $B = 100$, cost weighting factors $r_T = 5$, $r_Q = 1$, $r_V = 1$, inertia coefficients $\alpha_Q = 0.8$, $\alpha_T = 1.1$, $\alpha_V = 0.8$, and softmax sharpness $\lambda = 0.01$. All experiments are conducted on a workstation with 8 Intel CPU cores, 30 GB RAM, and an NVIDIA RTX 4080 GPU.

5.5.4 Ablation Study: Neural Prediction Model Components

Table 5.3 presents the ablation study comparing four model configurations: single-task baseline, full multi-task model, model without position embedding ($w/o\ PE$), and model without the TimesNet block ($w/o\ TN$).

The full multi-task model consistently achieves the lowest prediction errors across all driving cycles. The improvement is most pronounced for temperature prediction, demonstrating the advantage of jointly modelling coupled electro-thermal dynamics where temperature evolution shares heat-generation mechanisms with the electrical states. Removing the positional embedding ($w/o\ PE$) degrades accuracy particularly under the dynamic US06 profile, confirming that explicit temporal-position information

Table 5.3: Ablation study of different model components under various test datasets (SOC and temperature errors averaged over cells).

Cycle	Configuration	SOC		Temperature (°C)		Voltage (V)	
		MAE	RMSE	MAE	RMSE	MAE	RMSE
FUDS	Single-task	0.192	1.88	0.053	0.36	0.401	0.524
	Multi-task (full)	**0.086**	**0.95**	**0.019**	**0.14**	**0.303**	**0.376**
	w/o PE	0.113	1.25	0.031	0.21	0.332	0.417
	w/o TN	0.129	1.38	0.037	0.26	0.351	0.439
US06	Single-task	0.218	2.01	0.061	0.39	0.428	0.556
	Multi-task (full)	**0.103**	**1.12**	**0.025**	**0.19**	**0.309**	**0.380**
	w/o PE	0.137	1.41	0.038	0.28	0.347	0.431
	w/o TN	0.156	1.58	0.044	0.32	0.368	0.462
BJDST	Single-task	0.187	1.90	0.052	0.35	0.396	0.519
	Multi-task (full)	**0.083**	**1.04**	**0.018**	**0.12**	**0.304**	**0.375**
	w/o PE	0.108	1.29	0.029	0.20	0.331	0.414
	w/o TN	0.124	1.44	0.035	0.25	0.349	0.438

enhances sensitivity to rapid load variations. Excluding the TimesNet block ($w/o\ TN$) causes the most substantial deterioration for temperature prediction, reflecting its critical role in capturing the multi-scale temporal dependencies associated with heat accumulation and dissipation. These results confirm that both architectural components make complementary contributions under the evaluated conditions.

Figure 5.6 further illustrates the prediction results for SOC non-uniformity, temperature non-uniformity, and pack output voltage, confirming the enhanced performance of the full multi-task model across all datasets.

5.5.5 Overall Balancing Control Performance

Figure 5.7 presents the comprehensive balancing results in a 5×3 layout: columns (a)–(c) correspond to BJDST, FUDS, and US06; rows 1–5 show SOC evolution, $SD(Q)$, temperature evolution, $SD(T)$, and pack output voltage, respectively.

The proposed parallel sampling-based NMPC strategy effectively mitigates both SOC

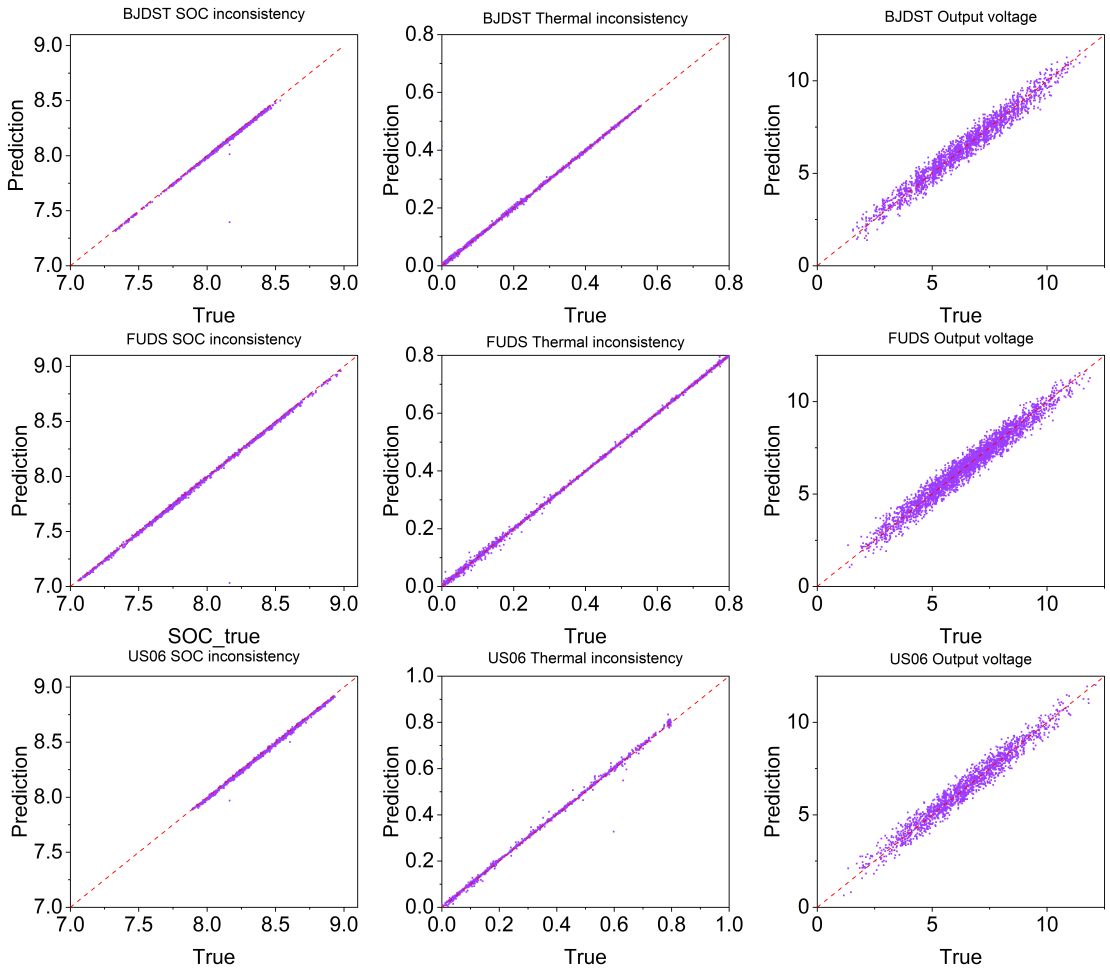


Figure 5.6: Prediction results for SOC non-uniformity, temperature non-uniformity, and pack output voltage under the three driving cycle datasets.

and temperature inhomogeneities while simultaneously maintaining the output voltage at the desired level. The SOC non-uniformity gradually decreases to near zero under all profiles; performance under US06 is slightly inferior because the aggressive dynamics induce larger thermal gradients, causing the controller to moderately prioritise thermal regulation over SOC balancing.

Initially, all cells start at room temperature (25 °C) while substantial SOC differences exist. During the early balancing stage, temperature differences temporarily increase as unequal balancing currents are applied. As SOC non-uniformity decreases and temperature differences become more prominent, the inertia-aware 2D cost map adaptively

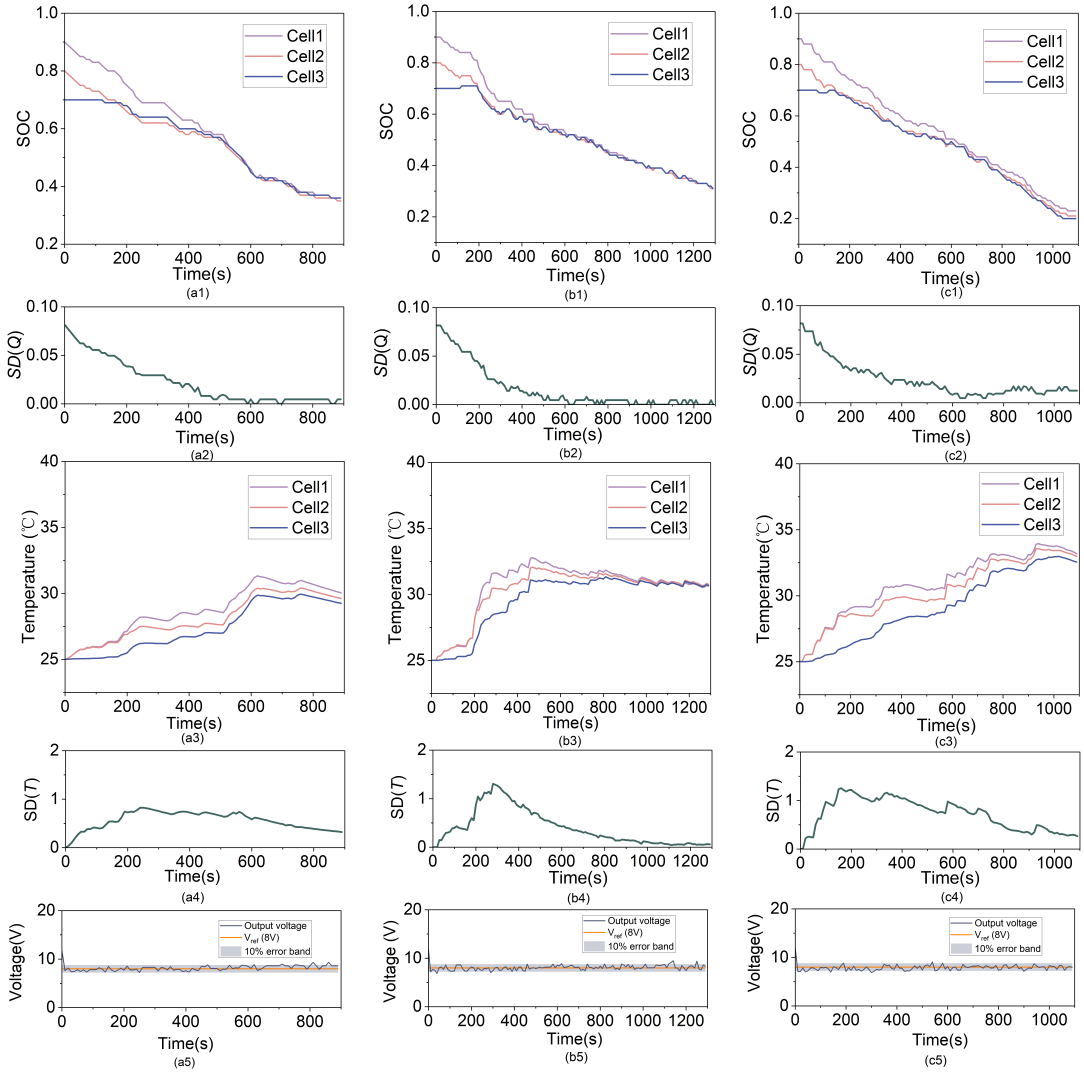


Figure 5.7: Balancing control results under three driving cycle profiles: (a) BJDST; (b) FUDS; (c) US06. Rows 1–5 show SOC trajectories, SOC non-uniformity $SD(Q)$, temperature trajectories, temperature non-uniformity $SD(T)$, and pack output voltage.

shifts the control priority toward thermal regulation, gradually reducing temperature inhomogeneity while maintaining approximate SOC uniformity. The pack output voltage regulation error remains within $\pm 10\%$ of the target (8 V) across all conditions and without requiring an additional power converter. The proposed framework exploits the existing active balancing infrastructure to satisfy all three objectives simultaneously.

Figure 5.8 further validates the strategy by showing the relationship between bal-

ancing current and cell SOC. The step-like current distribution, where cells with higher initial SOC discharge preferentially, is consistent with the fundamental principles of effective SOC balancing.

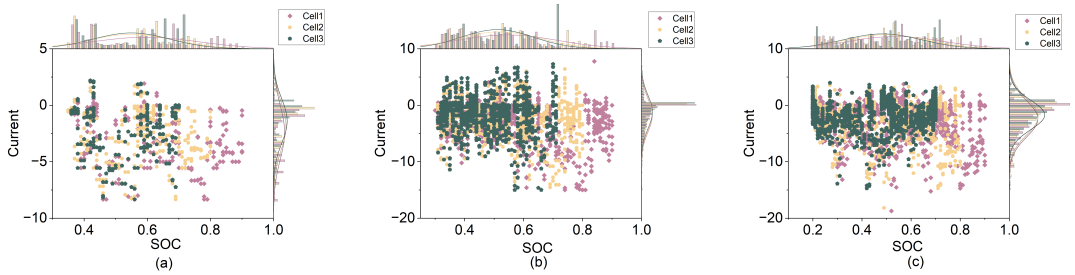


Figure 5.8: Balancing current distribution for individual cells: (a) BJDST; (b) FUDS; (c) US06. The step-like SOC–current profile confirms effective cell-level energy redistribution.

5.5.6 Parameter Sensitivity Analysis

Effect of prediction horizon N_p. Figure 5.9 shows the balancing results for $N_p \in \{10, 20, 50\}$ under the US06 profile. Increasing N_p significantly reduces temperature non-uniformity, as the longer horizon better captures the cumulative thermal response characteristic of the slow-dynamics state. The SOC non-uniformity slightly increases, reflecting the inherent trade-off between short-term electrical precision and long-term thermal regulation. Output voltage is insensitive to N_p variations. This behaviour is consistent with the framework's inertia-aware cost design: the slow-state inertia coefficient $\alpha_T > 1$ makes the cost function particularly sensitive to the length of the thermal prediction horizon, while the fast-state coefficient $\alpha_Q < 1$ limits the impact of horizon extension on SOC penalties.

Effect of cost function weighting factors. Figure 5.10 compares two weighting configurations: ($r_T = 5, r_Q = 1, r_V = 1$) versus ($r_T = 2, r_Q = 1, r_V = 1$). Increasing r_T effectively reduces temperature non-uniformity at the expense of a slight increase in SOC balancing error. This validates the framework's flexibility for application-specific tuning:

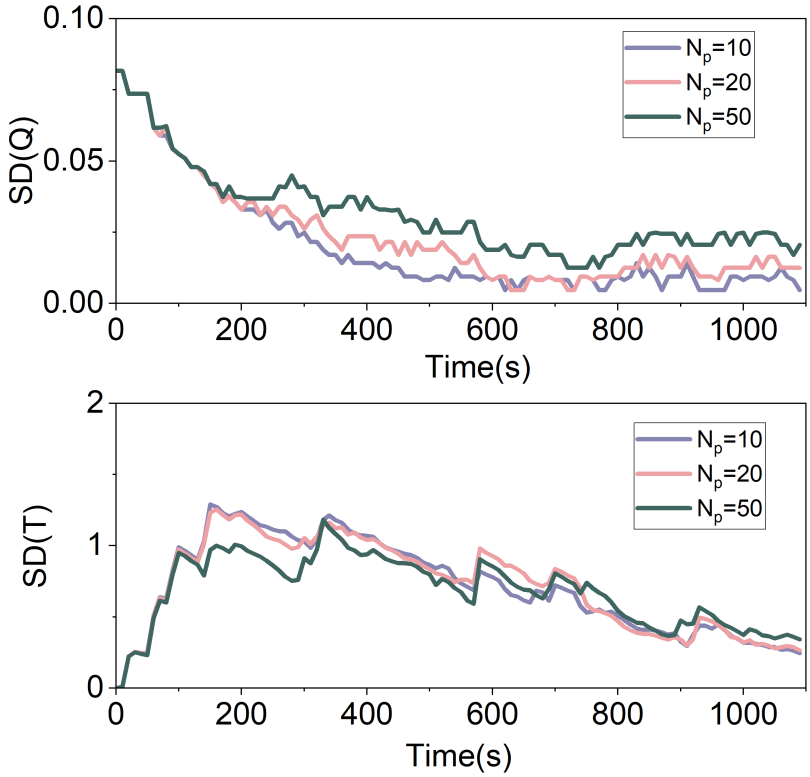


Figure 5.9: SOC and thermal non-uniformity under prediction horizons $N_p = 10$, 20, and 50 (US06 driving cycle).

the scalar weighting factors provide intuitive and decoupled handles for adjusting the multi-objective trade-off.

5.5.7 2D Cost Map Visualisation and Analysis

Figure 5.11 presents representative 2D cost maps for different action samples. The SOC and voltage columns display a clear colour gradient transitioning from dark (near-term) to light (long-term), while the temperature columns become progressively darker for later prediction steps. This directly reflects the inertia adjustment design: α_Q^j and α_V^j decay with step j (reducing electrical penalties at longer horizons), while α_T^j grows with j (amplifying cumulative temperature deviations). These observations confirm the physical consistency and interpretability of the proposed inertia-aware 2D cost map, which provides a visually transparent basis for the weighted aggregation of candidate

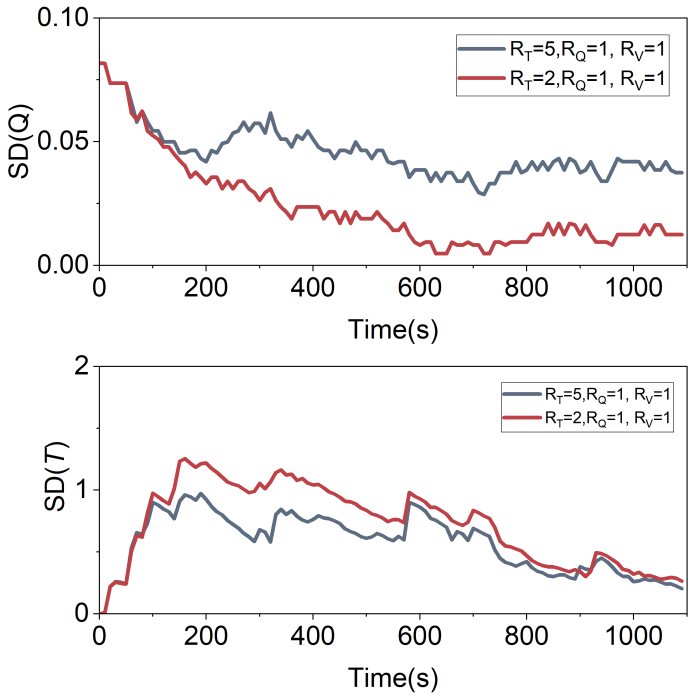


Figure 5.10: SOC and thermal non-uniformity under different cost function weighting factor configurations.

actions.

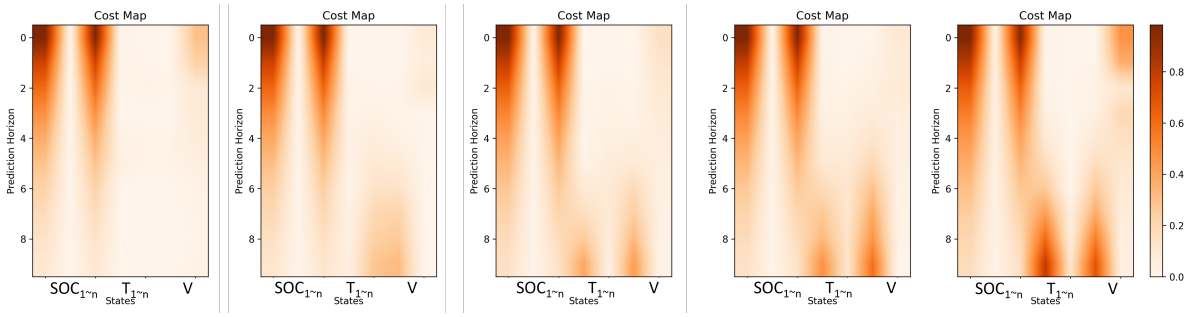


Figure 5.11: Visualised 2D cost map examples for different action samples at one sampling step.

5.5.8 Computation Efficiency Analysis

Table 5.4 reports balancing performance and computation time per control command for batch sizes $B \in \{100, 500, 1000\}$ under the US06 profile.

Table 5.4: Balancing performance and computation time under different batch sizes B (US06, $N_p = 20$).

Batch B	Mean $SD(Q)$	Mean $SD(T)$	Mean V_e	Time/cmd (s)
1000	0.023	0.742	0.473	0.602
500	0.045	0.576	0.367	0.330
100	0.029	0.650	0.459	**0.121**

At $B = 100$, each balancing command is computed in only 0.121 s, confirming real-time feasibility. Importantly, increasing B tenfold from 100 to 1000 increases computation time by only 5× rather than 10×, owing to the inherent parallelism of GPU batch inference. Balancing performance is robust across all batch sizes, validating the effectiveness of the softmax weighting mechanism even with a moderate candidate pool.

5.5.9 Output Voltage Regulation under Multiple Setpoints

Figure 5.12 presents simulation results under two target voltage setpoints of 8 V and 9 V, compared with an uncontrolled baseline. The proposed framework accurately regulates the output voltage to both setpoints within a ±10% error band throughout the entire simulation. Crucially, this performance is achieved without a dedicated power converter for DC-link stabilisation: the control algorithm coordinates balancing currents through the existing active balancing infrastructure to simultaneously satisfy SOC balancing, thermal regulation, and voltage control objectives, providing an integrated and resource-efficient solution.

5.5.10 Vision-Based Load Prediction for EV Deployment

The parallel-inference neural MPC framework relies on an accurate current load profile as an exogenous input at each control step. In practical EV deployment, the future load profile is not known a priori: it is determined by the vehicle speed trajectory, which itself depends on the road environment and driver behaviour. Providing the balancing controller with short-horizon load forecasts, rather than only instantaneous measurements, enables proactive and anticipatory balancing decisions. To this end, this

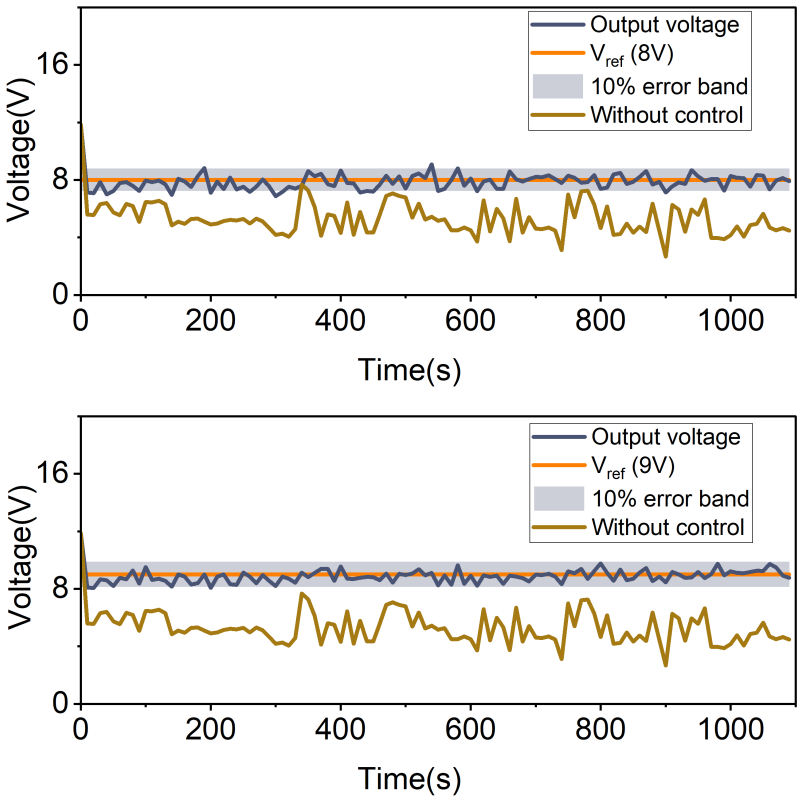


Figure 5.12: Output voltage regulation under setpoints of 8 V and 9 V, and the uncontrolled baseline.

subsection introduces a vision-based end-to-end load prediction pipeline. A convolutional neural network (CNN) extracts spatial features from forward-facing camera frames, which are temporally aggregated into a 2D feature representation and fed into a sequence prediction model to forecast future vehicle speed trajectories. The predicted speed profiles are converted into battery current load profiles through a physics-based vehicle dynamics module. The architecture is illustrated in Figure 5.13.

5.5.10.1 End-to-End Speed Prediction Model

The vision-based speed prediction component processes each incoming sequence of L consecutive video frames (with $L = 10$ in this work) through a three-stage pipeline: per-frame spatial feature extraction via a CNN encoder, temporal aggregation into a structured 2D representation, and multi-step speed forecasting via a sequence prediction

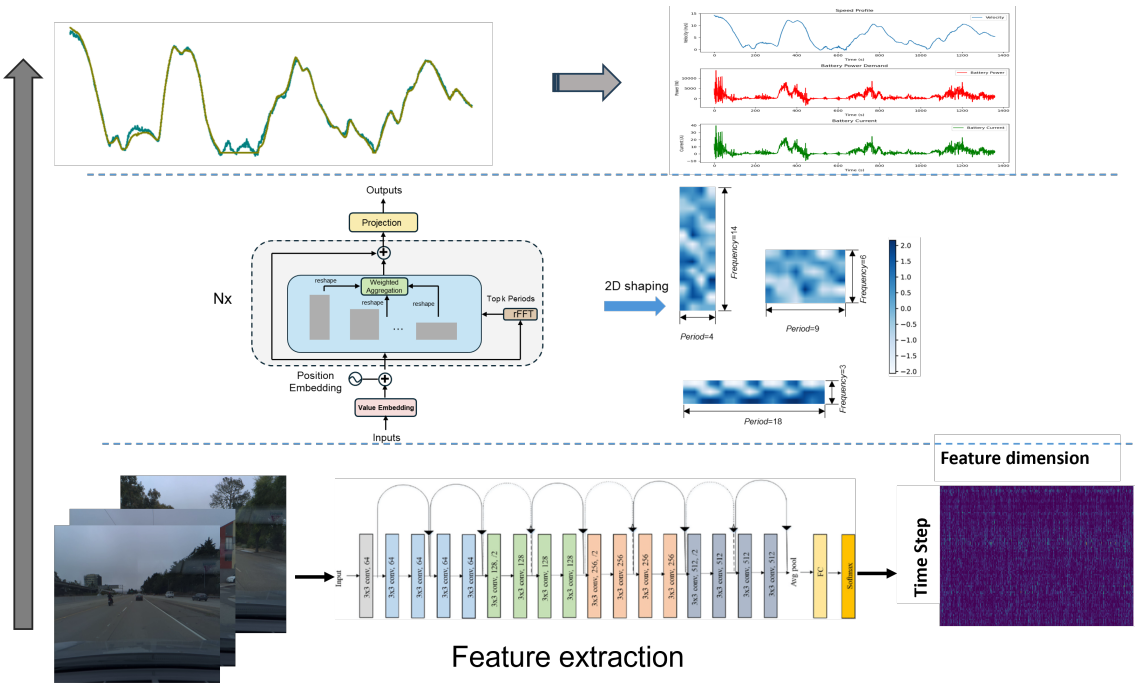


Figure 5.13: Overview of the vision-based load prediction pipeline.

model.

Each video frame $\mathbf{F}_\tau \in \mathbb{R}^{H\times W\times 3}$ is first passed through a CNN encoder ϕ_{enc} to produce a compact spatial feature vector:

$$\mathbf{f}_\tau = \phi_{\text{enc}}(\mathbf{F}_\tau) \in \mathbb{R}^{d_f}. \tag{5.31}$$

The encoder adopts a ResNet-inspired [147] backbone composed of five convolutional blocks, each containing a 3×3 convolution layer, batch normalisation, and a ReLU activation, followed by global average pooling and a fully connected projection to $\mathbb{R}^{d_f}$. Input frames are resized to 224×224 pixels prior to encoding. Encoder weights are initialised from ImageNet pre-training and fine-tuned jointly with the downstream sequence predictor in an end-to-end training scheme.

The resulting per-frame feature vectors are subsequently organised into a 2D temporal representation by stacking the features extracted from L consecutive frames along

the time axis:

$$\mathbf{Z}_t = \begin{bmatrix} \mathbf{f}_{t-L+1}^\top \\ \vdots \\ \mathbf{f}_t^\top \end{bmatrix} \in \mathbb{R}^{L \times d_f}. \tag{5.32}$$

This 2D structure mirrors the state–action input matrix $\mathbf{X}_t$ introduced in Section 5.3.2: rich multi-variate temporal information is organised as a structured matrix rather than a flattened vector, enabling more effective spatio-temporal feature extraction by the downstream predictor and maintaining architectural consistency with the broader neural modelling framework of this chapter.

This structured representation is then passed to a sequence prediction model ψ, which forecasts the vehicle speed trajectory over a prediction horizon of H_p steps:

$$\hat{\mathbf{v}}_{t+1:t+H_p} = \psi\big(\mathbf{Z}_t\big) \in \mathbb{R}^{H_p}, \tag{5.33}$$

where $\hat{v}_{t+k}$ denotes the predicted speed at step $t+k$ in $\mathrm{m\,s^{-1}}$. The predictor ψ consists of a 1D convolutional temporal encoder applied along the time axis of $\mathbf{Z}_t$, followed by an MLP decoder that maps the encoded representation to the H_p-step forecast. The complete pipeline — encoder ϕ_{enc} and predictor ψ — is trained end-to-end by minimising the mean squared prediction error over the forecast horizon:

$$\mathscr{L}_{\text{speed}} = \frac{1}{H_p} \sum_{k=1}^{H_p} \big(\hat{v}_{t+k} - v_{t+k}\big)^2, \tag{5.34}$$

using the Adam optimiser with a learning rate of 10^{-3}, a batch size of 64, and early stopping based on validation MSE.

5.5.10.2 Physics-Based Speed-to-Load Conversion

The predicted speed trajectory $\hat{\mathbf{v}}_{t+1:t+H_p}$ is converted to a battery current load profile through a longitudinal vehicle dynamics model. The required traction force at vehicle speed v is:

$$F_{\text{trac}}(v, \dot{v}) = m\,\dot{v} + \tfrac{1}{2}\,\rho_{\text{air}}\,C_d\,A_f\,v^2 + m\,g\,C_r \cos\theta + m\,g \sin\theta, \tag{5.35}$$

where $\dot{v} = (v_{t+1} - v_t)/\Delta t$ is the finite-difference estimate of acceleration. The corresponding mechanical power demand is $P_{\text{mech}} = F_{\text{trac}} \cdot v$. Accounting for drivetrain and motor efficiency, the electrical power drawn from or returned to the battery pack is:

$$P_{\text{elec}}(v, \dot{v}) = \begin{cases} \dfrac{P_{\text{mech}}}{\eta_{\text{trans}}\,\eta_{\text{motor}}}, & P_{\text{mech}} > 0 \quad \text{(propulsion)}, \\ P_{\text{mech}} \cdot \eta_{\text{trans}}\,\eta_{\text{motor}}, & P_{\text{mech}} \leq 0 \quad \text{(regenerative braking)}. \end{cases} \tag{5.36}$$

The predicted battery pack current at each future step is then obtained as:

$$\hat{I}_{t+k} = \frac{P_{\text{elec}}\left(\hat{v}_{t+k}, \hat{\dot{v}}_{t+k}\right)}{V_{\text{battery}}}, \quad k = 1, \ldots, H_p, \tag{5.37}$$

where $\hat{\dot{v}}_{t+k} = (\hat{v}_{t+k} - \hat{v}_{t+k-1})/\Delta t$.

The vehicle parameters employed in this module are listed in Table 5.5. The parameter values are selected to be representative of a mid-size passenger EV operating on highway routes, consistent with the target deployment scenario of this study. Specifically, the air density and gravitational acceleration correspond to standard sea-level atmospheric conditions; the aerodynamic drag coefficient, frontal area, rolling resistance coefficient, and vehicle mass fall within the ranges widely reported for mid-size passenger vehicles; the drivetrain and motor efficiency values reflect nominal operating conditions of production EV powertrains [148]; and the nominal pack voltage of 350 V is representative of contemporary mid-range EV battery systems, which typically operate in the 300–400 V range [149]. The road gradient is set to $\theta = 0$ as a simplifying assumption appropriate to the predominantly flat highway corridor of Interstate 280 featured in the dataset.

5.5.10.3 Experimental Evaluation

The vision-based model is trained and evaluated on the **comma2k19** dataset [150], a publicly available real-world highway driving dataset. The dataset comprises over 33 hours of commute footage recorded on California's Interstate 280, divided into 2,019 one-minute segments. Data were collected using comma EON devices equipped with a road-facing RGB camera operating at 20 Hz (1164 × 874 pixels), GPS, thermometers, a 9-axis

Table 5.5: Vehicle dynamics parameters used in the physics-based speed-to-load conversion module.

Parameter	Symbol	Value	Unit	Description
Vehicle mass	m	1500	kg	Total vehicle mass
Aerodynamic drag coeff.	C_d	0.30	—	Air resistance coefficient
Air density	ρ_{air}	1.225	$\mathrm{kg\,m^{-3}}$	Standard sea-level value
Frontal area	A_f	2.5	$\mathrm{m^2}$	Vehicle frontal cross-section
Rolling resistance coeff.	C_r	0.01	—	Tyre–road resistance
Gravitational accel.	g	9.81	$\mathrm{m\,s^{-2}}$	Standard gravity
Road gradient	θ	0	rad	Flat road assumption
Transmission efficiency	η_{trans}	0.90	—	Drivetrain efficiency
Motor efficiency	η_{motor}	0.85	—	Electric motor efficiency
Battery voltage	V_{battery}	350	V	Nominal pack output voltage

IMU, and full CAN bus data. The highway setting spans a wide range of vehicle speeds, acceleration events, and traffic interactions, providing sufficiently diverse supervision for high-power operating regimes. Some of the segment frames are subsampled to 5 Hz, and a sliding window of $L = 10$ consecutive frames is used as the model input, yielding a total of approximately 29,143 labelled samples across the full dataset. The dataset is partitioned in a 7:1:2 ratio into training, validation, and test splits based on non-overlapping video segments, resulting in 20,400 training samples, 2,914 validation samples, and 5,829 test samples, respectively. This segment-level partitioning ensures that no temporal leakage occurs between splits.

The pipeline is evaluated on the held-out test split comprising 5,829 samples. Single-step-ahead speed prediction accuracy is assessed using the mean absolute error (MAE), root mean squared error (RMSE), mean squared error (MSE), and coefficient of determination (R^2), with results reported in Table 5.6.

Table 5.6: Single-step-ahead speed prediction performance on the test dataset.

Metric	Value	Unit
MAE	0.277	$\mathrm{m\,s^{-1}}$
RMSE	0.447	$\mathrm{m\,s^{-1}}$
MSE	0.200	$(\mathrm{m\,s^{-1}})^2$
R^2	0.991	—

The model attains an R^2 of 0.991, indicating that the predicted trajectories account for over 99% of the variance in the ground-truth speed signal across the test set. The RMSE of 0.447 $\mathrm{m\,s^{-1}}$ represents less than 3% of the observed speed range (0–14.85 $\mathrm{m\,s^{-1}}$), confirming strong overall tracking fidelity.

To characterise prediction accuracy across different operating conditions, the test samples are partitioned into three speed regimes; the corresponding results are summarised in Table 5.7. In the mid-speed range (3–8 $\mathrm{m\,s^{-1}}$), the model performs best, achieving an RMSE of 0.198 $\mathrm{m\,s^{-1}}$ and an MAE of 0.157 $\mathrm{m\,s^{-1}}$. In the low-speed regime ($v < 3\ \mathrm{m\,s^{-1}}$), prediction error is higher (RMSE = 0.473 $\mathrm{m\,s^{-1}}$), primarily because visual features provide limited discriminative cues ahead of imminent stop-and-go events: the camera field of view conveys insufficient information about traffic queues or signal states that govern full stops. In the high-speed regime ($v \geq 8\ \mathrm{m\,s^{-1}}$), prediction is slightly degraded (RMSE = 0.559 $\mathrm{m\,s^{-1}}$) during prolonged near-constant-velocity cruise segments, where inter-frame visual differences are minimal and temporal context carries comparatively little predictive signal. Despite these localised degradations, only 4.4% of test samples exhibit an absolute error exceeding 1.0 $\mathrm{m\,s^{-1}}$, and these are predominantly concentrated at sharp speed transitions rather than distributed uniformly across the trajectory.

Table 5.7: Regime-level speed prediction performance on the test dataset.

Speed Regime	Samples (%)	MAE ($\mathrm{m\,s^{-1}}$)	RMSE ($\mathrm{m\,s^{-1}}$)
Low speed ($v < 3\ \mathrm{m\,s^{-1}}$)	35.9%	0.353	0.473
Mid speed ($3 \leq v < 8\ \mathrm{m\,s^{-1}}$)	29.4%	0.157	0.198
High speed ($v \geq 8\ \mathrm{m\,s^{-1}}$)	34.7%	0.299	0.559
Overall	**100%**	**0.277**	**0.447**

The speed prediction errors are propagated through the physics-based conversion module to assess their effect on the derived battery current load profiles. Because the traction force model (Equation 5.35) is a smooth, monotone function of speed and acceleration, moderate speed prediction errors incur correspondingly bounded deviations in the predicted current: a speed error of 0.447 $\mathrm{m\,s^{-1}}$ (RMSE) at typical highway speeds

propagates to a current error well within the tolerance band required by the downstream NMPC optimisation. The resulting current forecasts therefore provide the balancing controller with a reliable short-horizon load estimate at each control step, enabling anticipatory SOC redistribution ahead of high-demand events such as aggressive acceleration. The modular design of the pipeline, with the visual predictor and the physics conversion module trained and deployed independently, further facilitates targeted improvements to either component without requiring full system retraining.

Figure 5.14 presents a representative test segment drawn from the test dataset, together with the battery current load profile derived through the physics-based conversion module. The speed trajectory exhibits broad operational diversity: sustained high-speed cruising at up to approximately 14.8 $\mathrm{m\,s^{-1}}$, repeated deceleration-to-stop events, and extended phases of low-speed manoeuvring. These dynamics translate directly into a highly variable battery current profile — peak motoring current reaches approximately 40 A during aggressive acceleration transients, while regenerative braking events produce negative current excursions of up to 10 A. The pronounced temporal variability of the load profile underscores the necessity of accurate short-horizon speed forecasting for proactive balancing control.

5.5.10.4 Integration with the Battery Balancing Control Framework

The predicted current load profile $\hat{\mathbf{I}}_{t+1:t+H_p}$, obtained by applying Equation (5.37) to the vision-model forecasts, serves as the exogenous load input to the parallel sampling-based NMPC framework. Within each control step, candidate action sequences are evaluated by the multi-task neural model under the predicted load profile, replacing the instantaneous load measurement used in the baseline configuration. This perception-informed control architecture enables the balancing controller to anticipate imminent high-current demands, such as those arising from aggressive acceleration events, and initiate proactive SOC redistribution and thermal management before cell imbalances can develop. The CNN-based speed predictor and the multi-task balancing model are trained independently and deployed in a modular, plug-and-play fashion, such that

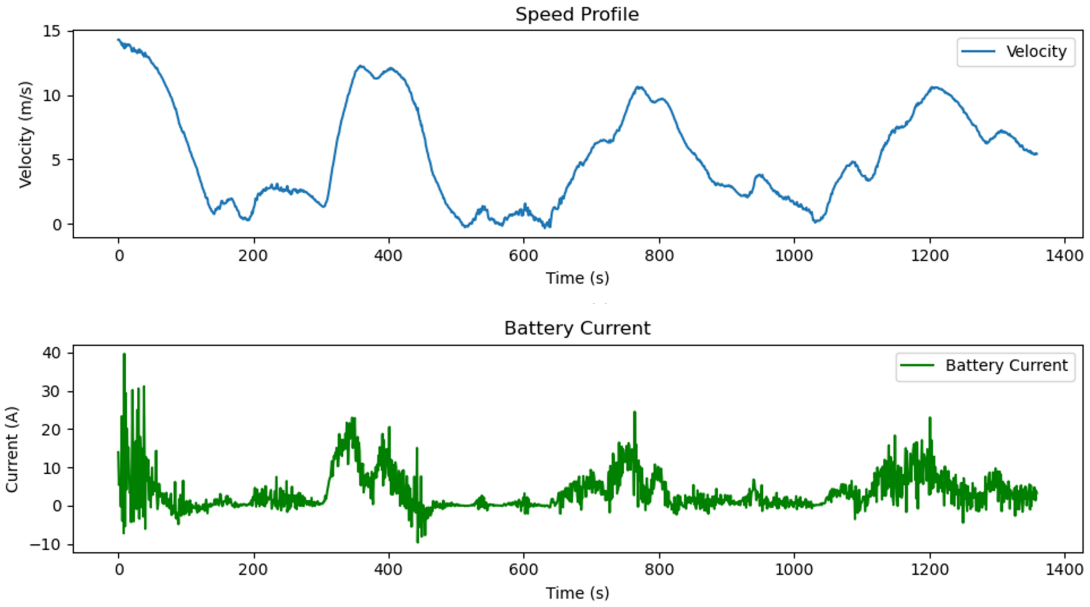


Figure 5.14: Representative test segment from the test dataset: (top) measured vehicle speed profile; (bottom) corresponding battery current load profile.

advances in visual odometry or driving behaviour prediction can be directly leveraged to improve balancing performance without modifying the core control framework.

5.6 Chapter Summary

This chapter has presented a 2D Cost Map-Guided Parallel-Inference Neural modelling and Predictive Control framework for the coupled electrical and thermal dynamics of a converter-based modular battery pack. The framework addresses the challenge of predicting and balancing states with different response times while enabling efficient multi-objective control without an explicit pack-level state-space model or gradient-based neural-network optimisation.

The main conclusions are as follows:

1. A control-oriented, multi-task deep neural model was developed for multi-state coupled dynamical systems. The 2D state-action input formulation, which integrates historical state sequences across all subsystems and state types with the current control action vector, enables simultaneous prediction of fast-dynamics

states, slow-dynamics states, and system-level outputs under multiple candidate actions. Applied to battery pack balancing, the model achieves consistently lower prediction errors across all three driving cycle datasets compared to single-task baselines, with maximum SOC MAE below 0.11 and temperature MAE below 0.03 per cell.

2. A decomposed multi-rate prediction architecture was validated in which an MLP subnetwork captures fast single-step electrical transitions while a TimesNet block with downstream MLP decoder captures the multi-scale temporal dependencies of thermally governed slow dynamics. An ablation study confirmed that positional embedding and the TimesNet block make complementary contributions: removing either component degrades prediction accuracy, with the effect most pronounced for temperature prediction under dynamic load profiles.

3. An inertia-aware 2D cost map was introduced as a physics-motivated and interpretable multi-objective evaluation metric. By assigning state-specific exponential inertia coefficients—decaying ($\alpha^{\mathrm{f}} \leq 1$) for fast states and amplifying ($\alpha^{\mathrm{s}} \geq 1$) for slow states—the cost map represents their different temporal responses along the prediction horizon. In the battery case study, the coefficients $\alpha_Q = 0.8$, $\alpha_T = 1.1$, $\alpha_V = 0.8$ provided the basis for weighted action selection.

4. A parallel sampling-based neural MPC framework was proposed that exploits GPU batch processing to evaluate candidate action trajectories without gradient computation. Per-command execution time at $B = 100$ is 0.121 s, supporting real-time execution for the tested configuration, while a tenfold increase in B results in approximately a fivefold increase in computation time. Simulation studies under three driving cycles (BJDST, FUDS, US06) demonstrated simultaneous multi-objective balancing: maximum temperature difference within 5 °C, terminal SOC imbalance below 1%, and output-voltage regulation error within ±10%.

5. The sensitivity analyses confirmed that the prediction horizon N_p primarily governs slow-state (thermal) regulation accuracy, with longer horizons improving

thermal performance at marginal fast-state cost, a behaviour directly attributable to the inertia-aware cost design. The scalar weighting factors provide intuitive and decoupled tuning handles for adapting the multi-objective trade-off to application-specific requirements.

The proposed modelling and control concepts, including the 2D state-action input formulation, decomposed multi-rate prediction subnetworks, inertia-aware cost mapping, and sampling-based parallel optimisation, may be transferable to other multi-state systems with heterogeneous dynamics, subject to system-specific retraining, parameter selection, and validation. Future work will focus on experimental validation using real battery systems, extension to larger and more diverse battery-pack architectures, and robustness enhancement under battery ageing, parameter variations, and external disturbances.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

6.1 Summary of the Thesis

This thesis has addressed the fundamental challenge of developing accurate, computationally efficient, and physically consistent modelling and control frameworks for complex multi-physics engineering systems, where purely data-driven and purely physics-based approaches each exhibit critical limitations. Motivated by the need to bridge the complementary strengths of data-driven learning and first-principles physical knowledge, a unified framework of AI-orchestrated physics-informed modelling and predictive control has been proposed and validated across two industrially significant engineering domains in energy-intensive manufacturing and battery management: transient heat conduction modelling in steel reheating furnaces, and advanced battery management systems covering both single-cell charging control and multi-cell pack balancing.

The principal technical contributions have been developed and demonstrated, corresponding to Chapters 3, 4, and 5 of the thesis. Each contribution addresses a distinct but complementary aspect of the overarching research challenge: the fidelity-efficiency trade-off in physics-informed surrogate modelling, the systematic exploitation of historical operational data in repetitive control processes, and the multi-objective real-time

control of heterogeneous coupled systems. Together, they establish a coherent methodological framework for AI-orchestrated engineering systems that is grounded in physical principles, validated against industrially representative benchmarks, and designed for generality beyond the specific case studies considered.

6.2 Conclusions

6.2.1 Physics-Informed EngGeneNet for 2D Transient Heat Conduction Modelling

Chapter 3 demonstrated that embedding governing physical laws directly into the architecture and training process of a deep neural network can simultaneously resolve the long-standing computational bottleneck of real-time temperature field prediction and the data scarcity challenge inherent in industrial reheating furnace environments.

The key conclusions, together with the principal assumptions and scope of applicability, are as follows:

1. EngGeneNet combines a Crank–Nicolson-based 2D physics-informed loss, hard-constrained convective and radiative boundary conditions, and the temperature-dependent Fourier number embedded through the Engineering-Gene module. This formulation improves physical consistency and learning efficiency: ablation studies show at least a 90% reduction in MAE and an 85% reduction in RMSE relative to the plain physics-informed UNet baseline.

2. Knowledge distillation from the ADI solver and transfer learning reduce the dependence on dense industrial labels. Under the tested changes in stock batches and operating conditions, the framework achieves greater than 90% prediction-error reduction and a 75% reduction in retraining time. These results assume that the governing heat-transfer equation, boundary conditions, and parameter ranges used for adaptation remain representative of the target process.

3. On a pilot-scale walking-beam bloom reheating furnace, EngGeneNet achieves accuracy comparable to ADI while reducing per-frame computation time by 96%, enabling real-time closed-loop use.

The present validation is limited to 2D rectangular stock and the tested operating range; extension to irregular or 3D geometries, stronger thermo-mechanical coupling, and substantial distribution shift requires further validation. Nevertheless, the physics-informed CNN and Eng-Gene principles are transferable to other distributed-parameter thermal systems when their governing relations can be specified reliably.

6.2.2 Historical Data-Driven Self-Learning Predictive Control for Repetitive Processes

Chapter 4 established a principled and computationally tractable mechanism for harnessing the information embedded in historical charging cycles to continuously improve closed-loop control performance, addressing the gap between the trial-to-trial learning capability of iterative learning control and the real-time constraint-handling and predictive optimisation strengths of MPC.

The key conclusions, together with the principal assumptions and scope of applicability, are as follows:

1. The cascading linearised thermoelectric model preserves the essential coupling between charging current and heat generation while keeping the MPC problem convex. Experimental validation gives voltage RMSE of approximately 3.4 mV and internal-temperature RMSE of approximately 0.82°C, providing a tractable predictive model for self-learning charging control.

2. The historical-state convex mapping constraint uses the convex hull of previous trajectories to reduce the feasible search space, while the memory-function-enhanced cost transfers useful information from earlier charging iterations into the current optimisation. Together they improve computational predictability and progressively

refine the charging-current trajectory; the convex mapping constraint reduces MPC solve-time variability by approximately 80% without increasing the optimisation structure beyond the quadratic formulation.

3. Across the tested temperature thresholds, the method reduces charging duration by 70–100 s relative to the constrained GPC benchmark and remains robust to parameter uncertainties of up to 10%.

These findings assume repetitive charging tasks, sufficiently informative historical trajectories, and a model operating within the considered battery and temperature ranges. The present study does not establish performance over long-term ageing, broad cell-to-cell variability, or unseen charging protocols; these are important limitations. The framework is nevertheless generalisable to other repetitive constrained processes in which historical state-input data can be incorporated without violating safety constraints.

6.2.3 2D Cost Map-Guided Parallel-Inference Neural Modelling and Predictive Control Framework for Battery Pack Balancing

Chapter 5 developed an end-to-end AI-orchestrated framework for simultaneous multi-objective battery pack balancing, addressing the challenges of coupled electro-thermal dynamics, state inertia heterogeneity, and real-time multi-objective optimisation that collectively exceed the capability of conventional model-based or purely data-driven approaches.

The key conclusions, together with the principal assumptions and scope of applicability, are as follows:

1. The unified multi-task state-action model, using a structured 2D representation with decomposed fast- and slow-dynamics subnetworks, improves prediction of coupled electro-thermal states. Across the three driving-cycle datasets, maximum

SOC MAE remains below 0.11 and temperature MAE below 0.03 per cell; ablation results confirm the contribution of positional embedding and the TimesNet block.

2. The inertia-aware 2D cost map incorporates state-specific time constants into an interpretable multi-objective evaluation, while parallel Gaussian sampling and softmax-weighted aggregation provide gradient-free action selection. GPU batch inference achieves a per-command decision time of 0.121 s at batch size $B = 100$, supporting real-time operation without requiring a dedicated power converter.

3. Under the BJDST, FUDS, and US06 driving cycles, the framework maintains an inter-cell temperature difference below 5 °C, terminal SOC non-uniformity below 1%, and output-voltage error within ±10%.

These results assume that the learned predictor, cell configuration, thermal model, and action space remain representative of deployment conditions. Validation is currently limited to the studied pack scale and driving profiles, so larger packs, ageing cells, measurement noise, and distribution shifts remain open limitations. The structured predictor and inertia-aware cost formulation are potentially transferable to other coupled multi-state systems with heterogeneous dynamics, subject to retraining and recalibration.

6.3 Recommendations for Future Work

While the proposed frameworks have demonstrated strong performance in their respective case studies, several important directions for future research are identified to extend, strengthen, and broaden their applicability.

6.3.1 Advanced Physics-Informed Architectures for Complex Geometries and Multi-Physics Coupling

The EngGeneNet framework was developed for 2D transient heat conduction in rectangular steel stock. Extending the framework to three-dimensional geometries, irregular cross-sections, and multi-physics coupling between heat conduction and mechanical

deformation during rolling would significantly broaden its industrial applicability. Three-dimensional CNN architectures or hybrid CNN-graph network models could be explored for volumetric temperature field prediction. The Eng-Gene concept that encoding domain-specific nonlinearities as architectural primitives is equally applicable to the coupled thermo-mechanical problem, where constitutive relationships for thermal expansion and stress evolution could be embedded as additional Eng-Gene modules.

6.3.2 Uncertainty Quantification and Robust Physics-Informed Control

The frameworks developed in this thesis operate largely in a deterministic setting without principled uncertainty quantification. Incorporating Bayesian or ensemble-based uncertainty estimation into the neural prediction models would enable the derivation of confidence intervals for predicted states, which could be propagated into robust or chance-constrained MPC formulations. For the parallel sampling-based NMPC, uncertainty-aware cost maps could explicitly account for prediction uncertainty in the evaluation of candidate actions, improving robustness under model mismatch and distribution shift. For the self-learning MPC, a Bayesian treatment of the memory function and convex mapping constraint would provide formal guarantees on the improvement of control performance across iterations under bounded model uncertainty.

6.3.3 Extension to Multi-Agent and Distributed Energy Systems

The self-learning MPC and parallel-inference NMPC frameworks both operate on centralised single-system models. Extending these frameworks to distributed or multi-agent settings, such as coordinated charging of electric vehicle fleets, distributed energy storage networks, or cascaded industrial processes, represents a promising and practically relevant direction. In such settings, the memory function concept could be extended to encode shared historical information across multiple agents, enabling collective learning of system-level policies that improve coordination efficiency without requiring centralised

global optimisation.

6.3.4 Integration of Reinforcement Learning with Physics-Informed Structure

The self-learning MPC framework and the parallel sampling-based NMPC both employ optimisation-based approaches to exploit historical data and improve control performance. An interesting future direction is the integration of reinforcement learning (RL) with physics-informed model architectures, where the RL policy is constrained by embedded physical laws to ensure safety-critical constraint satisfaction throughout exploration and exploitation. Physics-informed reward shaping where reward signals are designed to be consistent with conservation principles and physical feasibility conditions could substantially improve the sample efficiency and safety of RL-based control in the battery and thermal process domains.

The research presented in this thesis has demonstrated that the integration of artificial intelligence with physical laws and engineering domain knowledge represents a methodologically sound and practically powerful framework for modelling and controlling complex engineering systems. By systematically embedding governing equations, physically meaningful architectural components, and historical operational data into learning and optimisation frameworks, the proposed AI-orchestrated approaches achieve levels of accuracy, efficiency, and physical consistency that are individually unattainable by either purely data-driven or purely physics-based methods.

The two application domains addressed in this thesis, energy-intensive manufacturing (steel reheating furnace thermal modelling) and advanced battery management, are industrially significant in their own field and represent exemplary cases for a much broader class of energy-intensive, multi-physics engineering systems where the proposed methodological contributions are directly applicable. As AI technologies continue to mature and industrial systems generate increasingly rich operational data, the physics-informed and data-driven framework developed in this thesis provides a coherent and extensible foundation for future intelligent engineering systems that are simultaneously

accurate, interpretable, efficient, and trustworthy.

Published Papers

1. X. Han and K. Li, "2D cost map-guided deep neural modelling and control for battery pack balancing," *Applied Energy* (Under Review).

2. X. Han, K. Li, F. Gao, and Y. Zhou, "Historical data-driven self-learning control of battery charging with convex mapping constraints," *Journal of Energy Storage*, vol. 152, p. 120430, 2026.

3. X. Han, K. Li, C. Wang, J. Zhang, and Y. Hu, "Transient heat conduction modelling for real-time operation and control of steel-making reheating furnaces – a physics-informed knowledge distillation-assisted EngGeneNet approach," *Control Engineering Practice*, vol. 164, p. 106472, 2025.

4. X. Han, Y. Hu, and K. Li, "Dynamic modelling for improved furnace operation and control," in *Encyclopedia of Systems and Control Engineering*. Elsevier, 2026, pp. 245–261.

5. X. Han and K. Li, "AI-orchestrated data-driven modelling and control in battery management systems: A review," in *2024 6th International Conference on Industrial Artificial Intelligence (IAI)*. IEEE, 2024, pp. 1–7.

6. X. Han and K. Li, "Vision-Based Spatio-Temporal Convolutional Load Forecasting for Battery Pack Balancing in Electric Vehicles", in *2026 8th International Conference on Industrial Artificial Intelligence (IAI)*. IEEE, 2026, (Accepted).

BIBLIOGRAPHY

[1] A. Windmann, P. Wittenberg, M. Schieseck, and O. Niggemann, "Artificial intelligence in industry 4.0: A review of integration challenges for industrial systems," in *2024 IEEE 22nd International Conference on Industrial Informatics (INDIN)*. IEEE, Aug. 2024, p. 1–8. [Online]. Available: http://dx.doi.org/10.1109/INDIN58382.2024.10774364

[2] Q. Wang, Y. Li, and R. Li, "Integrating artificial intelligence in energy transition: A comprehensive review," *Energy Strategy Reviews*, vol. 57, p. 101600, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2211467X24003092

[3] S. Rani, A. Kataria, S. Kumar, and V. Karar, "A new generation cyber-physical system: A comprehensive review from security perspective," *Computers & Security*, vol. 148, p. 104095, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404824004000

[4] C. Giattino, E. Mathieu, V. Samborska, and M. Roser, "Data page: Scholarly publications in artificial intelligence," Our World in Data, 2023, part of: "Artificial Intelligence". Data adapted from the Center for Security and Emerging Technology (CSET). Archived on November 14, 2025. [Online]. Available: https://archive.ourworldindata.org/20251114-100207/grapher/annual-scholarly-publications-on-artificial-intelligence.html

[5] C. Giattino, E. Mathieu, V. Samborska, and M. Roser, , "Data page: External funding for privately held ai companies raising above $1.5 million," Our World

in Data, 2023, part of: “Artificial Intelligence”. Data adapted from Quid via the AI Index Report and the U.S. Bureau of Labor Statistics. Archived on November 19, 2025. [Online]. Available: https://archive.ourworldindata.org/20251119-141827/grapher/private-investment-in-artificial-intelligence.html

[6] B. Mahesh, “Machine learning algorithms-a review,” *International Journal of Science and Research (IJSR).[Internet]*, vol. 9, pp. 381–386, 2020.

[7] N. Sharma, R. Sharma, and N. Jindal, “Machine learning and deep learning applications-a vision,” *Global Transitions Proceedings*, vol. 2, no. 1, pp. 24–28, 2021.

[8] L. Zhang, S. Wang, and B. Liu, “Deep learning for sentiment analysis: A survey,” *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, no. 4, p. e1253, 2018.

[9] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah, “Transformers in vision: A survey,” *ACM Comput. Surv.*, vol. 54, no. 10s, Sep. 2022. [Online]. Available: https://doi.org/10.1145/3505244

[10] H. Ahlawat, N. Aggarwal, and D. Gupta, “Automatic speech recognition: A survey of deep learning techniques and approaches,” *International Journal of Cognitive Computing in Engineering*, vol. 6, pp. 201–237, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2666307424000573

[11] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, Y. Du, C. Yang, Y. Chen, Z. Chen, J. Jiang, R. Ren, Y. Li, X. Tang, Z. Liu, P. Liu, J.-Y. Nie, and J.-R. Wen, “A survey of large language models,” 2025. [Online]. Available: https://arxiv.org/abs/2303.18223

[12] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, “Extracting and composing robust features with denoising autoencoders,” in *Proceedings of the 25th international conference on Machine learning*, 2008, pp. 1096–1103.

[13] X. Xie, “A review of recent advances in surface defect detection using texture analysis techniques,” *ELCVIA: electronic letters on computer vision and image analysis*, pp. 1–22, 2008.

[14] Y. Wu, M. Yuan, S. Dong, L. Lin, and Y. Liu, “Remaining useful life estimation of engineered systems using vanilla lstm neural networks,” *Neurocomputing*, vol. 275, pp. 167–179, 2018.

[15] A. Paleyes, R.-G. Urma, and N. D. Lawrence, “Challenges in deploying machine learning: A survey of case studies,” *ACM Computing Surveys*, vol. 55, no. 6, p. 1–29, Dec. 2022. [Online]. Available: http://dx.doi.org/10.1145/3533378

[16] M. M. Forootan, I. Larki, R. Zahedi, and A. Ahmadi, “Machine learning and deep learning in energy systems: A review,” *Sustainability*, vol. 14, no. 8, 2022. [Online]. Available: https://www.mdpi.com/2071-1050/14/8/4832

[17] Y. Zhang, S. Zhang, and V. Dinavahi, “A survey of machine learning applications in advanced transportation systems: Trends, techniques, and future directions,” *eTransportation*, vol. 24, p. 100417, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2590116825000244

[18] G. Peng, Y. Cheng, Y. Zhang, J. Shao, H. Wang, and W. Shen, “Industrial big data-driven mechanical performance prediction for hot-rolling steel using lower upper bound estimation method,” *Journal of Manufacturing Systems*, vol. 65, pp. 104–114, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S027861252200142X

[19] F. Doshi-Velez and B. Kim, “Towards a rigorous science of interpretable machine learning,” 2017. [Online]. Available: https://arxiv.org/abs/1702.08608

[20] A. Barredo Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, and F. Herrera, “Explainable artificial intelligence (xai): Concepts, taxonomies,

opportunities and challenges toward responsible ai," *Information Fusion*, vol. 58, pp. 82–115, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1566253519308103

[21] S. Kasilingam, R. Yang, S. K. Singh, M. A. Farahani, R. Rai, and T. Wuest, "Physics-based and data-driven hybrid modeling in manufacturing: a review," *Production & Manufacturing Research*, vol. 12, no. 1, p. 2305358, 2024. [Online]. Available: https://doi.org/10.1080/21693277.2024.2305358

[22] C. Zhang, K. Li, and J. Deng, "Real-time estimation of battery internal temperature based on a simplified thermoelectric model," *Journal of Power Sources*, vol. 302, pp. 146–154, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378775315304390

[23] M. Kurucan, M. Özbaltan, Z. Yetgin, and A. Alkaya, "Applications of artificial neural network based battery management systems: A literature review," *Renewable and Sustainable Energy Reviews*, vol. 192, p. 114262, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1364032123011206

[24] M. F. McCarthy, "Singular surfaces and waves," in *Continuum Mechanics of Single-Substance Bodies*. Elsevier, 1975, pp. 449–521.

[25] J. P. Milaszewicz, "Improving jacobi and gauss-seidel iterations," *Linear Algebra and Its Applications*, vol. 93, pp. 161–170, 1987.

[26] X. Han, K. Li, C. Wang, J. Zhang, and Y. Hu, "Transient heat conduction modelling for real-time operation and control of steel-making reheating furnaces - a physics-informed knowledge distillation-assisted enggenenet approach," *Control Engineering Practice*, vol. 164, p. 106472, 2025.

[27] X. Han, K. Li, F. Gao, and Y. Zhou, "Historical data-driven self-learning control of battery charging with convex mapping constraints," *Journal of Energy Storage*, vol. 152, p. 120430, 2026.

[28] X. Han and K. Li, "2d cost map-guided deep neural modelling and control for battery pack balancing," *Applied Energy (Under Review)*.

[29] X. Han and K. Li, "Ai-orchestrated data-driven modelling and control in battery management systems: A review," in *2024 6th International Conference on Industrial Artificial Intelligence (IAI)*. IEEE, 2024, pp. 1–7.

[30] X. Han, Y. Hu, and K. Li, "Dynamic modeling for improved furnace operation and control," in *Encyclopedia of Systems and Control Engineering*. Elsevier, 2026, pp. 245–261.

[31] A. Dogan and D. Birant, "Machine learning and data mining in manufacturing," *Expert Systems with Applications*, vol. 166, p. 114060, 2021.

[32] K. G. Liakos, P. Busato, D. Moshou, S. Pearson, and D. Bochtis, "Machine learning in agriculture: A review," *Sensors*, vol. 18, no. 8, p. 2674, 2018.

[33] S. Athey, "The impact of machine learning on economics," in *The economics of artificial intelligence: An agenda*. University of Chicago Press, 2018, pp. 507–547.

[34] M. Gheisari, G. Wang, and M. Z. A. Bhuiyan, "A survey on deep learning in big data," in *2017 IEEE international conference on computational science and engineering (CSE) and IEEE international conference on embedded and ubiquitous computing (EUC)*, vol. 2. IEEE, 2017, pp. 173–180.

[35] S. Cai, Z. Wang, S. Wang, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks for heat transfer problems," *Journal of Heat Transfer*, vol. 143, no. 6, 2021.

[36] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.

[37] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving

nonlinear partial differential equations," *Journal of Computational physics*, vol. 378, pp. 686–707, 2019.

[38] L. Yang, X. Meng, and G. E. Karniadakis, "B-pinns: Bayesian physics-informed neural networks for forward and inverse pde problems with noisy data," *Journal of Computational Physics*, vol. 425, p. 109913, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0021999120306872

[39] M.-C. Lai, Y. Song, X. Yuan, H. Yue, and T. Zeng, "The hard-constraint pinns for interface optimal control problems," *SIAM Journal on Scientific Computing*, vol. 47, no. 3, pp. C601–C629, 2025. [Online]. Available: https://doi.org/10.1137/23M1601249

[40] A. D. Jagtap and G. E. Karniadakis, "Extended physics-informed neural networks (xpinns): A generalized space-time domain decomposition based deep learning framework for nonlinear partial differential equations," *Communications in Computational Physics*, vol. 28, no. 5, 11 2020. [Online]. Available: https://www.osti.gov/biblio/2282003

[41] W. Zhang and J. Li, "Cpinns: A coupled physics-informed neural networks for the closed-loop geothermal system," *Computers & Mathematics with Applications*, vol. 132, pp. 161–179, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0898122123000032

[42] R. Pan, F. Xiao, and M. Shen, "ro-pinn: A reduced order physics-informed neural network for solving the macroscopic model of pedestrian flows," *Transportation Research Part C: Emerging Technologies*, vol. 163, p. 104658, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X24001797

[43] Y. Bai, X. Chen, C. Gong, and J. Liu, "Impinn: Improved physics-informed neural networks for solving inverse problems," in *2024 International Conference on*

Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2024, pp. 189–198.

[44] Z. Mao, A. D. Jagtap, and G. E. Karniadakis, "Physics-informed neural networks for high-speed flows," *Computer Methods in Applied Mechanics and Engineering*, vol. 360, p. 112789, 2020.

[45] S. Cai, Z. Wang, C. Chryssostomidis, and G. E. Karniadakis, "Heat transfer prediction with unknown thermal boundary conditions using physics-informed neural networks," in *Fluids engineering division summer meeting*, vol. 83730. American Society of Mechanical Engineers, 2020, p. V003T05A054.

[46] B. Zhang, F. Wang, and L. Qiu, "Multi-domain physics-informed neural networks for solving transient heat conduction problems in multilayer materials," *Journal of Applied Physics*, vol. 133, no. 24, p. 245103, 06 2023.

[47] S. Wang, H. Zhang, and X. Jiang, "Physics-informed neural network algorithm for solving forward and inverse problems of variable-order space-fractional advection–diffusion equations," *Neurocomputing*, vol. 535, pp. 64–82, 2023.

[48] L. Xu, C. Yang, X. Xu, and N. Chen, "Physics-informed data-driven model of dehydration reaction stage in the sintering process of ternary cathode materials," *Control Engineering Practice*, vol. 147, p. 105929, 2024.

[49] K. Li, "ENG-GENES: A new genetic modelling approach for nonlinear dynamic systems," in *Proceedings of the 16th IFAC World Congress*, Prague, Czech Republic, 2005, pp. 162–167.

[50] J. Drgoňa, J. Arroyo, I. Cupeiro Figueroa, D. Blum, K. Arendt, D. Kim, E. P. Ollé, J. Oravec, M. Wetter, D. L. Vrabie, and L. Helsen, "All you need to know about model predictive control for buildings," *Annual Reviews in Control*, vol. 50, pp. 190–232, 2020.

[51] P. Karamanakos, E. Liegmann, T. Geyer, and R. Kennel, "Model predictive control of power electronic systems: Methods, results, and challenges," *IEEE Open Journal of Industry Applications*, vol. 1, pp. 95–114, 2020.

[52] G. Williams, P. Drews, B. Goldfain, J. M. Rehg, and E. A. Theodorou, "Aggressive driving with model predictive path integral control," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 1433–1440.

[53] R. Chi, Y. Hui, and Z. Hou, *Data-driven iterative learning control for discrete-time systems*. Springer, 2022.

[54] P. Schmitz, T. Faulwasser, and K. Worthmann, "Willems' fundamental lemma for linear descriptor systems and its use for data-driven output-feedback mpc," *IEEE Control Systems Letters*, vol. 6, pp. 2443–2448, 2022.

[55] C. De Persis and P. Tesi, "Formulas for data-driven control: Stabilization, optimality, and robustness," *IEEE Transactions on Automatic Control*, vol. 65, no. 3, pp. 909–924, 2020.

[56] J. G. Rueda-Escobedo, E. Fridman, and J. Schiffer, "Data-driven control for linear discrete-time delay systems," *IEEE Transactions on Automatic Control*, vol. 67, no. 7, pp. 3321–3336, 2022.

[57] Y. Zhou, K. Gao, D. Li, Z. Xu, and F. Gao, "Data-efficient constrained learning for optimal tracking of batch processes," *Industrial & Engineering Chemistry Research*, vol. 60, no. 43, pp. 15 658–15 668, 2021.

[58] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in *Proceedings of the 31st International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, E. P. Xing and T. Jebara, Eds., vol. 32, no. 1. Bejing, China: PMLR, 22–24 Jun 2014, pp. 387–395. [Online]. Available: https://proceedings.mlr.press/v32/silver14.html

[59] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017. [Online]. Available: https://arxiv.org/abs/1707.06347

[60] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, J. Dy and A. Krause, Eds., vol. 80. PMLR, 10–15 Jul 2018, pp. 1861–1870. [Online]. Available: https://proceedings.mlr.press/v80/haarnoja18b.html

[61] S. Fujimoto, H. van Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, J. Dy and A. Krause, Eds., vol. 80. PMLR, 10–15 Jul 2018, pp. 1587–1596. [Online]. Available: https://proceedings.mlr.press/v80/fujimoto18a.html

[62] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep reinforcement learning: A brief survey," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 26–38, 2017.

[63] K. Dong, Y. Luo, Y. Wang, Y. Liu, C. Qu, Q. Zhang, E. Cheng, Z. Sun, and B. Song, "Dyna-style model-based reinforcement learning with model-free policy optimization," *Knowledge-Based Systems*, vol. 287, p. 111428, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950705124000637

[64] A. Pozzi, A. Incremona, and D. Toti, "Imitation learning-driven approximation of stochastic control models," *Applied Intelligence*, vol. 55, no. 12, pp. 1–19, 2025.

[65] Y. Hu, C. Tan, J. Broughton, and P. A. Roach, "Development of a first-principles hybrid model for large-scale reheating furnaces," *Applied Energy*, vol. 173, pp. 555–566, 2016.

[66] E. Kilinç, D. Kaya, F. Ç. Kiliç, M. Eyidoğan, M. Özkaymak, O. Taylan, and W. Pedrycz, "An energy efficiency analysis of an industrial reheating furnace and an implementation of efficiency enhancements methods," *Energy exploration & exploitation*, vol. 32, no. 6, pp. 989–1003, 2014.

[67] D. Jones, "Available and emerging technologies for reducing greenhouse gas emissions from the iron and steel industry," *US EPA, Office of Air Quality Planning and Standards, Sector Policies and Programs Division*, p. 25, 2012.

[68] "Modelling and simulation of steel reheating processes under oxy-fuel combustion conditions – technical and environmental perspectives," *Energy*, vol. 185, pp. 730–743, 2019.

[69] J. C. Jaeger, *Application of the Theory Of Heat Conduction to Geothermal Measurements*. American Geophysical Union (AGU), 1965, ch. 2, pp. 7–23. [Online]. Available: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/GM008p0007

[70] J. Crank and P. Nicolson, "A practical method for numerical evaluation of solutions of partial differential equations of the heat-conduction type," in *Mathematical proceedings of the Cambridge philosophical society*, vol. 43. Cambridge University Press, 1947, pp. 50–67.

[71] C.-T. Wu, S.-W. Wu, R.-P. Niu, C. Jiang, and G. Liu, "The polygonal finite element method for solving heat conduction problems," *Engineering Analysis with Boundary Elements*, vol. 155, pp. 935–947, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S095579972300379X

[72] D. R. Croft and D. G. Lilley, *Heat transfer calculations using finite difference equations / David R. Croft and David G. Lilley.* London: Applied Science Publishers, 1977.

[73] B.-D. Nie and B.-Y. Cao, "Three mathematical representations and an improved adi method for hyperbolic heat conduction," *International Journal of Heat*

and Mass Transfer, vol. 135, pp. 974–984, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0017931018342418

[74] R. Čiegis, R. Čiegis, and O. Suboč, "Parallel 3d adi scheme for partially dimension reduced heat conduction problem," *Informatica*, vol. 33, no. 3, pp. 477–497, 2022.

[75] I. Bozsóki, A. Géczy, and B. Illés, "Component level modelling of heat transfer during vapour phase soldering with finite difference adi approach," *International Journal of Heat and Mass Transfer*, vol. 128, pp. 562–569, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0017931018331478

[76] J. M. Rhine and R. J. Tucker, "Modelling of gas-fired furnaces and boilers and other industrial heating processes," *(No Title)*, 1991.

[77] Y. Hu, J. Niska, J. Broughton, E. McGee, C.-K. Tan, A. Matthew, P. Roach, and S. MEFOS, "Zone modelling coupled with dynamic flow pattern for the prediction of transient performance of metal reheating," in *Proceeding of AISTech2014-The Iron & Steel Technology Conference and Exposition*, 2014, pp. 5–8.

[78] S. M. Acosta, A. L. Amoroso, Â. M. O. Sant'Anna, and O. C. Junior, "Predictive modeling in a steelmaking process using optimized relevance vector regression and support vector regression," *Annals of Operations Research*, vol. 316, no. 2, pp. 905–926, 2022.

[79] Q. Lu, S. Liu, W. Li, and X. Jin, "Combination of thermodynamic knowledge and multilayer feedforward neural networks for accurate prediction of ms temperature in steels," *Materials & Design*, vol. 192, p. 108696, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0264127520302306

[80] E. Fonda, A. Pandey, J. Schumacher, and K. R. Sreenivasan, "Deep learning in turbulent convection networks," *Proceedings of the National Academy of Sciences*, vol. 116, no. 18, pp. 8667–8672, 2019. [Online]. Available: https://www.pnas.org/doi/abs/10.1073/pnas.1900358116

[81] X. Yuan, S. Qi, Y. Wang, and H. Xia, "A dynamic cnn for nonlinear dynamic feature learning in soft sensor modeling of industrial process data," *Control Engineering Practice*, vol. 104, p. 104614, 2020.

[82] Y. Liang, S. Li, C. Yan, M. Li, and C. Jiang, "Explaining the black-box model: A survey of local interpretation methods for deep neural networks," *Neurocomputing*, vol. 419, pp. 168–182, 202).

[83] B. Zhang, F. Wang, and L. Qiu, "Multi-domain physics-informed neural networks for solving transient heat conduction problems in multilayer materials," *Journal of Applied Physics*, vol. 133, no. 24, p. 245103, 06 2023.

[84] J. Wang, G. Li, L. Wei, Z. Yi, and X. Wang, "Investigation on thermal characteristics of walking reheating furnace for beam blank," *International Communications in Heat and Mass Transfer*, vol. 128, p. 105646, 2021.

[85] Z. Ahmed, S. Lecompte, T. de Raad, and M. De Paepe, "Steady state model of a reheating furnace for determining slab boundary conditions," *Energy Procedia*, vol. 158, pp. 5844–5849, 2019.

[86] K. Liu, K. Li, Q. Peng, and C. Zhang, "A brief review on key technologies in the battery management system of electric vehicles," *Frontiers of mechanical engineering*, vol. 14, pp. 47–64, 2019.

[87] R. Rao, S. Vrudhula, and D. N. Rakhmatov, "Battery modeling for energy aware system design," *Computer*, vol. 36, no. 12, pp. 77–87, 2003.

[88] M. Doyle, T. F. Fuller, and J. Newman, "Modeling of galvanostatic charge and discharge of the lithium/polymer/insertion cell," *Journal of the Electrochemical society*, vol. 140, no. 6, p. 1526, 1993.

[89] S. Santhanagopalan, Q. Guo, P. Ramadass, and R. E. White, "Review of models for predicting the cycling performance of lithium ion batteries," *Journal of power sources*, vol. 156, no. 2, pp. 620–628, 2006.

[90] Y. Song, D. Liu, H. Liao, and Y. Peng, "A hybrid statistical data-driven method for on-line joint state estimation of lithium-ion batteries," *Applied Energy*, vol. 261, p. 114408, 2020.

[91] B. Y. Liaw, G. Nagasubramanian, R. G. Jungst, and D. H. Doughty, "Modeling of lithium ion cells—a simple equivalent-circuit model approach," *Solid state ionics*, vol. 175, no. 1-4, pp. 835–839, 2004.

[92] E. Locorotondo, L. Pugi, L. Berzi, M. Pierini, and G. Lutzemberger, "Online identification of thevenin equivalent circuit model parameters and estimation state of charge of lithium-ion batteries," in *2018 IEEE International Conference on Environment and Electrical Engineering and 2018 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I&CPS Europe)*, 2018, pp. 1–6.

[93] Q.-K. Wang, Y.-J. He, J.-N. Shen, Z.-F. Ma, and G.-B. Zhong, "A unified modeling framework for lithium-ion batteries: An artificial neural network based thermal coupled equivalent circuit model approach," *Energy*, vol. 138, pp. 118–132, 2017.

[94] Z. Deng, X. Lin, J. Cai, and X. Hu, "Battery health estimation with degradation pattern recognition and transfer learning," *Journal of Power Sources*, vol. 525, p. 231027, 2022.

[95] C. Zhang, Y. Zhu, G. Dong, and J. Wei, "Data-driven lithium-ion battery states estimation using neural networks and particle filtering," *International Journal of Energy Research*, vol. 43, no. 14, pp. 8230–8241, 2019.

[96] Y. Li, K. Li, X. Liu, Y. Wang, and L. Zhang, "Lithium-ion battery capacity estimation — a pruned convolutional neural network approach assisted with transfer learning," *Applied Energy*, vol. 285, p. 116410, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306261920317773

[97] D. Yang, X. Zhang, R. Pan, Y. Wang, and Z. Chen, "A novel gaussian process regression model for state-of-health estimation of lithium-ion battery using charging curve," *Journal of Power Sources*, vol. 384, pp. 387–395, 2018.

[98] P. Sun and Z. Wang, "Research of the relationship between li-ion battery charge performance and soh based on miga-gpr method," *Energy Procedia*, vol. 88, pp. 608–613, 2016.

[99] D. Lin, X. Zhang, L. Wang, and B. Zhao, "State of health estimation of lithium-ion batteries based on a novel indirect health indicator," *Energy Reports*, vol. 8, pp. 606–613, 2022.

[100] Z. He, M. Gao, G. Ma, Y. Liu, and S. Chen, "Online state-of-health estimation of lithium-ion batteries using dynamic bayesian networks," *Journal of Power Sources*, vol. 267, pp. 576–583, 2014.

[101] S. Schwunk, N. Armbruster, S. Straub, J. Kehl, and M. Vetter, "Particle filter for state of charge and state of health estimation for lithium–iron phosphate batteries," *Journal of Power Sources*, vol. 239, pp. 705–710, 2013.

[102] X. Tang, K. Liu, X. Wang, B. Liu, F. Gao, and W. D. Widanage, "Real-time aging trajectory prediction using a base model-oriented gradient-correction particle filter for lithium-ion batteries," *Journal of Power Sources*, vol. 440, p. 227118, 2019.

[103] X. Han, Z. Wang, and Z. Wei, "A novel approach for health management online-monitoring of lithium-ion batteries based on model-data fusion," *Applied Energy*, vol. 302, p. 117511, 2021.

[104] J. Zhang, P. Wang, Q. Gong, and Z. Cheng, "Soh estimation of lithium-ion batteries based on least squares support vector machine error compensation model," *Journal of Power Electronics*, vol. 21, pp. 1712–1723, 2021.

[105] Y. Li, K. Li, Y. Xie, J. Liu, C. Fu, and B. Liu, "Optimized charging of lithium-ion battery for electric vehicles: Adaptive multistage constant current–constant voltage charging strategy," *Renewable energy*, vol. 146, pp. 2688–2699, 2020.

[106] A. A.-H. Hussein and I. Batarseh, "A review of charging algorithms for nickel and lithium battery chargers," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 3, pp. 830–838, 2011.

[107] A. Kaleem, I. U. Khalil, S. Aslam, N. Ullah, S. Al Otaibi, and M. Algethami, "Feedback pid controller-based closed-loop fast charging of lithium-ion batteries using constant-temperature–constant-voltage method," *Electronics*, vol. 10, no. 22, p. 2872, 2021.

[108] A. Samanta, M. Sharma, and S. Williamson, "A supercapacitor and fuzzy-pid controller-based active charge balancing scheme for lithium-ion batteries," in *IECON 2022 – 48th Annual Conference of the IEEE Industrial Electronics Society*, 2022, pp. 1–6.

[109] Y. Parvini and A. Vahidi, "Maximizing charging efficiency of lithium-ion and lead-acid batteries using optimal control theory," in *2015 American Control Conference (ACC)*, 2015, pp. 317–322.

[110] P. Ramadass, B. Haran, P. M. Gomadam, R. White, and B. N. Popov, "Development of first principles capacity fade model for li-ion cells," *Journal of the Electrochemical Society*, vol. 151, no. 2, p. A196, 2004.

[111] G.-J. Chen, Y.-H. Liu, Y.-S. Cheng, and H.-Y. Pai, "A novel optimal charging algorithm for lithium-ion batteries based on model predictive control," *Energies*, vol. 14, no. 8, 2021. [Online]. Available: https://www.mdpi.com/1996-1073/14/8/2238

[112] S. Park, A. Pozzi, M. Whitmeyer, H. Perez, W. T. Joe, D. M. Raimondo, and S. Moura, "Reinforcement learning-based fast charging control strategy for li-ion batteries," in *2020 IEEE Conference on Control Technology and Applications (CCTA)*, 2020, pp. 100–107.

[113] T. Duraisamy and D. Kaliyaperumal, "Machine learning-based optimal cell balancing mechanism for electric vehicle battery management system," *IEEE Access*, vol. 9, pp. 132 846–132 861, 2021.

[114] F. Kazhamiaka, S. Keshav, and C. Rosenberg, "Adaptive battery control with neural networks," in *Proceedings of the Tenth ACM International Conference on Future Energy Systems*, ser. e-Energy '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 536–543.

[115] F.-J. Lin, M.-S. Huang, P.-Y. Yeh, H.-C. Tsai, and C.-H. Kuan, "Dsp-based probabilistic fuzzy neural network control for li-ion battery charger," *IEEE Transactions on Power Electronics*, vol. 27, no. 8, pp. 3782–3794, 2012.

[116] K. Li, J. Peng, G. Irwin, L. Piroddi, and W. Spinelli, "Estimation of nox emissions in thermal power plants using eng-genes neural networks," *IFAC Proceedings Volumes*, vol. 38, no. 1, pp. 115–120, 2005.

[117] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer, 2015, pp. 234–241.

[118] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[119] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.

[120] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[121] Y. Wang, W. Li, R. Fang, H. Zhu, and Q. Peng, "Capacities prediction and correlation analysis for lithium-ion battery-based energy storage system," *Control Engineering Practice*, vol. 125, p. 105224, 2022.

[122] K. M. Tan, T. S. Babu, V. K. Ramachandaramurthy, P. Kasinathan, S. G. Solanki, and S. K. Raveendran, "Empowering smart grid: A comprehensive review of energy storage technology and application with renewable energy integration," *Journal of Energy Storage*, vol. 39, p. 102591, 2021.

[123] A. Eftekhari, "Lithium batteries for electric vehicles: From economy to research strategy," *ACS Sustainable Chemistry & Engineering*, vol. 7, no. 6, pp. 5602–5613, 2019.

[124] K. Liu, Y. Gao, C. Zhu, K. Li, M. Fei, C. Peng, X. Zhang, and Q.-L. Han, "Electrochemical modeling and parameterization towards control-oriented management of lithium-ion batteries," *Control Engineering Practice*, vol. 124, p. 105176, 2022.

[125] L. Saw, A. A. O. Tay, and L. W. Zhang, "Thermal management of lithium-ion battery pack with liquid cooling," in *2015 31st Thermal Measurement, Modeling & Management Symposium (SEMI-THERM)*, 2015, pp. 298–302.

[126] A. A. Pesaran, "Battery thermal models for hybrid vehicle simulations," *Journal of Power Sources*, vol. 110, no. 2, pp. 377–382, 2002.

[127] J. Lin, X. Liu, S. Li, C. Zhang, and S. Yang, "A review on recent progress, challenges and perspective of battery thermal management system," *International Journal of Heat and Mass Transfer*, vol. 167, p. 120834, 2021.

[128] S. Zhu, C. He, N. Zhao, and J. Sha, "Data-driven analysis on thermal effects and temperature changes of lithium-ion battery," *Journal of Power Sources*, vol. 482, p. 228983, 2021.

[129] C. Zou, C. Manzie, and D. Nešić, "Model predictive control for lithium-ion battery optimal charging," *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 2, pp. 947–957, 2018.

[130] S. Qin and T. A. Badgwell, "A survey of industrial model predictive control technology," *Control Engineering Practice*, vol. 11, no. 7, pp. 733–764, 2003. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0967066102001867

[131] S. P. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[132] F. Guo, L. D. Couto, G. Mulder, K. Trad, G. Hu, O. Capron, and K. Haghverdi, "A systematic review of electrochemical model-based lithium-ion battery state estimation in battery management systems," *Journal of Energy Storage*, vol. 101, p. 113850, 2024.

[133] D. Bernardi, E. Pawlikowski, and J. Newman, "A general energy balance for battery systems," *Journal of the electrochemical society*, vol. 132, no. 1, p. 5, 1985.

[134] C. Zhang, K. Li, and J. Deng, "Real-time estimation of battery internal temperature based on a simplified thermoelectric model," *Journal of Power Sources*, vol. 302, pp. 146–154, 2016.

[135] K. Liu, K. Li, and C. Zhang, "Constrained generalized predictive control of battery charging process based on a coupled thermoelectric model," *Journal of Power Sources*, vol. 347, pp. 145–158, 2017.

[136] A. Aitio and D. A. Howey, "Predicting battery end of life from solar off-grid system field data using machine learning," *Joule*, vol. 5, no. 12, pp. 3204–3220, 2021.

[137] Z. Deng, X. Hu, X. Lin, Y. Che, L. Xu, and W. Guo, "Data-driven state of charge estimation for lithium-ion battery packs based on gaussian process regression," *Energy*, vol. 205, p. 118000, 2020.

[138] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[139] H. Wu, T. Hu, Y. Liu, H. Zhou, J. Wang, and M. Long, "Timesnet: Temporal 2d-variation modeling for general time series analysis," *arXiv preprint arXiv:2210.02186*, 2022.

[140] W. Huang and J. A. Abu Qahouq, "Energy sharing control scheme for state-of-charge balancing of distributed battery energy storage system," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 5, pp. 2764–2776, 2015.

[141] C. Zhang, Y. Jiang, J. Jiang, G. Cheng, W. Diao, and W. Zhang, "Study on battery pack consistency evolutions and equilibrium diagnosis for serial- connected lithium-ion batteries," *Applied Energy*, vol. 207, pp. 510–519, 2017, transformative Innovations for a Sustainable Future – Part II.

[142] X. Na, H. Kang, T. Wang, and Y. Wang, "Reverse layered air flow for li-ion battery thermal management," *Applied Thermal Engineering*, vol. 143, pp. 257–262, 2018.

[143] S.-C. Choi, J.-Y. Jeon, T.-j. Yeo, Y.-J. Kim, D.-Y. Kim, and C.-Y. Won, "State-of-charge balancing control of a battery power module for a modularized battery for electric vehicle," *Journal of Electrical Engineering and Technology*, vol. 11, pp. 629–638, 05 2016.

[144] M. Zeraati, M. E. Hamedani Golshan, and J. M. Guerrero, "Distributed control of battery energy storage systems for voltage regulation in distribution networks with high pv penetration," *IEEE Transactions on Smart Grid*, vol. 9, no. 4, pp. 3582–3593, 2018.

[145] J. Gazzarri, "Battery modeling," 2025, mATLAB Central File Exchange. Retrieved January 9, 2025.

[146] F. Zheng, Y. Xing, J. Jiang, B. Sun, J. Kim, and M. Pecht, "Influence of different open circuit voltage tests on state of charge online estimation for lithium-ion batteries," *Applied Energy*, vol. 183, pp. 513–525, 2016.

[147] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[148] F. A. Silva, "Modern electric, hybrid electric, and fuel cell vehicles, third edition [book news]," *IEEE Industrial Electronics Magazine*, vol. 12, no. 4, pp. 46–48, 2018.

[149] L. Lu, X. Han, J. Li, J. Hua, and M. Ouyang, "A review on the key issues for lithium-ion battery management in electric vehicles," *Journal of Power Sources*, vol. 226, pp. 272–288, 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378775312016163

[150] H. Schafer, E. Santana, A. Haden, and R. Biasini, "A commute in data: The comma2k19 dataset," 2018.

APPENDIX A: ADI METHOD DERIVATION AND CALCULATION FOR 2D TRANSIENT HEAT CONDUCTION

This appendix provides a complete mathematical derivation of the Alternating Direction Implicit (ADI) method as applied to the 2D transient heat conduction problem formulated in Chapter 3. The objective is to make the computational procedure fully explicit: from the governing PDE and its finite-difference discretisation, through the construction of the two half-step tridiagonal linear systems, to the matrix solution algorithm. All notation is consistent with Chapter 3.

A.1 Governing Equation and Spatial Discretisation

A.1 Governing Equation and Spatial Discretisation

The 2D transient heat conduction governing equation for an interior node $(x, y) \in D_I$ is (equation 3.1 of Chapter 3):

$$\frac{\partial T}{\partial t} = \frac{k(T)}{\rho\, C_p(T)} \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right). \tag{A.1}$$

The computational domain is discretised on a uniform square mesh of spacing $d = \Delta x = \Delta y$. Node (i, j) carries temperature $T_t(i, j)$ at time step t, where $i = 1, \ldots, N_x$ and $j = 1, \ldots, N_y$ index the grid in the x- and y-directions, respectively. A full time step of size Δt is split into two successive half-steps of size $\Delta t/2$.

The key dimensionless parameter governing the calculation is the *Fourier number*

(equation 3.14 of Chapter 3):

$$Fo = \frac{k(T)\,\Delta t}{\rho\, C_p(T)\, d^2}. \tag{A.2}$$

Note that Fo is defined here with respect to the *full* time step Δt; each half-step therefore corresponds to $Fo/2$. This convention is carried consistently through all subsequent derivations.

The second-order central-difference approximations for the spatial derivatives at node (i, j) are:

$$\left.\frac{\partial^2 T}{\partial x^2}\right|_{(i,j)} = \frac{T(i+1,j) - 2T(i,j) + T(i-1,j)}{d^2} + \mathcal{O}(d^2), \tag{A.3}$$

$$\left.\frac{\partial^2 T}{\partial y^2}\right|_{(i,j)} = \frac{T(i,j+1) - 2T(i,j) + T(i,j-1)}{d^2} + \mathcal{O}(d^2). \tag{A.4}$$

A.2 Energy-Balance Derivation of the ADI Finite-Difference Equations

The ADI method avoids the stability restriction of the fully explicit scheme by treating one spatial direction implicitly and the other explicitly in each half-step, alternating the implicit direction between the two half-steps. The derivation below follows the energy-balance approach for an interior node (i, j) with unit depth; the cross-sectional face area is $A = d \times 1 = d$. Thermal conductivity k, density ρ, and specific heat C_p are treated as uniform within each step for notational clarity (the temperature-dependent case in Chapter 3 simply updates Fo locally at each node).

A.2.1 First Half-Step (x-direction implicit, y-direction explicit)

The temperature advances from $T_t(i, j)$ at time t to an intermediate value $T_{t+\frac{1}{2}}(i, j)$, written $T'(i, j)$ for compactness. The net heat conducted into node (i, j) during the interval

$\Delta t/2$ equals the change in stored energy:

$$\underbrace{k\big[T'(i-1,j)-T'(i,j)\big]+k\big[T'(i+1,j)-T'(i,j)\big]}_{\text{implicit (x-direction): unknowns at } t+\frac{1}{2}}$$
$$+\underbrace{k\big[T_t(i,j-1)-T_t(i,j)\big]+k\big[T_t(i,j+1)-T_t(i,j)\big]}_{\text{explicit (y-direction): known values at } t} = \rho\, d^2\, C_p\, \frac{T'(i,j)-T_t(i,j)}{\Delta t/2}. \tag{A.5}$$

Dividing through by k and noting that $\rho\, d^2\, C_p/k = \Delta t/Fo$ (from equation (A.2)), the right-hand side becomes $(2/Fo)\big[T'(i,j)-T_t(i,j)\big]$. Collecting unknown T' terms on the left and known T_t terms on the right:

$$T'(i-1,j)-\left(\frac{2}{Fo}+2\right)T'(i,j)+T'(i+1,j) = -T_t(i,j-1)-T_t(i,j+1)+\left(2-\frac{2}{Fo}\right)T_t(i,j). \tag{A.6}$$

First half-step system (fixed row j, unknowns $T'(i,j)$, $i=1,\ldots,N_x$):

$$T'(i-1,j)-\left(\frac{2}{Fo}+2\right)T'(i,j)+T'(i+1,j) = -T_t(i,j-1)-T_t(i,j+1)+\left(2-\frac{2}{Fo}\right)T_t(i,j). \tag{A.7}$$

The unknowns $\{T'(i,j)\}$ on the left-hand side correspond to a single j-row (constant j, varying i); all right-hand side quantities involve only the known temperatures $T_t(\cdot,\cdot)$ at the old time level and are therefore pre-computable before the tridiagonal solve.

A.2.2 Second Half-Step (y-direction implicit, x-direction explicit)

The second half-step advances $T'(i,j)$ to the final temperature $T_{t+1}(i,j)$, written $T''(i,j)$. The roles of the two spatial directions are reversed:

$$\underbrace{k\big[T'(i-1,j)-T'(i,j)\big]+k\big[T'(i+1,j)-T'(i,j)\big]}_{\text{explicit (x-direction): known values at } t+\frac{1}{2}}$$
$$+\underbrace{k\big[T''(i,j-1)-T''(i,j)\big]+k\big[T''(i,j+1)-T''(i,j)\big]}_{\text{implicit (y-direction): unknowns at } t+1} = \rho\, d^2\, C_p\, \frac{T''(i,j)-T'(i,j)}{\Delta t/2}. \tag{A.8}$$

After the same algebraic reduction (with T' now playing the role of the known field):

Second half-step system (fixed column i, unknowns $T''(i,j)$, $j=1,\ldots,N_y$):

$$T''(i,j-1)-\left(\frac{2}{Fo}+2\right)T''(i,j)+T''(i,j+1) = -T'(i-1,j)-T'(i+1,j)+\left(2-\frac{2}{Fo}\right)T'(i,j). \tag{A.9}$$

The unknowns $\{T''(i,j)\}$ form a single i-column (constant i, varying j); the right-hand side contains only the intermediate values $T'(\cdot,\cdot)$ from the first half-step.

A.3 Tridiagonal Matrix Form and Coefficient Identification

For each fixed row j in the first half-step, equation (A.7) constitutes a system of N_x coupled linear equations in the N_x unknowns $\{T'(1,j),\dots,T'(N_x,j)\}$. Defining the scalar diagonal coefficient

$$a \triangleq -\left(\frac{1}{Fo}+2\right), \tag{A.10}$$

and off-diagonal coefficient $b \triangleq 1$, the system for row j takes the standard tridiagonal form:

$$\begin{bmatrix} a & b & & & \\ b & a & b & & \\ & \ddots & \ddots & \ddots & \\ & & b & a & b \\ & & & b & a \end{bmatrix} \begin{bmatrix} T'(1,j) \\ T'(2,j) \\ \vdots \\ T'(N_x-1,j) \\ T'(N_x,j) \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_{N_x-1} \\ r_{N_x} \end{bmatrix}, \tag{A.11}$$

where the right-hand side entries are

$$r_i = -T_t(i,j-1) - T_t(i,j+1) + \left(\frac{2}{Fo}-2\right)T_t(i,j) + (\text{boundary correction terms}), \tag{A.12}$$

with the boundary correction terms arising from known boundary temperatures when $i=1$ or $i=N_x$ (see Section A.4 below).

An identical tridiagonal structure applies for each column i in the second half-step (equation (A.9)), with the same diagonal and off-diagonal coefficients and the right-hand side constructed from the intermediate values $T'(\cdot,\cdot)$.

A.4 Boundary Condition Incorporation

Chapter 3 imposes a general Robin (mixed) boundary condition of the form (equation 3.3):

$$-k(T)\frac{\partial T}{\partial \vec{n}} = h_c(T-T_A) + \sigma e(T^4 - T_A^4), \tag{A.13}$$

where $\vec{n}$ is the outward unit normal at the surface. In the ADI framework, boundary values are either prescribed (Dirichlet) or updated consistently with the temperature at the previous sub-step.

For the simplified Dirichlet case (fixed surface temperature T_s), all boundary nodes carry the fixed value T_s and are excluded from the tridiagonal solve. Their contribution enters only as constant corrections to the right-hand sides r_1 and r_{N_x} of the interior systems:

$$r_1 \leftarrow r_1 - b\,T_s \quad \text{(correction for left boundary node)}, \tag{A.14a}$$

$$r_{N_x} \leftarrow r_{N_x} - b\,T_s \quad \text{(correction for right boundary node)}. \tag{A.14b}$$

An analogous correction applies at the top and bottom boundaries during the second half-step.

For the general Robin boundary in equation (A.13), the boundary-node energy balance at the surface (e.g., left face $i = 0$, any j) yields the one-sided finite difference expression:

$$-k\,\frac{T(1,j) - T(0,j)}{d} = h_c\big(T(0,j) - T_A\big) + \sigma e\,\big(T(0,j)^4 - T_A^4\big), \tag{A.15}$$

from which $T(0,j)$ can be determined given $T(1,j)$ and the ambient conditions, then substituted as a Dirichlet value into equation (A.11).

A.5 Complete ADI Algorithm for One Time Step

Collecting the above, the complete procedure for advancing the temperature field from $T_t(i,j)$ to $T_{t+1}(i,j)$ by one full time step Δt is presented as Algorithm 3.

Algorithm 3 ADI solution procedure for 2D heat conduction

Input: T_t, material properties $k(T)$, ρ, $C_p(T)$, time step Δt, mesh spacing d
Output: T_{t+1}

Step 1: Compute the Fourier number:
  $Fo \leftarrow k(T_t)\,\Delta t \,/\, \left(\rho\, C_p(T_t)\, d^2\right)$
  $a \leftarrow -(2/Fo + 2), \quad b \leftarrow 1$

Step 2: First half-step (x-direction implicit):
for $j = 1$ **to** N_y **do**
    Assemble the right-hand side vector $\{r_i\}$ via Eq. (A.12) using T_t
    Apply boundary corrections (Eqs. (A.14a)–(A.14b))
    Solve tridiagonal system (Eq. (A.11)) via the Thomas algorithm
    Store intermediate field $\{T'(i,j)\}_{i=1}^{N_x}$
end for

Step 3: Second half-step (y-direction implicit):
for $i = 1$ **to** N_x **do**
    Assemble the right-hand side vector $\{r_j\}$ via Eq. (A.9) using T'
    Apply boundary corrections (Eqs. (A.14a)–(A.14b), adapted for y-direction)
    Solve tridiagonal system via the Thomas algorithm
    Store updated field $\{T''(i,j)\}_{j=1}^{N_y}$
end for
Set $T_{t+1}(i,j) \leftarrow T''(i,j)$ for all interior nodes

Step 4: Update boundary values:
Compute boundary temperatures via Eq. (A.15) (Robin)
return T_{t+1}

APPENDIX B THEORETICAL ANALYSIS OF THE CONVEX MAPPING CONSTRAINT IN THE SELF-LEARNING MPC FRAMEWORK

This appendix provides rigorous theoretical justification for the convex mapping constraint and its integration into the self-learning MPC optimisation problem formulated in Chapter 4. Two key properties are established: (i)recursive feasibility, the augmented optimisation problem remains feasible at every time step and iteration once it is feasible at initialisation; and (ii)asymptotic convergence, the augmented cost $J_k(\mathbf{x}_k(t))$ is strictly decreasing along the time axis, guaranteeing convergence of the tracking error and the charging current trajectory.

Throughout this appendix, the notation of Chapter 4 is used. The prediction model is the linearised battery thermal sub-model (equation 4.2), the augmented optimisation problem is equation 4.20, and the convex mapping constraint on the terminal predicted state is equation 4.15. For completeness, the key definitions are recalled first.

B.1 Preliminaries and Notation Recap

Prediction model. At time step t of iteration k, the predicted state $\hat{\mathbf{x}}_k(j \mid t)$ evolves according to

$$\hat{\mathbf{x}}_k(j+1 \mid t) = A\,\hat{\mathbf{x}}_k(j \mid t) + B\,u_k(j \mid t) + C(t+j), \qquad j = 0,\dots,N_p-1, \tag{B.1}$$

with $\hat{\mathbf{x}}_k(0 \mid t) = \mathbf{x}_k(t)$ and $\mathbf{x}_k(t) \in \mathbb{R}^n$, $u_k(t) \in \mathbb{R}^m$.

Admissible constraint set. The compact constraint set $\mathscr{F}$ on state and input is defined by the box constraints

$$\mathscr{F} \triangleq \{(\mathbf{x}, u) \mid \Gamma^u u \le \Lambda_k^u(t), \quad \Gamma^x \mathbf{x} \le \Lambda_k^x(t)\}, \tag{B.2}$$

where Γ^u, Γ^x, $\Lambda_k^u(t)$, $\Lambda_k^x(t)$ are as defined in equations 4.6 and 4.5 of Chapter 4.

Historical state dataset. After completing the $(k-1)$-th iteration, the historical state dataset and memory function dataset are

$$\mathbf{X}_{k-1} = \{X_{k-1}(0), X_{k-1}(1), \ldots, X_{k-1}(N)\}, \tag{B.3}$$

$$\mathbf{M}_{k-1} = \{M_{k-1}(0), M_{k-1}(1), \ldots, M_{k-1}(N)\}. \tag{B.4}$$

Convex mapping constraint (terminal state). The terminal predicted state is constrained to lie within the convex hull of a sliding window of historical states centred at $t+N_p$:

$$\hat{\mathbf{x}}_k(N_p \mid t) = \sum_{j=-d}^{d} \theta_{k-1,j} X_{k-1}(t+N_p+j), \tag{B.5}$$

with non-negative convex combination coefficients satisfying

$$\theta_{k-1,j} \ge 0, \qquad \sum_{j=-d}^{d} \theta_{k-1,j} = 1. \tag{B.6}$$

Augmented optimisation problem. The complete self-learning MPC problem solved at each (k,t) is

$$\begin{aligned} \min_{\{\theta_{k-1,j}\},\, \mathbf{u}_k(t)} \; & J_k(t) + \sum_{j=-d}^{d} \theta_{k-1,j} M_{k-1}(t+N_p+j) \\ \text{s.t.} \quad & \text{(B.1), (B.2), (B.5), (B.6),} \end{aligned} \tag{B.7}$$

where $J_k(t)$ is the standard quadratic MPC cost defined in equation 5.22, and the optimal cost is denoted $J_k(\mathbf{x}_k(t))$.

B.2 Proposition 1 (Recursive Feasibility)

Proposition 1. *Consider the battery thermal sub-model* (B.1) *subject to the constraints* (B.2)*, controlled by the self-learning MPC algorithm of Chapter 4*

(Algorithm 1). If the standard MPC problem (i.e., problem (B.7) *without the convex mapping constraint* (B.5)*) is feasible at every time step t of the first iteration $k = 1$, then the augmented problem* (B.7) *is recursively feasible for all $t \in \{0, 1, \ldots, N\}$ and all $k \geq 1$.*

Proof. We proceed by mathematical induction over the iteration index k and, within each iteration, over the time index t.

Base case ($k = 1$). For the first iteration, the convex mapping constraint (B.5) is initialised using the feasible state trajectory obtained by solving the standard MPC problem with the initial CC-CV charging profile. Denote the resulting feasible state and input sequences as

$$\Delta\mathbf{X}_1(0) = \{\hat{\mathbf{x}}_1(0 \mid 0), \ldots, \hat{\mathbf{x}}_1(N_p \mid 0)\}, \quad \mathbf{u}_1(0) = \{u_1(0 \mid 0), \ldots, u_1(N_p - 1 \mid 0)\}.$$

These sequences satisfy all constraints in $\mathscr{F}$. Because the identical initial condition $\mathbf{x}_k(0) = \mathbf{x}_0$ holds for all k, the same sequences constitute feasible candidates for the initial time step $t = 0$ of any subsequent iteration. The historical dataset $\mathbf{X}_0$ is thus well-defined, and the convex hull constraint (B.5) is satisfiable at $t = 0$, $k = 1$.

Inductive step. Suppose that at some time t of iteration $k \geq 2$, the augmented problem (B.7) has a feasible solution, denoted $\mathbf{u}_k^*(t)$ with optimal convex coefficients $\{\theta_{k-1,j}^*\}$. The corresponding terminal predicted state satisfies

$$\hat{\mathbf{x}}_k^*(N_p \mid t) = \sum_{j=-d}^{d} \theta_{k-1,j}^* X_{k-1}(t + N_p + j). \tag{B.8}$$

We construct a candidate feasible solution at time $t + 1$ as follows. Define the shifted input sequence

$$\mathbf{u}_k^\bullet(\cdot \mid t+1) = \left[u_k^{*\top}(1 \mid t),\ \ldots,\ u_k^{*\top}(N_p - 1 \mid t),\ u_k^{\bullet\top}(N_p \mid t+1)\right]^\top, \tag{B.9}$$

where the appended terminal input is chosen as the convex combination of historical control inputs:

$$u_k^\bullet(N_p \mid t+1) = \sum_{j=-d}^{d} \theta_{k-1,j}^* u_{k-1}(t + N_p + j). \tag{B.10}$$

Since the historical inputs $u_{k-1}(\cdot)$ all satisfy the constraints $\mathscr{F}$, and the coefficients $\{\theta^*_{k-1,j}\}$ are non-negative and sum to one, the convex ombination $u^\bullet_k(N_p \mid t+1)$ also lies within $\mathscr{F}$ by convexity of the constraint set.

The corresponding terminal predicted state at time $t+1$ is

$$\hat{\mathbf{x}}^\bullet_k(N_p \mid t+1) = A\,\hat{\mathbf{x}}^*_k(N_p \mid t) + B\,u^\bullet_k(N_p \mid t+1) + C(t+N_p). \tag{B.11}$$

Substituting equation (B.8) and using the linearity of A and B:

$$\begin{aligned}\hat{\mathbf{x}}^\bullet_k(N_p \mid t+1) &= A\sum_{j=-d}^{d}\theta^*_{k-1,j}X_{k-1}(t+N_p+j) + B\sum_{j=-d}^{d}\theta^*_{k-1,j}u_{k-1}(t+N_p+j) + C(t+N_p)\\ &= \sum_{j=-d}^{d}\theta^*_{k-1,j}\big[A\,X_{k-1}(t+N_p+j) + B\,u_{k-1}(t+N_p+j) + C(t+N_p)\big]\\ &= \sum_{j=-d}^{d}\theta^*_{k-1,j}X_{k-1}(t+N_p+j+1).\end{aligned} \tag{B.12}$$

The last equality follows because $X_{k-1}(t+N_p+j+1)$ is precisely the one-step-ahead evolution of $X_{k-1}(t+N_p+j)$ under the historical dynamics. Therefore, $\hat{\mathbf{x}}^\bullet_k(N_p \mid t+1)$ is a convex combination of $\{X_{k-1}(t+N_p+j+1)\}_{j=-d}^{d}$, which lies within the convex hull of the window of historical states centred at $t+1+N_p$. The same coefficients $\{\theta^*_{k-1,j}\}$ thus satisfy the convex mapping constraint (B.5) at time $t+1$, and the candidate sequence $\mathbf{u}^\bullet_k(\cdot \mid t+1)$ is feasible. By induction, the augmented problem (B.7) is feasible for all t and all $k \geq 1$. □

> **Remark B.1.** Proposition 1 reveals a structural advantage of the convex mapping constraint over unconstrained MPC: because the feasible candidate at time $t+1$ is constructed explicitly from the historical trajectory of the previous iteration, feasibility at $t+1$ is guaranteed without requiring any additional assumption on the system matrices or the weighting matrices Q and H. The only requirement is that the initialisation (CC-CV profile) provides a feasible starting point for the standard MPC problem at $k = 1$.

B.3 Proposition 2 (Asymptotic Convergence)

Proposition 2. *Under the conditions of Proposition 1, suppose there exists a stabilising feedback gain K such that $(A+BK)$ is Schur stable and $u_k(t) = K\mathbf{x}_0$ satisfies the input constraint $\mathscr{F}$ for all k. Then the augmented cost $J_k(\mathbf{x}_k(t))$ is strictly decreasing along the time axis, and the tracking error and charging current trajectory are asymptotically convergent.*

Proof. The proof proceeds in three steps: (i) a key inequality relating the value function to the memory function via the Bellman equation; (ii) a bound on the augmented cost at successive time steps; (iii) a strict decrease argument establishing convergence.

Step 1: Bellman-type inequality for the memory function.

Recall from Chapter 4 that the memory function $M_k(t)$ encodes the minimum achievable cumulative cost-to-go from time step t, computed recursively backward from the terminal step:

$$M_k(N) = L_k\big(\mathbf{x}_k(N+1), \delta u_k(N)\big), \tag{B.13}$$

$$M_k(t) = L_k\big(\mathbf{x}_k(t+1), \delta u_k(t)\big) + M_k(t+1), \quad t < N. \tag{B.14}$$

For any convex combination of historical states $\sum_{j=-d}^{d} \theta_{k-1,j} X_{k-1}(t+j) \in \operatorname{conv}\mathbf{X}_{k-1}(t)$, the convexity of the stage cost $L_k(\cdot,\cdot)$ (which is quadratic and hence convex) and Jensen's inequality yield

$$\sum_{j=-d}^{d} \theta_{k-1,j} M_{k-1}(t+j) \geq M_{k-1}\left(\sum_{j=-d}^{d} \theta_{k-1,j} X_{k-1}(t+j) \right). \tag{B.15}$$

This shows that the convex hull of the memory function at time t provides an upper bound on the value achievable from within the convex hull of historical states. Using the Bellman decomposition, the memory function at any time t further satisfies

$$M_{k-1}(t) = L_{k-1}\big(\mathbf{x}_{k-1}(t+1), \delta u_{k-1}(t)\big) + M_{k-1}(t+1). \tag{B.16}$$

Combining this with the fact that the historical trajectory $\mathbf{x}_{k-1}(\cdot)$ satisfies all constraints in $\mathscr{F}$ and the prediction model (B.1), we obtain the key bound: for any $\mathbf{x}_k(t) \in \mathrm{conv}\mathbf{X}_{k-1}(t)$,

$$\sum_{j=-d}^{d} \theta_{k-1,j} M_{k-1}(t+N_p+j) \geq \max_{\Delta} \left[V_k\left(A\,\mathbf{x}_k(t) + B\,u_k(t) + \Delta(\mathbf{x}_k(t))\right) + L_k\left(\mathbf{x}_k(t+1), \delta u_k(t)\right)\right], \tag{B.17}$$

where $\Delta(\mathbf{x}_k(t))$ denotes the model-state discrepancy arising from the difference between the nominal prediction model and the actual system trajectory, and $V_k(\cdot)$ is the associated optimal value function.

Step 2: Bounding the augmented cost at time $t+1$.

At time t of iteration k, let the optimal solution of the augmented problem (B.7) be $\mathbf{u}_k^*(t)$ and $\{\theta_{k-1,j}^*\}$. The optimal augmented cost is

$$J_k\left(\mathbf{x}_k(t)\right) = J_k^*(t) + \sum_{j=-d}^{d} \theta_{k-1,j}^* M_{k-1}(t+N_p+j). \tag{B.18}$$

Expanding $J_k^*(t)$ using the Bellman equation:

$$J_k^*(t) = \max_{\Delta} \left[L_k\left(\hat{\mathbf{x}}_k^*(1 \mid t), \delta u_k^*(0 \mid t)\right) + J_k^*\left(\hat{\mathbf{x}}_k^*(1 \mid t)\right)\right]. \tag{B.19}$$

Substituting into the augmented cost and applying inequality (B.17) to bound the memory term gives

$$J_k\left(\mathbf{x}_k(t)\right) \geq L_k\left(\hat{\mathbf{x}}_k^*(1 \mid t), \delta u_k^*(0 \mid t)\right) + J_k\left(\hat{\mathbf{x}}_k^*(1 \mid t)\right). \tag{B.20}$$

By the optimality of problem (B.7) at time $t+1$, the actual augmented cost satisfies

$$J_k\left(\mathbf{x}_k(t+1)\right) \leq J_k\left(\hat{\mathbf{x}}_k^*(1 \mid t)\right). \tag{B.21}$$

Step 3: Strict decrease and convergence.

Combining inequalities (B.20) and (B.21):

$$J_k\left(\mathbf{x}_k(t+1)\right) - J_k\left(\mathbf{x}_k(t)\right) \leq -L_k\left(\hat{\mathbf{x}}_k^*(1 \mid t), \delta u_k^*(0 \mid t)\right) < 0, \tag{B.22}$$

where the strict inequality follows from the positive-definiteness of the weighting matrices $Q > 0$ and $H > 0$, which ensure $L_k(\mathbf{x}, \delta u) > 0$ for any non-zero $(\mathbf{x}, \delta u)$. This establishes that $J_k(\mathbf{x}_k(t))$ is a strictly decreasing function along the time axis t within each iteration.

Since $J_k(\mathbf{x}_k(t))$ is bounded below by zero and strictly decreasing, the sequence converges. The stage cost $L_k(\hat{\mathbf{x}}_k^*(1 \mid t), \delta u_k^*(0 \mid t)) \to 0$ as $t \to \infty$, which implies $\hat{\mathbf{x}}_k^*(1 \mid t) \to \mathbf{x}_r$ and $\delta u_k^*(0 \mid t) \to 0$, establishing asymptotic convergence of the tracking error to zero and stabilisation of the charging current trajectory. Furthermore, since the result holds uniformly across iterations k, the input trajectory along the iteration axis is also asymptotically convergent.

B.4 Corollary: Computational Stability of the Convex Mapping Constraint

Corollary B.1. *The incorporation of the convex mapping constraint B.5 reduces the feasible solution space of the MPC optimisation at each time step t relative to the unconstrained problem, and this reduction is monotonically non-decreasing as the historical dataset $\mathbf{X}_{k-1}$ shrinks in diameter across iterations. Consequently, the worst-case solve time of the quadratic programme* (B.7) *is bounded and non-increasing across iterations.*

Proof sketch. The augmented problem (B.7) is a convex quadratic programme (QP) because: (i) the objective is quadratic in $\mathbf{u}_k(t)$ (from $J_k(t)$) and linear in $\{\theta_{k-1,j}\}$ (from the memory term); and (ii) all constraints are affine. The convex mapping constraint (B.5) introduces $2d+1$ additional equality and inequality constraints relative to the standard MPC problem, but does not alter the convex structure of the feasible set $\mathscr{F}$.

The convex hull $\mathrm{conv}\mathbf{X}_{k-1}(t)$ is a bounded polytope whose diameter is determined by the spread of historical state trajectories at time $t+N_p$. As iterations progress and the charging current converges toward a stable profile, the spread of $\mathbf{X}_{k-1}(t)$ diminishes—i.e., successive iterates cluster more tightly—and the polytope shrinks. A smaller polytope means a more tightly constrained terminal state, which in turn restricts the feasible set of the prediction sequence $\{\hat{\mathbf{x}}_k(j \mid t)\}_{j=0}^{N_p}$ and reduces the number of active constraints evaluated at the boundary. Standard results in parametric QP theory (see, e.g., Boyd

and Vandenberghe [131]) then imply that the worst-case computational burden is non-increasing as the feasible region contracts. This is consistent with the empirical observation in Chapter 4 that the standard deviation of solve time is reduced by approximately 80% compared to standard MPC without the convex mapping constraint.