

Neural Network-Based Thermal Modeling of Small Satellites

A First-Principles Approach

Ullas Bhat

Neural Network-Based Thermal Modeling of Small Satellites

A First-Principles Approach

by

Ullas Bhat

To obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Thursday, July 13th, 2023

Student number: 5266882
Project duration: 08/2022 - 07/2023
Thesis committee: Dr. A. Menicucci Associate Professor, TU Delft
Dr. S. Speretta Assistant Professor, TU Delft
Dr. I. Uriol Balbin Lecturer, TU Delft

Cover: Rendering of Delfi-PQ in orbit, taken from MLABSPACE [<http://mlabspace.com/>]



Preface

The past couple years of my life, in pursuit of my M.Sc. degree has been a tumultuous yet rewarding journey, largely influenced by the onset of a global pandemic. The isolation and the uncertainty it fostered initially left me wondering about my progression, both as a student and a prospective young professional. This thesis project has been instrumental in helping me find my niche, allowing me to rediscover my fascination with space.

First and foremost, I would like to express my deepest gratitude to my supervisor, Dr. Stefano Speretta. Having followed a Master's course you took, I approached you to possibly explore the use of machine learning for space applications. The ensuing dive into this vast subject, resulting in this work, has been a truly rewarding experience. The challenging queries you posed and insightful feedback you provided have been instrumental in guiding and organizing the chaotic way in which my mind, fortunately or unfortunately, works. It has been a truly challenging, but fulfilling endeavor.

I would also like to thank the Delfi-C³, Delfi-PQ and FUNcube-1 teams for providing the telemetry used for this study.

To all my friends, old and new, and too many to name in this short space, thank you for coming into my life and making the past couple of years a little less daunting. Thanks for pushing me to be the best I can be. To the trio, Aaron, Brian and Bhanu, how far we have come! From spending countless waking hours building RC planes in the humid workshop to, well, here! The impromptu biking adventures, the late-night deep-dive into the most obscure corners of the internet, and all the other shenanigans in between, I will cherish these memories forever. You are what define *gezelligheid* for me. Thank you for making a foreign land feel like home. To Elena, I truly would not have finished the last leg of this journey without all concentrated hours of work interspersed with the numerous coffee breaks and chilling in the Sun. Thank you for helping me see the proverbial light at the end of the tunnel.

Last, but certainly not the least, I would like to thank my parents, Udayachala and Suma. Thank you for your unwavering support and encouragement, and for always believing in me. You nurtured my dreams, allowed me to chart my course, stumble, and learn from my mistakes while providing a safety net for me to fall back on. Your sacrifices have paved my way, and I owe my success to you. This work is dedicated to you, and I hope to make you proud.

*Ullas Bhat
Delft, June 2023*

Table of contents

Preface	i
List of Figures	iv
List of Tables	vii
List of Symbols and Abbreviations	ix
Abstract	x
1 Introduction	1
1.1 Overview of thesis	4
2 Background	5
2.1 Recurrent neural networks	5
2.2 First-principles neural networks	8
2.3 Satellites and Telemetry Data	11
3 Research questions	14
4 Methodology	17
4.1 Data pre-processing	17
4.2 Data-driven models	18
4.3 First-principles based models	20
4.4 Computational framework	22
5 Orbit Propagation	23
5.1 Two-line elements	23
5.2 Orbit propagators	25
5.2.1 Two-body model	26
5.2.2 Simplified General Perturbations 4 model	27
6 Data preprocessing	31
6.1 Orbit overlaying	31
6.2 Removing outliers	33
6.2.1 Data thresholding	33
6.2.2 Time correction	34
6.2.3 Constant section removal for FUNcube-1	36
6.2.4 Hampel filter	37
6.3 Training and testing split	39
6.4 Data normalization	39
6.5 Data selection	40

7	Data-driven models	42
7.1	FUNcube-1 temperature prediction	42
7.1.1	Temperature-based models	42
7.1.2	Illumination-based models	45
7.2	Delfi-PQ temperature prediction	52
8	First-principles models	55
8.1	Curve-fit model	55
8.1.1	Validation of curve-fit model using FUNcube-1 data	56
8.1.2	Hybrid networks for Delfi-PQ using curve-fit model	56
8.2	Thermal network model	61
8.2.1	Thermal network model for Delfi-PQ	61
8.2.2	Assumptions for the thermal simulation	63
8.2.3	Physical property estimation	64
8.2.4	Residual networks	73
8.2.5	Batched optimization	73
9	Conclusion	77
9.1	Answers to research questions	77
9.2	Future work recommendations	79
	References	81
A	Telemetry fields	85
B	Delfi-PQ thermal network	88
C	Code	90

List of Figures

2.1	LSTM cell structure	6
2.2	Schematic of a series hybrid model	9
2.3	Schematic of a parallel hybrid model	10
2.4	Delfi-C ³ [22]	12
2.5	FUNcube-1	13
5.1	Example of a TLE	23
5.2	Histogram of the time between Two-Line Elements (TLEs)	25
5.3	Histogram position errors of two-body model after a propagation duration of two days	27
5.4	Specific energy of Delfi-C ³ starting from 2017-07-04 05:54:27 to 2017-07-07 02:31:32 as predicted by the two-body model and calculated from TLEs	28
5.5	Histogram position errors of SGP-4 model after a propagation duration of two days	29
5.6	Specific energy of Delfi-C ³ starting from 2017-07-04 05:54:27.610 to 2017-07-07 02:31:32.086 as predicted by the SGP-4 model and calculated from TLEs	29
5.7	Effect of propagation step size on the predicted position error and computation time for SGP-4 model	30
6.1	Schematic of orbit overlaying (adapted from [53])	32
6.2	Delfi-C ³ TFSC temperatures before and after thresholding for telemetry data in 2012	34
6.3	Delfi-PQ temperatures before and after thresholding	35
6.4	Delfi-C ³ before and after field of view filtering for TFSC Z+X+ telemetry data in 2012	36
6.5	FUNcube-1 raw chassis and panel temperatures	37
6.6	FUNcube-1 black chassis temperature before and after constant section removal	37
6.7	Hample filter applied to Delfi-PQ solar panel Yp temperatures for the period of 2022-01-18 to 2022-03-29	39
6.8	Mean TFSC temperature of Delfi-C ³ for 2012	40
7.1	Predicted and actual black panel temperature for FUNcube-1 (2016-02-04 20:52:00 to 2016-02-04 22:28:00) using a temperature-based LSTM model	44
7.2	7.2a, 7.2b, 7.2c: Predicted and actual black panel temperature for FUNcube-1 during anomaly (2016-02-04 07:56:00 to 2016-02-04 09:30:00) using temperature-based LSTM model 7.2d, 7.2e, 7.2f: Difference between actual and predicted black panel temperature for FUNcube-1 during anomaly	45
7.3	Predicted and actual black panel temperature for FUNcube-1 using an illumination-based LSTM model	46

7.4	FUNCube-1 black panel temperature prediction error statistics for different window sizes	46
7.5	Schematic of the combined illumination-based and temperature-based residual models	47
7.6	Predicted and actual black panel temperature for FUNCube-1 using illumination-based LSTM model with residual network	48
7.7	FUNCube-1 black panel temperature prediction error statistics for different residual input window sizes	48
7.8	7.8a, 7.8b, 7.8c: Predicted and actual black panel temperature for FUNCube-1 during anomaly (2016-02-04 07:56:00 to 2016-02-04 09:30:00) using an illumination-based LSTM model with an input window size of 96 time steps and temperature-based residual model 7.8a, 7.8b, 7.8c: Difference between actual and predicted black panel temperature by the base and combined models for FUNCube-1 during anomaly	49
7.9	Measured and predicted temperatures of all 8 temperature channels in FUNCube-1 telemetry data using an illumination-based LSTM model with an input window size of 96 time steps	49
7.10	Prediction error μ and σ for different residual input window sizes of the combined model for all 8 temperature channels in FUNCube-1 telemetry data	50
7.11	Measured and predicted temperatures of all 8 temperature channels in FUNCube-1 telemetry data using an illumination-based LSTM model with an input window size of 96 time steps and a residual model with an input window size of 1 timestep	51
7.12	Measured and predicted temperatures of all 8 temperature channels in FUNCube-1 telemetry data using an illumination-based LSTM model with an input window size of 96 time steps and a residual model with an input window size of 5 timesteps	51
7.13	Predicted and actual temperatures of all 8 temperature channels in FUNCube-1 telemetry data during anomaly (2016-02-04 07:56:00 to 2016-02-04 09:30:00) using an illumination-based LSTM model with an input window size of 96 time steps and a residual model with an input window size of 5 timesteps	52
7.14	Predicted and measured temperatures of Delfi-PQ using an illumination-based LSTM model with an input window size of 94 time steps	53
7.15	Kernel Density Estimation (KDE) of the prediction error on the test set of Delfi-PQ using an illumination-based LSTM model with an input window size of 94 time steps	54
7.16	Error statistics of the combined illumination base and residual models of various step sizes for Delfi-PQ	54
8.1	Curve-fit model prediction on FUNCube-1 test dataset	57
8.2	KDE of the prediction error of the illumination-based model and the curve-fit model on the FUNCube-1 test dataset.	57
8.3	Exponential curve-fit on Delfi-PQ training dataset	58
8.4	KDE of the prediction error of the illumination-based model and the curve-fit model on the Delfi-PQ test dataset	59

8.5	Hybrid model predictions for Delfi-PQ for different number of simulated orbits used for training.	59
8.6	Prediction error of the hybrid model for Delfi-PQ for different number of simulated orbits used for training.	60
8.7	Error statistics of the combined hybrid base and residual models of various training simulated orbits and input temperature input window sizes for Delfi-PQ	60
8.8	Star-point reduction	63
8.9	Simulated temperature of Xp solar panel for 680 minutes	65
8.10	Base thermal network simulation	65
8.11	KDE of errors between simulated temperatures and telemetry data	66
8.12	Effect of initial temperature on the thermal network simulation for solar panel Xp	66
8.13	Effect of perturbations on the thermal network simulation for solar panel Xp	67
8.14	Thermal network simulation for $f = 0.7$	69
8.15	Mean temperatures of the Delfi-PQ telemetry between 2022-01 and 2022-04	69
8.16	Simulated temperatures of the thermal network with optimized values of α and ϵ	71
8.17	Simulated temperatures of the thermal network with optimized values of α , ϵ and C_p	71
8.18	KDE of the prediction error of the thermal network model and the combined models	74
8.19	Prediction error statistics of the thermal network combined model for different input window sizes	74
8.20	Variation of optimized α and ϵ for batches of 45 days	75
8.21	Variation of Root Mean Square Error (RMSE) of the optimization results and the spread of the data for batches of 45 days	76
B.1	Internal stack of Delfi-PQ [44]	88
B.2	Location of nodes of the 13-node thermal network model [13]	89

List of Tables

2.1	Orbital characteristics of Delfi-C ³ (calculated using TLE generated on 2023-06-18 02:21:36)	11
2.2	Orbital characteristics of Delfi-PQ (calculated using TLE generated on 2023-06-18 02:21:36)	12
2.3	Orbital characteristics of Funcube-1 (calculated using TLE generated on 2023-06-18 02:21:36)	13
5.1	Description of the fields in the TLE	24
5.3	Time difference statistics between TLEs for Delfi-C ³ , Delfi-PQ, and FUNcube-1	25
7.1	LSTM model architecture	43
7.2	Prediction error in black panel temperature for FUNcube-1 during anomalous orbit (2016-02-04 07:56:00 to 2016-02-04 09:30:00) using temperature based LSTM model	44
7.3	Prediction error (all channels) for FUNcube-1 during anomalous orbit (2016-02-04 07:56:00 to 2016-02-04 09:30:00) using combined illumination-based model ($w = 96$) and residual models	50
7.4	Summary of FUNcube-1 models	52
8.1	Curve-fit model parameters for Delfi-PQ.	58
8.2	Thermal couplings (W/K) between nodes in the 13 node thermal network model for Delfi-PQ	62
8.3	Physical properties of nodes	62
8.4	Thermal couplings (W/K) between nodes in the reduced 8 node thermal network model for Delfi-PQ	63
8.5	Optimized values of α and ϵ for the panels	70
8.6	Optimized values of α , ϵ for the panels, and C_p for battery, and MCU	72
8.7	RMSE of the thermal network at different optimization stages	73
8.8	Summary of FUNcube-1 models	75

List of Abbreviations

AWSS Autonomous Wireless Sun Sensor

BEM Boundary Element Method

CFD Computational Fluid Dynamics

COTS Commercial Off-The-Shelf

EDL Entry-Descent-Landing

EWMA Exponentially Weighted Moving Average

GPU Graphical Processing Unit

KDE Kernel Density Estimation

LEO Low Earth Orbit

LSTM Long Short-Term Memory

MarCO Mars Cube One

MCU Microcontroller Unit

MLP Multi-Layer Perceptron

MSE Mean Squared Error

MSL Mars Science Laboratory

OBC On-Board Computer

ODE Ordinary differential equation

OOD Out-Of-Limits

RAP Radio Amateur Platform

ReLU Rectified Linear Unit

RMSE Root Mean Square Error

RNN Recurrent Neural Network

SGP4 Simplified General Perturbations 4

SMAP Soil Moisture Active Passive

SQP Sequential Quadratic Programming

TFSC Thin-Film Solar Cell

TLE Two-Line Element

Abstract

The use of small satellites, enabled by the standardization of the CubeSat specifications and miniaturization in electronics, has seen a rapid increase in the past decades. The low-cost and short development time of these satellites has made them an attractive option for both commercial and academic applications, making space exploration more accessible. However, these small satellites are prone to failures, leading to lost scientific potential. Mitigation of these failures forms the motivation for this thesis. Recent advances in neural networks have shown promise in the field of anomaly detection. The black-box nature of such models, however, makes it challenging to understand the reasoning behind their predictions.

Constraining the data-driven models with known physics can not only help us understand the reasoning behind their predictions, but also ensuring the model is consistent with the real-world behavior of the system. The work presented in this Master's thesis aims to demonstrate the advantages of such first-principles neural networks over purely data-driven models in thermal behavior modeling of small satellites. Baseline performance of data-driven Long Short-Term Memory (LSTM) networks is established using FUNCube-1 telemetry data, quantifying the temperature prediction accuracy of the models under ideal conditions. The limitations of these models, especially with sparse data, are then investigated, to highlight the need for more robust models.

First-principles models, based on a physics-informed curve-fit and simplified thermal network models, are then developed to constrain the data-driven model predictions. The first-principles models are shown to be more robust to sparse data, with the predictions on data not seen during training being more consistent with the real-world thermal behavior of the satellite. Methods to relate the first-principles model parameters to the physical properties of the satellite are also proposed and explored, to help extract the evolution of the thermal behavior of the satellite over time.

1

Introduction

The past twenty years have witnessed a remarkable increase in the capabilities and complexity of missions undertaken by small-scale satellites, a trend that began with the first CubeSat launch in 2000. By May 2023, the number of nanosatellites and CubeSats deployed into space has reached roughly 2300 and 2000, respectively, and are only expected to increase [32]. These figures demonstrate the growth and increased reliance on these smaller satellite systems in space exploration and scientific research.

The transformational shift in small satellite missions can be traced back to the standardization of CubeSat specifications in 1999 [23]. This standardization provided manufacturers and researchers with a platform for designing and developing CubeSats with greater certainty and efficiency. Moreover, it provided a clear definition and established guidelines for the dimensions, design, and deployment of these satellites, thereby facilitating their wider acceptance and use.

The 1990s also saw significant advances in the domains of micro-electronics and micro-control systems. These technologies have had a significant impact on the manufacturing process of CubeSats and other small satellites. Specifically, they have made it possible to miniaturize complex electronic systems into a small form factor while maintaining, or even improving, their performance. CubeSats can now be manufactured using readily available and relatively inexpensive Commercial Off-The-Shelf (COTS) components [21]. This miniaturization has been crucial to the success of CubeSats, allowing them to be adapted for complex tasks and missions.

The CubeSat standard introduced a unit of measurement called 1U, defined as a $10 \times 10 \times 10 \text{ cm}^3$ cube. Satellites can be assembled to align with these dimensions, resulting in a 1U CubeSats size. Moreover, these units can be combined to form larger satellite structures. To keep up with modern advancements, the CubeSat standard undergoes periodic revisions, the most recent of which was updated in February 2022 [39].

Historically, the primary applications of CubeSats were in educational endeavors, with over a 100 universities involved in CubeSat missions [9]. They have also been used as technology demonstrators, and in Earth observation missions [52]. Increasingly, CubeSats are being recognized for their potential in contributing to the broader scientific missions,

especially in the fields of astrophysics and planetary exploration missions. They have been tasked with diverse missions, from testing advanced propulsion systems [34] to studying planetary formation [14], demonstrating the impressive versatility of CubeSats. Their small dimensions and lower cost render them a compelling option for evaluating novel technologies or carrying out experiments in space that may otherwise prove prohibitively costly or risky with larger, more traditional spacecraft.

As of May 2023, 15 CubeSats have been sent on interplanetary missions [32], with NASA's InSight Mars lander mission in 2018 being the first. Two 6U CubeSats, named Mars Cube One (MarCO), were utilized to establish a communications link with the InSight lander during its descent into the Martian atmosphere [1]. This marked the first time CubeSats had been deployed in an interplanetary mission. The CubeSats enabled near real-time transmission of the InSight during its Entry-Descent-Landing (EDL) phase. MarCO-B was also the first CubeSat to capture an image of Mars. The success of this mission helped solidify the role of CubeSats in future interplanetary missions and opened up exciting new possibilities for their use in space exploration.

Despite the significant strides made in the field of CubeSat technology, these missions are not without their challenges. In particular, reliability issues have been a major hurdle, often impacting the success and effectiveness of these missions. In 2018, it was noted that about a quarter of all CubeSat missions suffered early loss of mission.

There are a multitude of approaches being explored to enhance CubeSat reliability and mitigate the risk of mission failure. One such strategy is the improvement of mission testing procedures [16]. By rigorously testing CubeSats under conditions that simulate the harsh environment of space, it is possible to identify potential points of failure before launch. This proactive approach allows for issues to be addressed in the design and construction stages, greatly enhancing the chance of mission success.

System redundancy is another valuable approach to improve CubeSat reliability [8]. This involves duplicating critical components or systems within the satellite. In the event that a primary system fails, the backup system can take over, thereby maintaining functionality and preventing mission failure. While redundancy can increase the complexity and weight of CubeSats, the benefits of enhanced reliability often outweigh these costs.

While the aforementioned strategies are crucial in enhancing the reliability of CubeSats during the design and development phase, anomaly detection methods can be used to identify deviations from expected behavior. Early detection of such irregularities is pivotal in enabling swift planning and implementation of corrective measures, thereby potentially preventing mission failure. In this way, anomaly detection methods complement the pre-launch reliability enhancements, offering a more comprehensive approach to ensuring mission success.

Most CubeSats maintain connectivity with ground stations, facilitating the collection of telemetry. This information is indispensable for assessing the current condition and performance of the CubeSat. However, for those CubeSats used for deep-space missions, the transmission of these telemetry streams can be affected by latency problems or disruptions in the downlink owing to the loss of direct line-of-sight. CubeSats missions developed by universities and other institutions with constrained budgets may face

difficulties in maintaining constant data links with their CubeSats due to limited access to ground stations.

This delay in receiving telemetry poses a significant challenge, as it can impose a lower limit for identifying and responding to anomalies occurring on board the CubeSat. With advancements in data transmission methods, the volume of data relayed by CubeSats has grown significantly. However, this increase in data transmission can further contribute to delays, particularly if the data needs to be manually reviewed to identify potential anomalies. This underscores the significance of automated data analysis and anomaly detection systems, which are capable of swiftly and precisely sifting through substantial volumes of data, thereby ensuring prompt identification of potential issues and mitigating the possibility of mission failure.

There are several anomaly detection methods that are used for space applications, such as a simple threshold applied to satellite parameters. Selecting this threshold might present challenges to mission designers, as all possible normal operating conditions must be accurately identified using on-ground testing and simulation data. This might limit the anomaly detection to only those that have been previously identified. Even when anomalies are detected, identifying the root cause of the anomaly can present a challenge. Successful identification of the cause of an anomaly can be used to address the issue using in-flight software updates, saving the mission, and providing data for future missions.

Satellite telemetry fundamentally takes the form of time series data. As outlined by Chandola, Banerjee, and Kumar [12], this kind of data, when considering anomalies, can be categorized into three distinct types:

1. **Point anomalies:** These are individual data points that exist in regions of the data space that are sparsely populated.
2. **Contextual anomalies:** These are data points that, while being outliers in comparison to their immediate, local values, do not necessarily fall within sparse regions of the data space. They are anomalies within a specific context, making their detection reliant on understanding the surrounding data.
3. **Collective anomalies:** These refer to sequences of data points that, taken together, deviate from expected behavior. Unlike point or contextual anomalies, collective anomalies aren't about singular data points, but rather about unusual collections or patterns of data in relation to the overall data set.

Machine learning techniques, in particular, various forms of Recurrent Neural Networks (RNNs), have emerged as powerful tools for anomaly detection. The inherent architecture of RNNs enables them to learn temporal dependencies in time series data, which makes them ideally suited for identifying contextual and collective anomalies.

However, the use of machine learning in anomaly detection isn't restricted to their role as classifiers to separate nominal data from anomalies. An equally significant use case lies in their potential for predicting future data points in the telemetry stream. The difference between the predicted and actual telemetry values can be used as part of a more comprehensive anomaly detection system to identify anomalies.

The use of machine learning for prediction problems, however, is not without challenges.

In particular, when the training data is sparse and does not adequately represent the full range of possible data points. Fully data-driven machine learning models are usually black-box models, which makes it difficult to interpret the results and understand the reasoning behind the predictions. The prediction of the models in data regions that are not well represented in the training data can be unreliable.

The main objective of this thesis, at a high level, is to explore the possibility of combining data-driven models with known physics to constrain the prediction space of the hybrid model to be more representative of the actual system behavior. Such hybrid models, also known as first-principles neural networks, have been used in the past to model various ground-based systems. This thesis aims to explore the use of such models for modelling the thermal behavior of small satellites. The work presented can be used to gain insights on not just “what” are the predictions of the model, but also “why” the model is making those predictions.

1.1. Overview of thesis

The thesis report is structured as follows: [Chapter 2](#) provides an overview of the state-of-the-art in anomaly detection methods and time series forecasting using machine learning. [Chapter 3](#) identifies the research gap and proposes the research questions tackled. [Chapter 4](#) describes the methodology adopted to answer the research questions at a high level, with the subsequent chapters providing more details. [Chapter 5](#) describes the orbit propagation techniques explored to generate the satellite trajectories. These trajectories are used to correlate the telemetry data with the satellite’s position in orbit. The orbit-overlaying technique used to transform the telemetry data from time domain to the satellites’ position in orbit, along with the preprocessing steps used to prepare the data for the data-driven and first-principles neural networks, are described in [Chapter 6](#). [Chapter 7](#) describes the performance of the various data-driven models explored in this study and demonstrates some of their limitations. [Chapter 8](#) describes the performance of the two first-principles models explored in this study, and compares their performance with the data-driven models, along with estimation of evolution of the parameters of the first-principles models. Finally, the main conclusions drawn from the results presented in this study and recommendations for future work are presented in [Chapter 9](#).

2

Background

Designing an effective anomaly detection system presents several intricate challenges. At its core, anomaly detection involves determining data regions that represent normal operations, with any deviating data points being flagged as anomalies. Crafting a data space that comprises all potential normal behaviors is a complex task, with the boundary between normal and anomalous often being fuzzy and dependent on the specific context.

Simple anomaly detection algorithms typically use static thresholds for parameters in the telemetry stream to discern anomalies. These Out-Of-Limits (OOL) methods categorize data points that fall outside the pre-set boundaries as anomalies [25]. While OOL methods have the advantage of simplicity and low computational cost, the process of setting the thresholds requires substantial input from mission designers. They must meticulously define the critical thresholds for all parameters, and despite these efforts, the approach may overlook subtle but potentially important parameter shifts.

Several data-driven anomaly detection techniques have been employed for space system telemetry. This includes nearest-neighbor approaches [27] and methods based on data clustering [18], [28]. While these methods, including OOL methods, are adept at detecting point anomalies, they struggle to identify contextual and collective anomalies due to the difficulty of understanding long-term dependencies in the telemetry data.

2.1. Recurrent neural networks

The detection of contextual and collective anomalies presents significant challenges due to the need for contextual understanding and the identification of broader patterns in the data. RNN is a specific type of neural network that can be used to solve this problem. Featuring a feedback loop, RNNs can incorporate temporal information, making them ideal for dealing with data that changes over time [35], [42]. They have been used effectively in time series forecasting, where past data points are used as inputs to predict future values.

However, RNNs are not without their limitations. A key issue is the *vanishing gradient problem*, first described by Hochreiter [24]. Neural network training often involves calculating error gradients at the output layer and backpropagating these gradients through the network to update the weights. During RNN training, the backpropagation step applies not only from the output layer to the input layer, but also over time, from the most recent data point to the earliest. This can result in the network struggling to learn long-term dependencies as it places a disproportionate weight on the most recent input when predicting the output, with the gradients used for learning the weights reducing exponentially over time. Hence, while RNNs present a promising approach for anomaly detection in time series data, overcoming the vanishing gradient problem remains a crucial challenge in fully exploiting their potential.

Gers, Schmidhuber, and Cummins [20] proposed a new modified type of RNN, the Long Short-Term Memory (LSTM) network, developed to address the vanishing gradient problem of traditional RNN. Figure 2.1¹ depicts the structure of an LSTM cell.

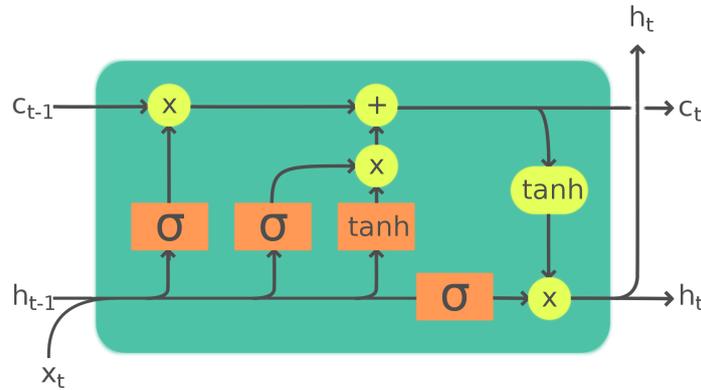


Figure 2.1: LSTM cell structure

LSTMs tackle this problem with a novel cell structure that includes a memory cell and three types of gates: a forget gate, an input gate, and an output gate. Each gate performs a specific function that collectively allows the LSTM to retain and manipulate information over extended periods, hence, overcoming the shortcomings of traditional RNNs.

$$\begin{aligned}
 f_t &= \sigma(W_{fh}h_{t-1} + W_{fx}x_t + b_f) \\
 i_t &= \sigma(W_{ih}h_{t-1} + W_{ix}x_t + b_i) \\
 o_t &= \sigma(W_{oh}h_{t-1} + W_{ox}x_t + b_o) \\
 \tilde{c}_t &= \tanh(W_{ch}h_{t-1} + W_{cx}x_t + b_c) \\
 c_t &= f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \\
 h_t &= o_t \odot \tanh(c_t)
 \end{aligned} \tag{2.1}$$

In the equations above, h is the vector of the hidden layer, c refers to the memory cell vector of the hidden layer, x is the input vector, b is the bias vector, and W is the weight

¹Image by Guillaume Chevalier, adapted under CC BY-SA 4.0 license

matrix. σ and \tanh are the sigmoid and tanh activation functions respectively described in Equation 2.2 and Equation 2.3.

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (2.2)$$

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (2.3)$$

The design of LSTMs enables them to learn and remember patterns over long sequences, making them highly suitable for applications involving time-series data, like telemetry data from CubeSats. By allowing the network to learn long-term dependencies, LSTMs are adept at modelling even complex relationships in the data. The application of LSTMs in various time series prediction applications has been extensively studied and has shown significant improvements over standard RNNs [10], [30], [48], [49].

A study on anLSTM-based anomaly detection method was carried out by Malhotra, Vig, Shroff, *et al.* [33]. An LSTM network was trained using the Space Shuttle Marotta valve data. The prediction error the nominal validation dataset and the LSTM-predicted data was presumed to exhibit a Gaussian distribution ($\varkappa = \varkappa(\mu, \sigma)$). The probability, p , of encountering an error value e is represented by the value of \varkappa at e . An observation x_i was categorized as anomalous if the value of p_i for this observation was smaller than a threshold $p_i < \tau$. The τ value is fixed based on the anomalous validation dataset to ensure optimal precision with the least false-positive rate within the validation dataset. The network demonstrated the ability to predict anomalies with 93% precision. It was observed that the nominal data points succeeding an anomaly were also designated as anomalous.

Hundman, Constantinou, Laporte, *et al.* [25] developed a dynamic thresholding technique for the LSTM-based anomaly detection approach. No assumptions about the error distribution pattern were made when determining the threshold. The prediction errors within the validation datasets were formed into a one-dimensional vector as follows:

$$e = [e^{(t-h)}, \dots, e^{(t-1)}, e^t] \quad (2.4)$$

where h represents the number of historical errors, determined by the size of the validation set. LSTMs are susceptible to error spikes due to sudden input changes, even when these changes signify normal behavior [47]. Hence, the error vector was smoothed using an Exponentially Weighted Moving Average (EWMA). A threshold value applied to the smoothed error values was used to classify if the data point was anomalous. The threshold was chosen such that, if all values above this threshold are discarded from the smoothed error vector e_s , it results in the maximum reduction in the mean and standard deviation of e_s . An anomaly pruning method was also applied to lessen the number of false positives reported by comparing the prediction error change near the anomaly to the standard prediction error noise present. This ensures that the regular noise in the telemetry stream is disregarded. Using this approach, the authors detected anomalies in the Soil Moisture Active Passive (SMAP) satellite and the Mars Science Laboratory (MSL) rover, Curiosity

telemetry data with an average prediction error of 5.9%.

The advantage of employing LSTMs over traditional Multi-Layer Perceptrons (MLPs) is that LSTMs can identify contextual anomalies by learning long-term dependencies in the data. The authors discovered that the LSTM-based approach had an average recall of 69% and an average precision of 90.3% of the point anomalies for the SMAP and MSL telemetry. Using anomaly pruning reduced the recall by an average of 4% while increasing the precision by 39%.

2.2. First-principles neural networks

As discussed briefly in the previous section, neural networks can be trained to predict complex, non-linear system behavior. The black-box nature of such models prevents the user from analyzing the modeling method. The problem of overfitting the network to the training data may render the network unable to extrapolate the relationships learned to real-world data that might lie outside the training data. The number of neurons in the network depends on the modeled behavior's complexity. The number of neurons, and thus the training time, can become significant while modeling systems with many non-linear governing equations.

Modeling the complete behavior of space systems such as the attitude and dynamic control model or the thermal model of a satellite is possible. However, the computational cost for such tasks can be prohibitive, especially for real-time prediction. One might consider a simplified model for ease of computation, but the accuracy of such models might not have an acceptable accuracy under all conditions. To mitigate this, a possible solution could be to use a simplified first-principles physical model, which is then augmented by a neural network to correct the residuals of the physical model. This approach of hybrid networks was studied as early as 1992 by Psychogios and Ungar [40] investigating the modeling of a batch-fed bioreactor.

In their study, the authors compare the performance of a standard MLP, a hybrid network, and an Extended Kalman filter with parameter estimation techniques. The MLP proved effective in predicting cell growth in most areas, but suffered for sections with noisy inputs. Furthermore, the MLP struggled to extrapolate data when inputs fell outside the range of the training data. The hybrid network, on the other hand, displayed better performance than a standard MLP, with enhanced capabilities in prediction and extrapolation. The incorporation of first-principles physics into the hybrid model significantly contributed to these improvements.

Additionally, the hybrid network was compared with a parameter estimation method using Sequential Quadratic Programming (SQP), complemented by an Extended Kalman filter for input noise removal. The researchers found that the hybrid network outperformed the parameter estimation method, particularly when parameters were time-variant.

It is of importance to note that the architecture of networks employed in early hybrid models were relatively basic, consisting of only a few hidden layers [38]. However,

the landscape of recent research regarding hybrid neural networks has expanded to accommodate the use of deep neural networks. These networks, equipped with multiple hidden layers, have led to significant improvements in the accuracy of predictions. Studies carried out by Chaffart and Ricardez-Sandoval [11], Bangi and Kwon [2], and Pinto, Mestre, Ramos, *et al.* [38] have substantiated the efficacy of hybrid networks in modeling complex industrial processes. The increased complexity of the neural network architectures in these studies were made possible due to advancements in computational power. These modern hybrid models exhibited performance trends akin to those observed in the earlier study by Psychogios and Ungar [40], despite the advancements in complexity.

The hybrid models studied by Chaffart and Ricardez-Sandoval [11], Bangi and Kwon [2] and Shah, Sheriff, Bangi, *et al.* [46] were series first-principles networks, with the neural network predicting the parameters of the physical model. The network weights are trained using the prediction error of the hybrid model. Since a black-box model is used to estimate the parameters used in the first-principles model, it can result in scenarios where the hybrid model prediction is accurate while the parameters predicted are physically infeasible. Constraints can be applied to these parameters as a possible solution to make a more robust approximation model. Such a series hybrid network might also fail to accurately predict the system behavior when the underlying first-principles model fails to account for some driving phenomenon of the process. A general series hybrid model is depicted in Figure 2.2.

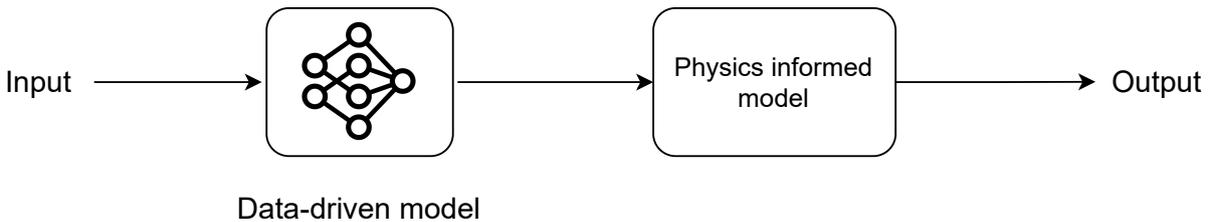


Figure 2.2: Schematic of a series hybrid model

Sun, Yang, Wang, *et al.* [50] studied the prediction capability of parallel hybrid networks, which use a neural network to correct the residuals of the first-principles model. A general parallel hybrid model is depicted in Figure 2.3. Sun, Yang, Wang, *et al.* [50] described the prediction of a cobalt removal process using a parallel hybrid network. The authors compared the hybrid model with the performance of the first-principles model alone and a purely data-driven MLP model. The authors demonstrated that the data-driven model had approximately 50% improvement in the RMSE over the first-principles model they used, and the hybrid model had approximately 50% improvement over the data-driven model.

In a study conducted by Johnson, Quackenbush, Sorensen, *et al.* [29], the application of first-principles based neural networks for the control of soft robots was examined. The employed model utilized two MLPs; a surrogate network developed based on the first-principles physics model, and an error network for compensating the residual inaccuracies of the surrogate network in predicting the actual behavior. Numerous factors pertaining to material properties were neglected in the first-principles physics model. By using current position data and the commanded pressure, the combined surrogate and error networks

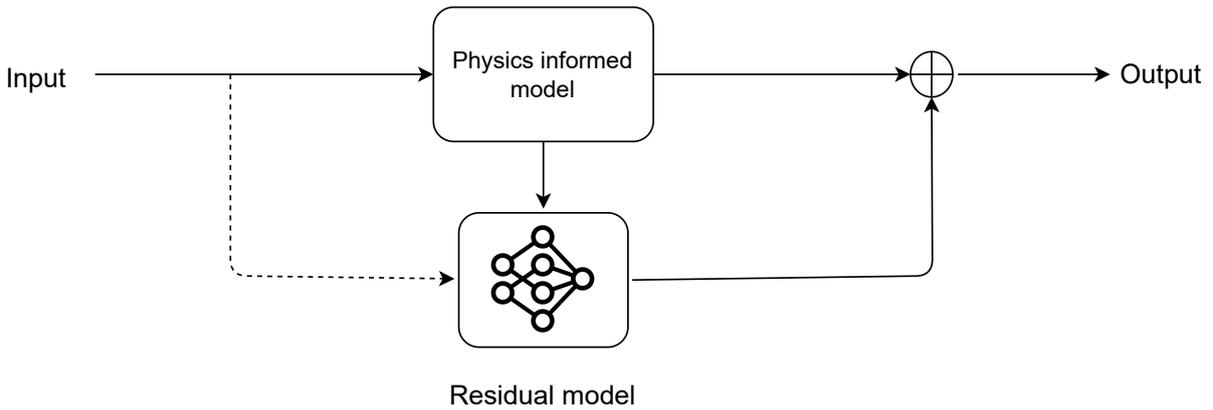


Figure 2.3: Schematic of a parallel hybrid model

were used to predict the robot’s behavior two seconds into the future. A controller loop was developed with the neural network in order to predict future outputs for calculating required inputs for the system to maintain a pre-set tracking angle. The study concluded that the mean tracking error was reduced by 52% when the error network complemented the surrogate network. A substantial reduction in the median tracking error by an order of magnitude was also witnessed. Gathering a large quantity of raw data to construct a pure data-driven neural network state estimator capable of capturing the intricate non-linear behavior with intricate component interactions inherent to the soft robot would have posed a challenge. However, the authors highlighted that the surrogate network could be trained solely on simulated data derived from the first-principles physics model, which was relatively straightforward to procure. Given that the error network is solely responsible for learning the residuals, its size is considerably smaller than the surrogate network, thereby reducing the need for extensive training data and the overall training time. The prediction accuracy of the hybrid model substantiates the capacity of parallel hybrid networks in capturing behavior that remains unmodeled in the first-principles model.

Quadcopters, at low speeds, are fairly uncomplicated systems to model. The modeling can be carried out by employing a mix of momentum exchange theory and blade element theory to ascertain the aerodynamic forces and moments influencing the system. However, these models fail to capture the interaction between the rotor wake and the drone frame, which gives rise to considerable forces and moments at high speeds. This airflow data can be modelled using Computational Fluid Dynamics (CFD) methods. Despite their efficacy, CFD methodologies are computationally taxing, rendering their application in real-time scenarios challenging. Bauersfeld, Kaufmann, Foehn, *et al.* [3] investigated the utility of a hybrid network grounded on the Boundary Element Method (BEM) theory as the first-principles physics model. They also considered other strategies, such as a MLP and a grey-box parametric model [51]. An MLP was run concurrently with the physics model to compensate for the residuals in the BEM model. It was observed that the hybrid model consistently outperformed the alternative methods studied. Even when the hybrid model was trained solely on data corresponding to slow-speed flight, it exhibited superior extrapolation of high-speed flight behavior compared to other approaches. The researchers also emulated different flight trajectories at varying speed regimes to scrutinize the impact

of speed on the predictive proficiency of the studied methods. At slower speeds, nearing hover, all models demonstrated approximately equal performance. Nevertheless, the hybrid BEM model significantly outperformed others in modeling forces and moments at high speeds.

Based on the aforementioned studies, the first-principles neural networks demonstrate superior capability to extrapolate training data for fitting actual data. The imposition of an underlying physics-based model ensures that input interactions align with physical interactions, thus facilitating improved generalization of predictions. These hybrid networks also offer the advantage of probing limitations in the first-principles model, such as overlooked interactions, by examining the residual behavior from the neural network. Series hybrid models excel when the first-principles align reasonably well with real-world modeling effects. However, their performance markedly deteriorates when the first-principles model neglects certain interactions. In contrast, parallel hybrid networks exhibit greater resilience to such modeling limitations.

2.3. Satellites and Telemetry Data

Data from three small satellites, namely Delfi-C³, Delfi-PQ and FUNcube-1 were used for this study

Delfi-C³ is the first satellite developed by TU Delft. It was launched on April 28, 2008, as a secondary payload on the PSLV-C9 rocket from the Satish Dhawan Space Centre in India. Delfi-C³ was inserted into a Sun-synchronous Low Earth Orbit (LEO). The orbital characteristics of Delfi-C³ are summarized in [Table 2.1](#).

Table 2.1: Orbital characteristics of Delfi-C³ (calculated using TLE generated on 2023-06-18 02:21:36)

Apoapsis	461.916 km
Periapsis	449.308 km
Orbital period	93.702 min
Eccentricity	0.00092250
Inclination	97.3012°
RAAN	190.1691°
Argument of periapsis	122.059°

Four Thin-Film Solar Cells (TFSCs), which are technology demonstrators from Airbus DS, Netherlands, along with the On-Board Computer (OBC) and the two Radio Amateur Platforms (RAPs) (RAP-1 and RAP-2), make up the seven available temperature measurements in the telemetry. The four TFSCs are labeled according to their position relative to the body co-ordinate frame depicted in [Figure 2.4](#): TFSC Z+X+, TFSC Z+X-, TFSC Z-Y+ and TFSC Z-Y-. The other technology demonstrator aboard the CubeSat, the Autonomous Wireless Sun Sensors (AWSSs) developed by TNO, Netherlands, was

intended to provide attitude measurements. The data from the AWSSs, however, have been challenging to recover [22].

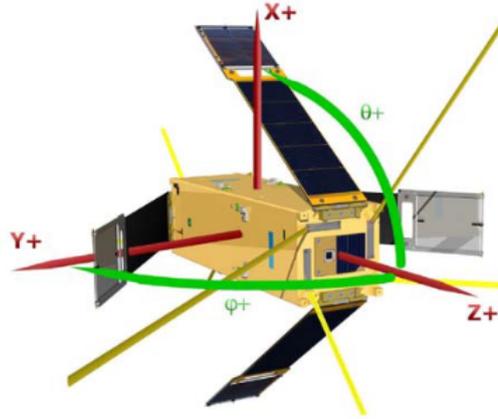


Figure 2.4: Delfi-C³ [22]

Delfi-PQ is the third satellite developed by TU Delft. Launched on January 13, 2022, as a rideshare on the SpaceX Transporter-3 mission, Delfi-PQ was inserted into a Sun-synchronous LEO. The orbital characteristics of Delfi-PQ are summarized in [Table 2.2](#).

Table 2.2: Orbital characteristics of Delfi-PQ (calculated using TLE generated on 2023-06-18 02:21:36)

Apoapsis	463.986 km
Periapsis	451.745 km
Orbital period	93.748 min
Eccentricity	0.00089530
Inclination	97.4595°
RAAN	240.9528°
Argument of periapsis	240.8804°

The Delfi-PQ telemetry contains seven temperature channels: one each for the four solar panel blocks (solar panels Xp, Xm, Yp and Ym), one on the Microcontroller Unit (MCU) and two for the battery. The two battery temperatures are measured using two different sensors, with the data from the TMP20 sensor chosen for use in this study due to its superior data quality, as ascertained from the preliminary analysis of the telemetry.

FUNcube-1, depicted in [Figure 2.5](#)¹, is a 1U CubeSat developed by AMSAT-UK. It was launched on November 21, 2013, as a secondary payload on the Dnepr rocket from the Yasnny launch base in Russia. FUNcube-1 was inserted into a sun-synchronous LEO. The orbital characteristics of FUNcube-1 are summarized in [Table 2.3](#).

The FUNcube-1 telemetry contains eight temperature channels: black chassis, silver chassis, black panel, silver panel, solar panels +X -X +Y and solar panel -Y.

¹Picture by Wouter Weggelaar, used under CC BY 3.0



Figure 2.5: FUNcube-1

Table 2.3: Orbital characteristics of Funcube-1 (calculated using TLE generated on 2023-06-18 02:21:36)

Apoapsis	643.956 km
Periapsis	572.501 km
Orbital period	96.858 min
Eccentricity	0.00511390
Inclination	97.6713°
RAAN	132.6663°
Argument of periapsis	272.0532°

3

Research questions

The state-of-the-art described in the preceding chapters demonstrate the potential of utilizing LSTM networks in time series predictions. These predictions serve as an initial step in the anomaly detection methodologies developed by Malhotra, Vig, Shroff, *et al.* [33] and Hundman, Constantinou, Laporte, *et al.* [25].

The research cited in [Section 2.2](#) showcases the application of first principles-based neural networks in modeling complex systems. The findings demonstrate that hybrid models can perform better when extrapolating training data to fit actual data. Improved performances are due to the machine learning model having access to the first-principles model, which keeps the input interactions consistent with physical interactions. This allows for better generalization of the predictions. By observing the behavior of residuals from a neural network, we can examine limitations in the first-principles model, such as overlooked interactions. The series hybrid models exhibited good performance when the first principles were relatively precise in real-world modeling effects. However, their performance significantly deteriorates when there are neglected interactions in the first principles model. Conversely, parallel hybrid networks demonstrate more resilience to such modeling constraints.

More precise modeling of telemetry data can augment the performance of anomaly detection methodologies. Studies into the use of first-principles neural networks for space-based applications is currently limited, with much of the studies primarily focusing on chemical and industrial process modeling.

This study aims to bridge this gap, by investigating the utilization of first-principles neural networks for modeling satellite telemetry data. Furthermore, the comparison of the results of these models with those of pure data-driven models can be used to validate the hypothesis that the first-principles models are superior at extrapolating data for satellite telemetry data. The findings from this study can be leveraged to enhance the performance of satellite models used in anomaly detection methodologies.

Purely data-driven models limited in their ability to extrapolate data, especially important to Delfi-PQ where the available telemetry data is discontinuous and sparse. Telemetry data for CubeSats by TU Delft, namely Delfi-C³ and Delfi-PQ, are readily available for

use in this study. In addition, limited data also available for FUNcube-1. Given the multiple temperature sensors installed on these satellites, from a preliminary analysis of the telemetry data, the satellite's thermal behavior was identified as a suitable candidate for modeling using first-principles neural networks.

Furthermore, two distinct first-principles models were identified for implementation in this study, with varying degrees of complexity. The first is a lumped capacitance model, and the second is a thermal network model. Despite their inherent differences, both models offer the advantage of simplicity in both their implementation and evaluation. This translates to relatively low inference times for the models, making them an efficient choice for this study.

The following research questions were identified for this study:

1. **How effective are the data-driven neural networks in predicting the temperatures of small satellites?**
 - i What prediction accuracy can be achieved using LSTM models when full orbit data is available?
 - ii What input data is required for the models?
 - iii How does the performance of these models differ between nominal and anomalous regions?
 - iv What is the effect of using residual networks in improving temperature prediction accuracy?
 - v How can the models cope with regions without data, such as for Delfi-PQ?
2. **How does the performance of first-principles model compare to the data-driven models?**
 - i What are the major sources of error in first-principles models, and how do they compare to data-driven models?
 - ii How does the performance vary with the different first-principles models?
 - iii How does incorporating physics-based models affect the performance in regions without data?
 - iv What role do residual networks play in improving/reducing the performance of first-principles models?
 - v How does the performance of first-principles models vary with different assumptions, especially on heat input?
3. **Can the evolution of the physical parameters of the satellite be estimated from the thermal network optimization process?**
 - i What physical parameters are critical to the thermal model of the satellite?
 - ii Can the available data be used to extract changes in the parameters of the first-principles model?

The first sub-question aims to develop a baseline for the performance of the data-driven models. LSTM models were chosen as the base data-driven model due to the architecture

of the LSTM cell being well-suited for time series predictions due to the presence of a memory cell. This has also been demonstrated in the studies cited in [Chapter 2](#).

The second sub-question aims to compare the performance of the first-principles models with the data-driven models, and to identify the major sources of error in the first-principles models.

The third sub-question aims to identify the physical parameters that are critical to the thermal model of the satellite, and to identify if the available data can be used to extract changes in the parameters of the first-principles model, which are very challenging to extract from purely data-driven models or to measure in-orbit.

4

Methodology

In this chapter, the methodology used to develop the thermal models, the rationale behind the choices made, and a high-level overview of the key results to justify the choices are discussed.

4.1. Data pre-processing

The telemetry from all three satellites was pre-processed to remove outliers before being used for training and testing the machine learning models. The telemetry from Delfi-C³ and Delfi-PQ is sparse, with data available only over receiving ground stations. For Delfi-C³, the absence of an on-board battery meant that the telemetry was only available when the satellite was in sunlight. The telemetry from FunCUBE-1, on the other hand, was available as a regularly sampled continuous time series, with one telemetry frame available per minute.

The telemetry frames were first transformed from time domain to spacial domain using the orbital overlaying technique described in [Section 6.1](#). This was done to reduce the sparsity of available telemetry for Delfi-C³ and Delfi-PQ, by mapping all available telemetry onto a single orbit. To keep the analyses consistent, the orbit overlaying was also performed on FunCUBE-1 data, even though the telemetry was available as a continuous time series.

As a prerequisite for the orbit overlaying, the position of the satellites at the time of each telemetry frame was required. This was obtained by propagating the TLEs of the satellites, using the Simplified General Perturbations 4 (SGP4). The selection of the SGP4 was based on the results described in [Section 5.2](#). To make the computation of the angles required for the orbit overlaying efficient, the position, and velocity vector of the satellites were initialized for every available TLE and propagated until the next available TLE using the SGP4. The TLEs do not provide the exact position of the satellite at $epoch = 0$, and contains some positional and velocity inaccuracy. Racelis and Joerger [41] show that the position error of the TLEs can be expected to have a standard deviation of $< 6 \text{ km}$ centered around $\pm 1 \text{ km}$. At LEO orbital velocity, this represents a “time” error of less

than 1 s, which was considered acceptable for modeling the relatively slower thermal behavior of the satellites. 3D Cartesian interpolators were then constructed using the propagated position and velocity vectors, and stored for use during the orbit overlaying. The use of pre-computed interpolators reduced the look-up time for the position and velocity vectors at any epoch within the time range of the available TLEs.

A Python package, *SSA-Calc*, was developed based on work by Zhang [53] to calculate the angles required for the orbit overlaying using the position and velocity vector interpolators. The orbit overlaying, Additionally, was used to conveniently visualize the large amount of telemetry data available for the three satellites. The telemetry was found to contain several outliers, which were cleaned using the procedure described in Section 6.2. The cleaned telemetry was then used for training and testing purely data-driven neural network models and first-principles based models.

4.2. Data-driven models

Purely data-driven models were used to establish a benchmark to compare the first-principles based models against. An important restriction was set on the data-driven models: the models only have access to data that the satellites can reasonably have access to in space, to allow future studies into the possibility of using the models onboard the satellite for autonomous operations. Thus, inputs such as the orbital position of the satellites were not used.

As an initial test case, a temperature-based univariate time series prediction problem was investigated for FunCUBE-1. The black panel temperature was selected as the channel investigated due to the large fluctuations from the mean temperature trend due to tumbling, presenting a challenging case to model. The prediction problem was formulated as follows:

Predict the black panel temperature T_{n+1} at time step t_{n+1} using the black panel temperatures $[T_{n-w+1}, T_{n-w+2}, \dots, T_{n-1}, T_n]$ at time steps $[t_{n-w+1}, t_{n-w+2}, \dots, t_{n-1}, t_n]$ as the inputs, where w is the input window size.

Since the FunCUBE-1 telemetry is available at 1 min interval, an input window size of $w = n$ corresponds to n min of previous telemetry. The input window size was found to have a significant impact on the prediction error of the model. For small window sizes of $w < 3$, the model was unable to fully resolve the temperature fluctuations due to tumbling. With an increase in w , up to approximately $w \approx 10$, the ability of the model to resolve fluctuations due to tumbling got better. For even larger input window sizes, the models regressed towards the mean temperature trend over the orbit. Since the weights and biases of each LSTM cell in the network is initialized pseudo-randomly before training, the exact boundary of these three regions is fuzzy.

To check the robustness of the model for real-life applications, the model predictions were tested with temperature data known to contain anomalies. Ideally, the model prediction error in the anomalous region would be expected to be larger than the nominal region,

with the prediction error returning to nominal as soon as the anomaly ends. This would allow the anomalous region to be successfully identified and isolated. Since the model input includes any previous anomalous temperatures, the predicted temperatures for small window sizes were observed to fit to the anomalous behavior, with the prediction error being similar to nominal data. For larger input window sizes, with more time steps required to fill the input window fully with the anomalous inputs, the prediction error was observed to be large in the anomalous region. But, due to the larger window size, the prediction error continued to be large even after the anomaly had ended.

As a possible solution to decouple the predicted temperatures from previous measured temperatures, an illumination-based model was investigated. The new prediction problem had the following formulation:

Predict the black panel temperature T_{n+1} at time step t_{n+1} using the satellite illumination $[I_{n-w+1}, I_{n-w+2}, \dots, I_{n-1}, I_n]$ at time steps $[t_{n-w+1}, t_{n-w+2}, \dots, t_{n-1}, t_n]$, where w is the input window size. The illumination input $I_i = 1$ if the satellite is in the illuminated section of the orbit at time step t_i , or $I_i = 0$ if the satellite is in eclipse.

The illumination status of the satellite can be possibly inferred based on the solar panel voltages, without requiring any orbital position information. FunCUBE-1's orbit has a fraction of eclipse of $f_e \approx 0.3$. The minimum input window size required to assign a unique sequence of illumination values to all sections of the orbit is $w_{min} = P \cdot (1 - f_e) \approx 68$ minutes, where P is the orbital period. The model, similar to temperature-based models with large input window sizes, regressed to the mean temperature trend. Since the satellite tumbling is not synchronous with the orbital period, without additional attitude information, the model was unable to resolve the temperature fluctuations due to tumbling.

Since the illumination-based models were decoupled from measured temperatures, anomalous temperatures did not affect the model predictions. Thus, by monitoring the prediction error, the anomalous regions can be successfully identified and isolated.

To resolve the temperature fluctuations due to tumbling, a residual network for the illumination-based model was investigated. The illumination-based model prediction and a small window of previous measured temperatures were used as the inputs to the residual network. The prediction problem for the residual network was formulated as follows:

Predict the base model residual R_{n+1} at time step t_{n+1} using the base model prediction T_{n+1} and the previous measured temperatures $[T_{n-w+1}, T_{n-w+2}, \dots, T_{n-1}, T_n]$ at time steps $[t_{n-w+1}, t_{n-w+2}, \dots, t_{n-1}, t_n]$ as the inputs, where w is the input window size.

The use of residual network allows having two different models: the base model which is independent of temperatures measured by on-board sensors, and a combined base and residual model which incorporates the measured temperatures. The combined model prediction was calculated as $T'_{n+1} = T_{n+1} + R_{n+1}$. Since the input window sizes investigated were small, the residual model was built as a feed-forward MLP instead of a LSTM network. For residual window sizes of $w > 5$, the combined model was able to resolve the temperature fluctuations due to tumbling, with larger step sizes having diminishing effect on the prediction error. For an illumination-based model with $w = 96$

and a residual model with $w = 10$, the combined model prediction error on the test data was found to have $\mu = 0.0^{\circ}C$ and $\sigma = 0.9^{\circ}C$.

The combined model was also found to be scalable to predict on all eight available temperature channels in the telemetry without any changes to the model architecture. With the additional information available from the other temperature channels, the combined model was able to resolve the temperature fluctuations due to tumbling even for small input window sizes of $w = 2$ for the residual network.

Delfi-PQ, with its telemetry being sparse and containing sections of orbit with no data, demonstrated the limitations of using a purely data-driven models. The predicted temperatures for sections of orbit with no data were found to be inconsistent with the expected temperature trends in these regions, thus reducing the confidence in the model predictions in these regions.

4.3. First-principles based models

As described in the previous section, purely data-driven models were found to have bad generalization in regions of orbit with no available training data. Since these data-driven models are black-box models, it is difficult to understand the exact behavior being modelled by the network. As a possible solution, hybrid physics-informed first-principles models were investigated. The inclusion of first-principles models in the hybrid model ensures that the model is consistent with the approximate, but known, physical behavior of the system. The first-principles models were also found to be robust to the lack of training data in regions of orbit with no data.

Two first-principles models were investigated: an exponential curve-fit based on the lumped capacitance model, and a thermal network model. If only conductive heat transfers are considered for the nodes, the rise, and fall of temperatures of the nodes can be modelled using exponential curves [5]. A curve-fit model was studied by using the training dataset to fit exponential curves for each of the available temperature channels to use as the first-principles model.

The use of such a curve-fit approach as the first-principles model was validated using the full orbit data available for FunCUBE-1. The curve-fit model was observed to have similar prediction error as the previously investigated illumination-based LSTM model. With its use validated on FunCUBE-1 data, a similar approach was used to improve the prediction confidence for Delfi-PQ telemetry in regions of orbit with no data. Similar to FunCUBE-1, the illumination-based LSTM model and curve-fit model had similar prediction error for Delfi-PQ data.

A hybrid LSTM model, combining the advantages of both data-driven and physics-informed approaches, was investigated by adding synthetic temperatures obtained using the curve-fit model to the available training dataset for Delfi-PQ. The resulting hybrid model was observed to have the same predictions as the illumination-based models for regions of orbit with available telemetry and predictions being closer to probable true temperatures

for sections without data, though this cannot be tested without additional data. Since, the predictions of the hybrid model and the illumination-based model were similar for regions with data, the residual networks developed for the illumination-based model could be reused with the hybrid model without further modifications.

The curve-fit model, though being better than purely data-driven models for predictions in regions with no data, still was limited due to each node being modelled independently due to the separate curve-fit performed per node. Though the data-driven part of the hybrid model might have learned the connection between the nodes, since they were not explicitly defined, the physical accuracy of such learned connections is not known. All heat transfers are also assumed to be conductive in the curve-fit model, which is a close approximation for the heat transfers between the nodes, but not for the heat received by the satellite from the Sun and Earth and the heat loss due to radiative emissions.

A more complex first-principles model, a thermal network model based on the electrical equivalence of heat transfer between the nodes [4], was studied as another physics informed model to tackle the limitations of the curve-fit model. The thermal network model for Delfi-PQ developed by Ruiz [44] and Cinotti, Lusvarghi, Marchese, *et al.* [13] was used as the starting model. The thermal simulation of the temperatures of the nodes was performed with several assumptions and simplifications to reduce the complexity of the model, described in more detail in Section 8.2.2. As a result of these assumptions, the thermal network model was found to have a large prediction error when compared to the measured temperatures. To improve the accuracy of the thermal network model, the parameters of the model, namely the absorptivity (α) and emissivity (ϵ) of the six outer panels, and the heat capacities of the battery and the MCU, were tuned using a stepwise optimization approach, using Nelder-Mead optimization algorithm [17], to minimize the RMSE of the model predictions. The resulting thermal network model was found to have a prediction error of $RMSE = 3.65^\circ C$ on the test data, which is an improvement over the prediction error of $RMSE = 7.5^\circ C$ for the original thermal network model. Similar to the curve-fit models, the prediction error of the physics-informed base model could be reduced by adding a data-driven residual network to the model. The inputs to the residual network were the base model prediction and the previous measured temperatures in an input window of size w .

The satellite's physical properties are expected to evolve with time due to aging. The mean temperature trend of the satellite is also expected to vary not only due to the changing physical properties but also due to the seasonal and long-term changes in the solar flux received by the satellite. To attempt to observe the impact of these changes on the optimized parameters of the thermal network model, the temperature telemetry was batched into periods of intervals ranging from 30 to 90 days, and the variation of the optimization results were investigated for different batches.

4.4. Computational framework

It was considered important to use open-source software packages to develop the tools and programs required to reach the proposed objectives. The use of open-source software packages would enable others to easily adapt the outcome of the thesis project to their requirements, enabling further technological development in the field. Python was chosen as the programming language of choice based on well documented pre-existing modules to build machine learning models and the availability of community support for commonly known issues. This enabled an efficient workflow for this thesis.

SatMAD, an open-source Python package with in-built integration with TLEs, was used for orbit propagation. All neural networks were implemented in Tensorflow, a feature-rich module for machine learning applications, with Graphical Processing Unit (GPU) accelerated computations enabled through the Nvidia CUDA and cuDNN libraries. The curve-fit and optimization algorithms were implemented using SciPy.

5

Orbit Propagation

5.1. Two-line elements

The TLE set is a data format used to encode sets of orbital elements at a point in time, or epoch, that describe the trajectories of Earth-orbiting satellites. The format was originally designed by the North American Aerospace Defense Command (NORAD) to make it easy for computers of the era to track satellites and predict their future locations.

A TLE is composed of two (and optionally a third) lines of ASCII text, each containing a series of specific and precisely formatted data elements. Each line contains a set of parameters that provide essential details about a satellite's orbit. An example of a TLE is shown in Figure 5.1, and the description of the fields in the TLE are summarized in Table 5.1.

```
      u
      DELFI C3

a      b      c      d      e      f      g      h      i      j
1  32789U  08021G  23001.00216029  .0002466  00000-0  11529-2      0      999  4
2  32789  97.3241  25.6810  0009233  7.1097  353.0268  15.19729105801961  8019  6
k      l      m      n      o      p      q      r      s      t
```

Figure 5.1: Example of a TLE

For this thesis project, the TLEs for the satellites Delfi-C³, Delfi-PQ, and FUNcube-1 were retrieved from the Space-Track¹ database. The histograms of the time intervals between all available TLEs of these three satellites, up to February 2023, are graphically represented in Figure 5.2. Table 5.2 summarizes key statistical data concerning this distribution. Notably, the average time interval between the TLEs for all the satellites is less than a day, with very few instances where the time difference exceeds two days. Consequently, the selection of an appropriate orbit propagator for this project was based on an assessment of the accuracy of various propagators over a propagation duration of two days. This choice was deemed rational given the relatively short average time interval between subsequent TLEs.

¹<https://www.space-track.org>

Table 5.1: Description of the fields in the TLE

Line 0 (optional)

Field	Description
u	Line number, always 0

Line 1

Field	Description
a	Line number, always 1
b	Satellite catalog number with classification type (U: unclassified, C: classified, S: secret)
c	International designator
d	Epoch (first two digits: year, remaining digits: day of the year)
e	First derivative of mean motion
f	Second derivative of mean motion (decimal point assumed)
g	B* drag term
h	Ephemeris type (always 0)
i	Element set number
j	Checksum

Line 2

Field	Description
k	Line number, always 2
l	Satellite catalog number
m	Orbit inclination (degrees)
n	Right ascension of the ascending node (degrees)
o	Eccentricity (decimal point assumed)
p	Argument of perigee (degrees)
q	Mean anomaly (degrees)
r	Mean motion (rev / day)
s	Revolution number at epoch
t	Checksum

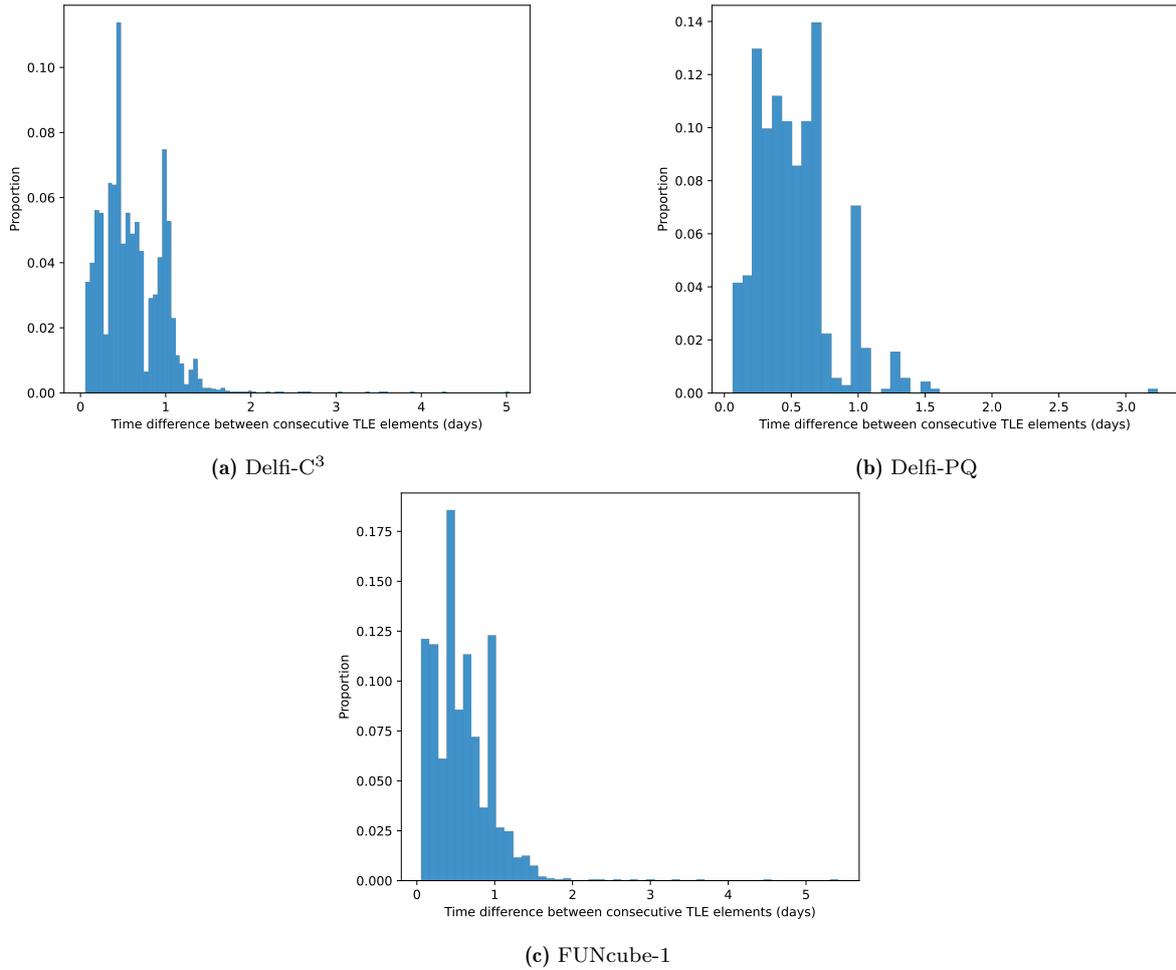


Figure 5.2: Histogram of the time between TLEs

Table 5.3: Time difference statistics between TLEs for Delfi-C³, Delfi-PQ, and FUNcube-1

Satellite	TLE time range	TLE time difference statistics		
		Maximum (days)	$\mu(days)$	$\sigma(days)$
Delfi-C ³	2008-20203	5.03	0.61	0.35
Delfi-PQ	2022-2023	3.23	0.53	0.29
FUNcube-1	2013-2023	5.42	0.58	0.37

5.2. Orbit propagators

The TLEs provide the ability to determine a satellite's position and velocity at the specific epoch of the TLE's formulation. However, these epochs do not align with the epochs of the available telemetry data. To address this discrepancy, numerical orbit propagation was used to calculate the intermediate positions and velocities between the available TLE epochs. This approach ensured a seamless representation of the satellite's motion, bridging the temporal gaps between subsequent TLEs.

SatMAD, an open-source Python package offering orbit propagation capabilities [26], was chosen as the computational framework for orbit propagation for this study, primarily due to its robust interoperability TLEs. Two propagation models were investigated: a numerical integration model based on a two-body system, and a propagator relying on the Simplified General Perturbations 4 (SGP-4) model.

5.2.1. Two-body model

A simple differential equation for the motion of a satellite around Earth is described in Equation 5.1. Some assumptions made here are:

- The Earth is modeled as a perfect sphere.
- The satellite is modeled as a point mass orbiting the Earth.
- The only force involved is the gravitational force between the satellite and the Earth. No other forces, such as atmospheric drag or radiation pressure, are considered.

$$\ddot{\vec{r}} = -\frac{\mu}{r^3} \vec{r} \quad (5.1)$$

In Equation 5.1, \vec{r} is the position vector of the satellite, μ is the gravitational parameter of the Earth, and r is the magnitude of the position vector.

Given an initial position and velocity vectors of a satellite, derived from the TLE, the position and velocity vectors at a future time can be determined by numerically integrating the Ordinary differential equation (ODE) described in Equation 5.1. Due to the simplicity of the model, the computational cost of the propagation is small.

Based on the results of the accuracy tests described in the SatMAD documentation^{2,3}, the DOP853 algorithm, which is an explicit eighth order Runge-Kutta method, was chosen as the ODE solver, with an absolute error tolerance of 10^{-14} and a relative error tolerance of 10^{-12} for each time step.

To assess the inaccuracy in the predicted satellite position, the satellite position and velocity were initialized by selecting 500 random TLEs for each of the three satellites. After a propagation duration of two days, the predicted positions and velocities were compared against other available TLEs available for the propagation duration. Figure 5.3 depicts the mean and standard deviation of the predicted position and velocity errors of the three satellites.

Since all three satellites are on similar orbits, their position errors were also observed to be similar (≈ 1000 km), which was deemed too large to be used for predicting the temperature of the satellites. To understand the source of this error, the predicted specific energy of the satellite was compared with the specific energy calculated from the TLEs. The specific energy was calculated using the *vis-viva* equation described in Equation 5.2.

²https://satmad-applications.readthedocs.io/en/latest/analyses/propagation/num_prop_performance_1.html

³https://satmad-applications.readthedocs.io/en/latest/analyses/propagation/num_prop_performance_2.html

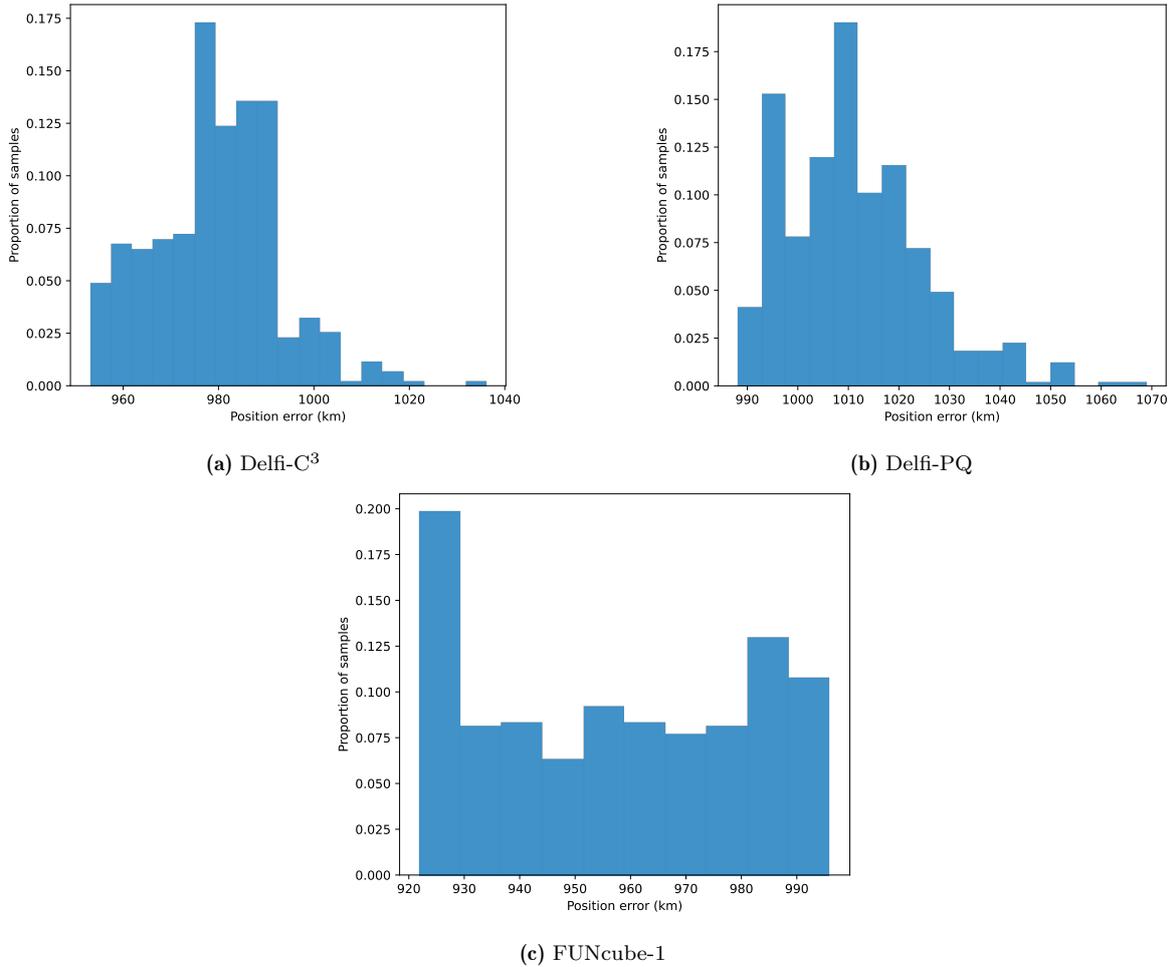


Figure 5.3: Histogram position errors of two-body model after a propagation duration of two days

$$\epsilon = \frac{v^2}{2} - \frac{\mu}{r} \quad (5.2)$$

In Equation 5.2, ϵ is the specific energy, v is the magnitude of the velocity vector, and r is the magnitude of the position vector.

Figure 5.4 depicts the results of this analysis for Delfi-PQ, with the results for the other two satellites being similar. The results indicate that the loss in specific energy of the satellite due to atmospheric drag, radiation pressure, non-spherical Earth, etc. are not being accounted for in the two-body model. Since the three satellites are in low Earth orbit, the atmospheric drag is the most significant perturbing force, and the two-body model is not suitable for predicting the satellite position.

5.2.2. Simplified General Perturbations 4 model

The Simplified General Perturbations 4 (SGP-4) model, developed by Ken Cranford and T. S. Kelso, is a widely used model for orbit propagation for satellites in Earth orbit.

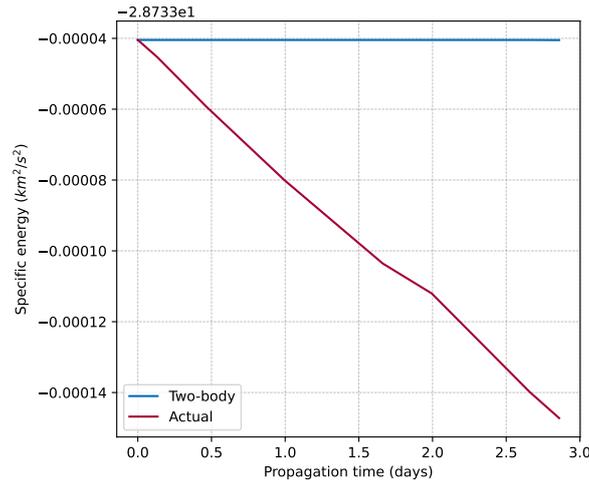


Figure 5.4: Specific energy of Delfi-C³ starting from 2017-07-04 05:54:27 to 2017-07-07 02:31:32 as predicted by the two-body model and calculated from TLEs

Furthermore, the model is deeply integrated with the use of TLEs. The key advantage of the SGP-4 model is its ability to predict the effects of atmospheric drag on a satellite's orbit, which can be significant for satellites in LEO. To model the atmospheric drag, the SGP-4 incorporates drag coefficients and atmospheric density models. In addition to atmospheric drag, other perturbations such as gravitational resonances caused by the Moon and the Sun, and the effect of Earth's oblateness are also considered by SGP-4. These perturbations can cause significant deviations from the idealized two-body motion, as described in [Section 5.2.1](#).

However, long-term perturbations caused by effects such as atmospheric variations and solar radiation pressure are not considered by the SGP-4 model. Since the time between subsequent TLEs available for Delfi-C³, Delfi-PQ and FUNcube-1 is fairly small, the SGP-4 model can be reinitialized frequently to account for the above-mentioned effects. The SGP-4 model is implemented in SatMAD, and the same methodology described in [Section 5.2.1](#) was used to assess the accuracy of the SGP-4 model. The results of this analysis are depicted in [Figure 5.5](#).

The results indicate that the SGP-4 model is significantly more accurate than the two-body model, with the mean position errors being of $O(10^0)$ km for all three satellites. The velocity errors are also observed to be small, of $O(10^{-2})$ km s⁻¹ for all three satellites. By comparing the specific energy of Delfi-PQ predicted by the SGP-4 model and the same energy calculated from TLEs, the SGP-4 model was also found to be more accurate than the two-body model. The result of this comparison is depicted in [Figure 5.6](#)

For this study, SGP-4 was chosen as the orbit propagator for all three satellites. The SGP-4 model was reinitialized at every epoch of the available TLEs, and the propagation duration was set to be the time difference between two subsequent TLEs.

The propagation step size used for the SGP-4 propagator has a significant impact on both the predicted position and velocity errors, and the computation time. This was studied by evaluating by initializing the SGP-4 model with randomized initial TLE and propagating the orbit for a duration of 2 days for various propagation step sizes ranging from 10 to

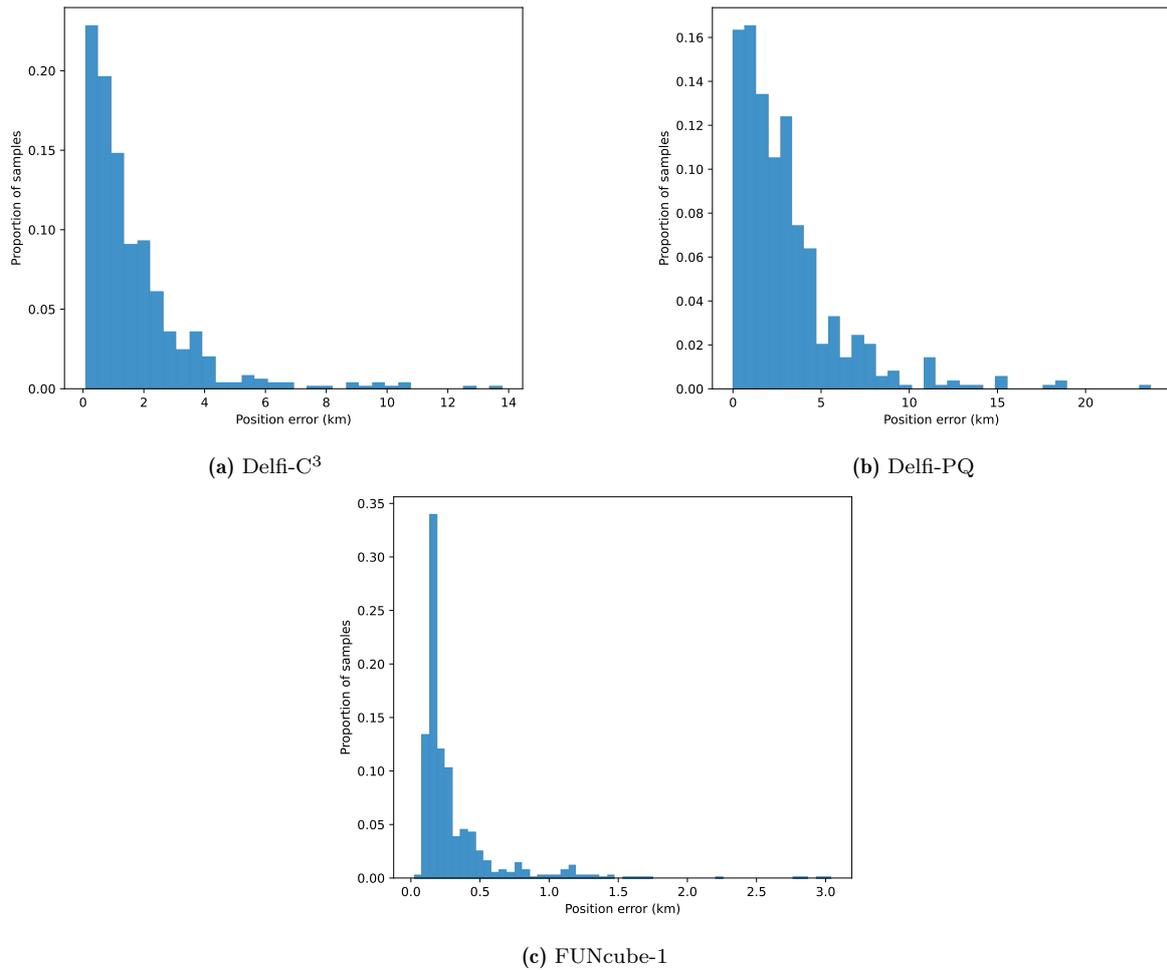


Figure 5.5: Histogram position errors of SGP-4 model after a propagation duration of two days

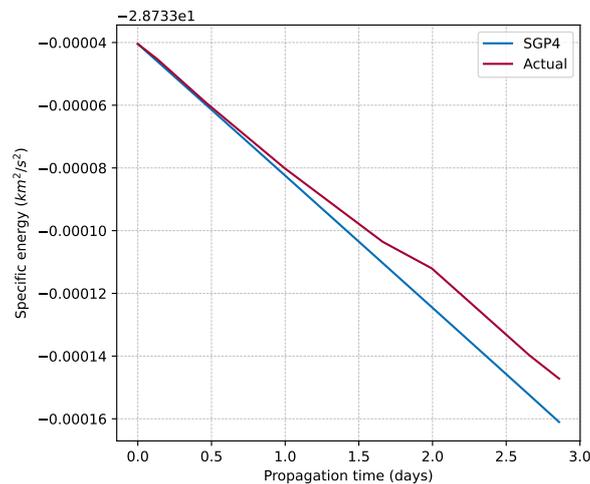


Figure 5.6: Specific energy of Delfi-C³ starting from 2017-07-04 05:54:27.610 to 2017-07-07 02:31:32.086 as predicted by the SGP-4 model and calculated from TLEs

500 seconds. The predicted position error and the computation time required for each step size could then be analyzed to determine the optimal step size. [Figure 5.7](#) depicts

the results of this analysis for Delfi-PQ, with the results for the other two satellites being similar. The results indicate that the position error remains relatively constant for step sizes lesser than 200 seconds, and then increases for larger step sizes. The computation time, on the other hand, reduces exponentially with increasing step size. Since the orbit propagation had to be performed only once, accuracy was prioritized over computational cost. A propagation step size of 100 seconds was chosen for this study, as it provides a good balance between the predicted position error and the computation time.

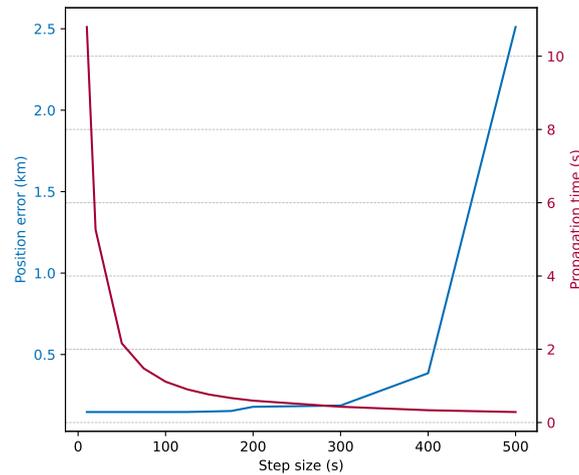


Figure 5.7: Effect of propagation step size on the predicted position error and computation time for SGP-4 model

6

Data preprocessing

Data preprocessing is an important step in building effective machine learning models. It involves transforming and cleaning of raw data into a format that is suitable for use with machine learning algorithms.

Raw data often has issues, such as missing values, inconsistencies, and errors. These issues can have a significant impact on the performance of machine learning algorithms, leading to inaccurate and unreliable results.

The data preprocessing steps of orbit overlaying and removing outliers are described in this chapter.

6.1. Orbit overlaying

The sparse nature of Delfi-C³ and Delfi-PQ telemetry data provides a challenge for building machine learning models. The method to overlay orbits proposed by Zhang [53] was adapted for use with all three satellites analyzed for this study. This enabled transforming the sparse data, distributed over a large time range, into a dense data mapped from time to angle along orbit.

The angle between the Sun vector \vec{S} and the orbital plane, named β for this study, can be calculated using Equation 6.1.

$$\beta = \frac{\pi}{2} - \cos^{-1} \left(\frac{\vec{H} \cdot \vec{S}}{|\vec{H}| |\vec{S}|} \right) \quad (6.1)$$

In Equation 6.1 \vec{H} is the angular momentum vector of the satellite, and \vec{S} is the Sun vector.

The angular momentum vector \vec{H} for the satellite in Earth orbit can be calculated using Equation 6.2. The Sun vector \vec{S} can be obtained from the ephemeris for the Sun at

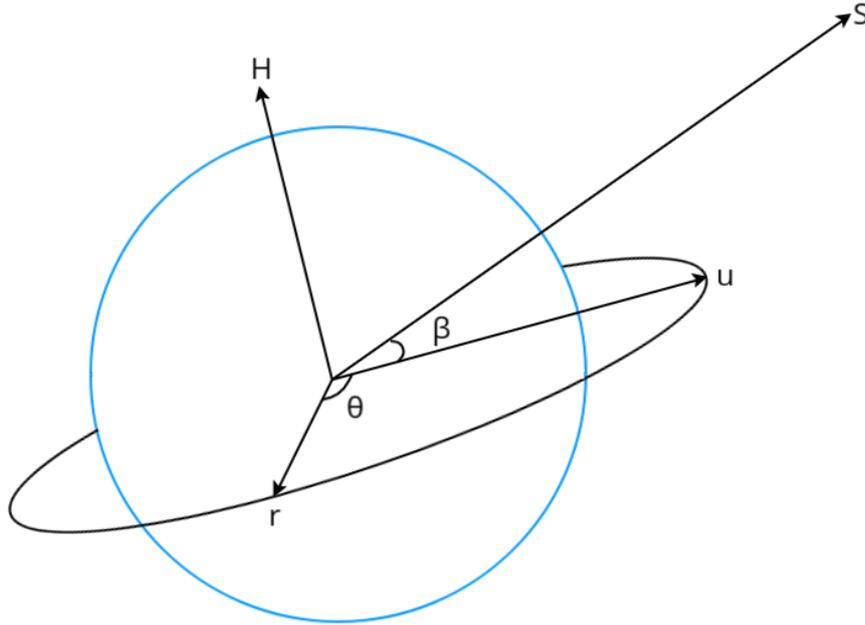


Figure 6.1: Schematic of orbit overlaying (adapted from [53])

satellite epoch time.

$$\vec{H} = \vec{r} \times \vec{v} \quad (6.2)$$

In Equation 6.2 \vec{r} is the satellite position vector, and \vec{v} is the satellite velocity vector.

The vector \vec{H} can now be scaled to form the vector \vec{N} , which is the vector from the orbital plane to the Sun, using Equation 6.3.

$$\vec{N} = \frac{\vec{H}}{|\vec{H}|} \cdot |\vec{S}| \sin \beta \quad (6.3)$$

In Equation 6.3 \vec{H} is the angular momentum vector of the satellite, \vec{S} is the Sun vector, and β is the angle between the Sun vector \vec{S} and the orbital plane.

If the vectors \vec{H} and \vec{S} do not point towards the same side of the orbital plane, the angle β calculated using Equation 6.1 will be negative. This ensures that the vector \vec{N} calculated using Equation 6.3 always points from the orbital plane to the Sun.

The Sun vector \vec{S} can be projected onto the orbital plane to form the vector \vec{u} using Equation 6.4.

$$\vec{u} = \vec{S} - \vec{N} \quad (6.4)$$

The angle θ is defined as the angle from the projected Sun vector \vec{u} to the satellite position vector \vec{r} , in the direction of the satellite's motion. It can be calculated from the angle between \vec{u} and \vec{r} using Equation 6.5.

$$\theta = \begin{cases} \cos^{-1} \left(\frac{\vec{u} \cdot \vec{r}}{|\vec{u}| |\vec{r}|} \right) & \text{if } \vec{H} \cdot (\vec{u} \times \vec{r}) \geq 0 \\ 2\pi - \cos^{-1} \left(\frac{\vec{u} \cdot \vec{r}}{|\vec{u}| |\vec{r}|} \right) & \text{if } \vec{H} \cdot (\vec{u} \times \vec{r}) < 0 \end{cases} \quad (6.5)$$

The angle θ is zero when the satellite position vector \vec{r} is aligned with the projected Sun vector \vec{u} , and increases as the satellite moves in the direction of its motion. For a section of orbit, from $\pi - \theta_e/2 < \theta < \pi + \theta_e/2$, where θ_e is the angular extent of the section of orbit in eclipse, the satellite is in eclipse. Since the Solar radiation is a major factor influencing the temperature of the satellite, it is useful to transform the angle θ to θ_T such that the angle θ_T is zero when the satellite exits eclipse and increases as the satellite moves in the direction of its motion. The fraction of the orbit in eclipse can be calculated using [Equation 6.6](#).

$$\theta_e = \begin{cases} 2 \cos^{-1} \left(\frac{\sqrt{h^2 + 2R_e h}}{(R_e + h) \cos \beta} \right) & \text{if } |\beta| < \beta^* \\ 0 & \text{if } |\beta| \geq \beta^* \end{cases} \quad (6.6)$$

In [Equation 6.6](#) β^* is the critical angle for the orbit to have an eclipse, calculated using [Equation 6.7](#), h is the altitude of the satellite, and R_e is the radius of the Earth.

$$\beta^* = \sin^{-1} \left(\frac{R_e}{R_e + h} \right) \quad (6.7)$$

[Equation 6.6](#) and [6.7](#) are valid only for circular orbits. Given the orbits of the three satellites analyzed have very low eccentricity, the errors introduced by using [Equation 6.6](#) and [6.7](#) were not considered for this study.

Using θ_e , the transformed angle θ_T can be calculated using [Equation 6.8](#).

$$\theta_T = \begin{cases} \theta + \pi - \theta_e/2 & \text{if } 0 \leq \theta < \pi + \theta_e/2 \\ \theta - \pi - \theta_e/2 & \text{if } \pi + \theta_e/2 \leq \theta < 2\pi \end{cases} \quad (6.8)$$

6.2. Removing outliers

6.2.1. Data thresholding

The data from the Delfi-C³ and Delfi-PQ satellites contains outliers, which are data points that are significantly different from the rest of the data. These outliers can be caused by a variety of reasons, such as sensor malfunction, data transmission errors. From a preliminary analysis of the datasets, temperatures outside the probable feasible range were found. A simple thresholding process was used to remove data outside a

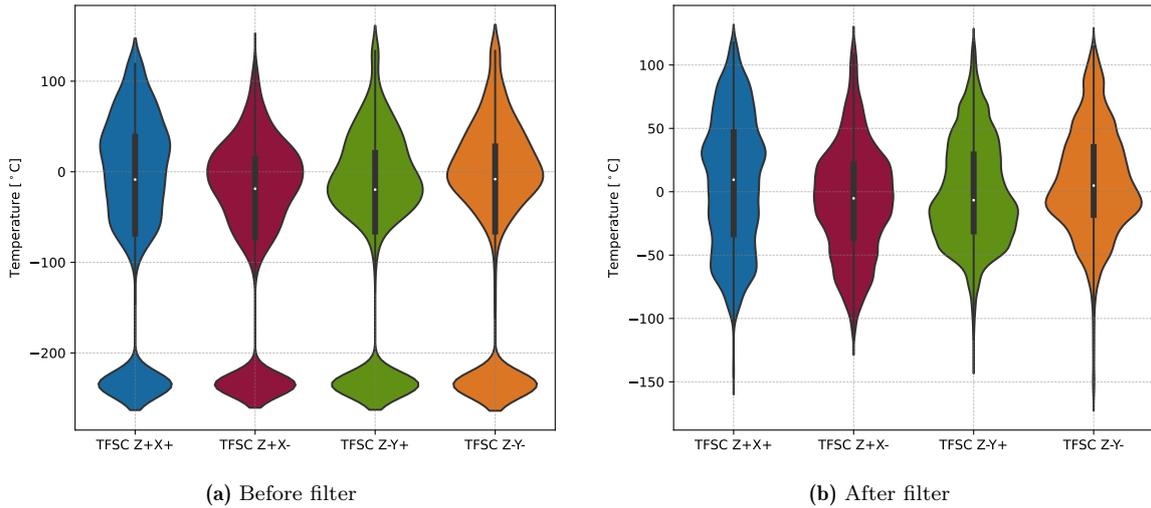


Figure 6.2: Delfi-C³ TFSC temperatures before and after thresholding for telemetry data in 2012

defined valid range.

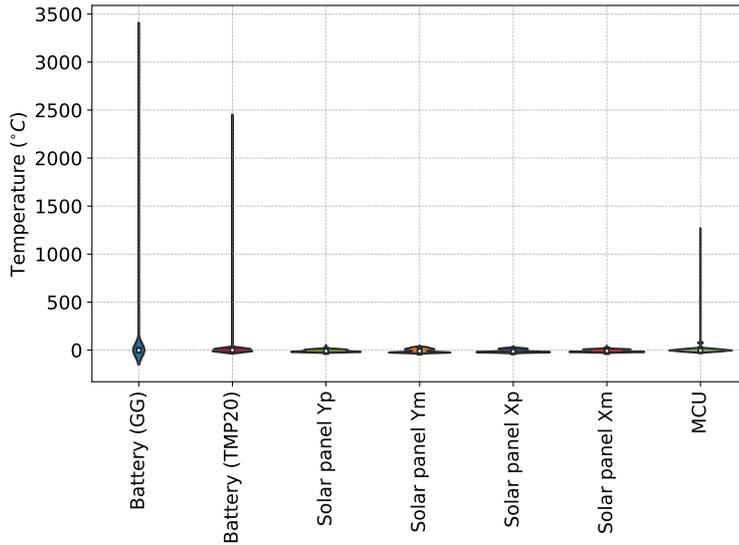
For Delfi-C³, it was observed that, for the OBC and RAP-1 temperatures, the valid data was always between -50°C and 50°C . Thus, the data outside this range was removed from the dataset. For the TFSC temperatures, the sensor used had a saturation temperature of 119.195°C , and defaulted to -235°C when no data was available. Thus, a threshold of -200°C to 118°C was used for the TFSC temperatures. Figure 6.2 depicts the TFSC temperatures before and after thresholding for telemetry data in 2012. The large cluster of data points at -235°C that are present in Figure 6.2a are removed in Figure 6.2b.

For Delfi-PQ, it was observed that the valid data was always between -40°C and 60°C . The valid range was determined by visually inspecting the temperatures, and iteratively removing clusters of non-feasible data.

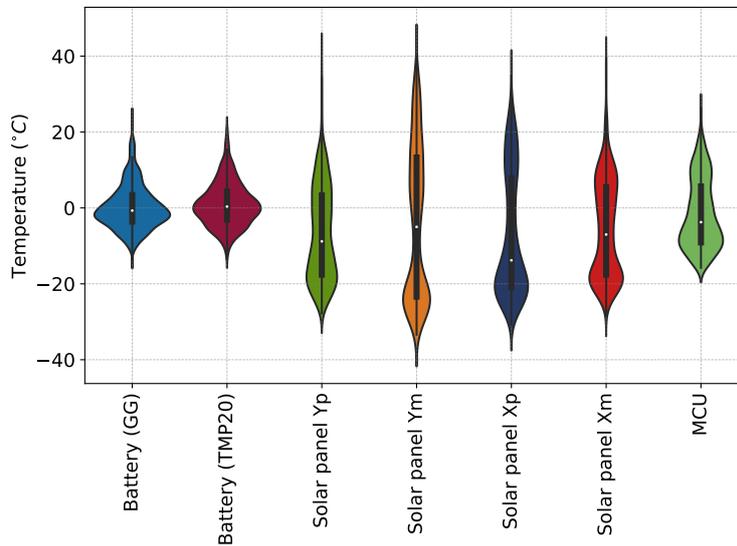
6.2.2. Time correction

Delfi-C³ is powered solely by the solar panels and does not have an onboard battery. Consequently, the satellite is only operational in the illumination section of its orbit. The raw data from Delfi-C³, after the orbit overlaying process was done as described in Section 6.1, showed that there was some data from the satellite in the eclipse section. Additionally, there was a seasonal shift observed in the data. It was determined that the satellite epoch time was reported in Dutch local time, which is UTC+2 in the summer and UTC+1 in the winter. The data was converted to UTC by subtracting 2 hours from the epoch time in the summer and 1 hour in the winter.

The telemetry still contained sections of data in the eclipse section. A closer examination revealed that the anomalies were primarily reported by a small number of ground stations. The timestamp attached to the telemetry data was the transmission time of the data from the ground station to the Delfi-C³ team, rather than the reception time of the telemetry frame. With some ground stations having a large time delay between receiving



(a) Before filter



(b) After filter

Figure 6.3: Delfi-PQ temperatures before and after thresholding

the telemetry frame and transmitting it to the Delfi-C³ team, the timestamp of the telemetry frame had to be examined closely. The delay was also found to be variable, making the correction process more difficult. Thus, a correction based on the ground station field of view was used.

The locations of the ground stations are stored as a six letter Maidenhead location code. The range of latitudes and longitudes of the satellite visible to the ground station can be calculated using [Equation 6.9](#). A tolerance of $\pm 3^\circ$ was used for the latitude and longitude range to account for the fact that the Maidenhead location code provides a rough estimate of the ground station location and the altitude of the satellite is not constant.

$$\Delta\phi, \Delta\psi = \phi, \psi \pm \sin^{-1} \left(\frac{R_e}{R_e + h} \right) \quad (6.9)$$

In Equation 6.9 $\Delta\phi$ and $\Delta\psi$ are the range of latitudes and longitudes respectively of the satellite visible to the ground station, ϕ and ψ are the latitude and longitude of the ground station, R_e is the radius of the Earth, and h is the altitude of the satellite.

The satellite latitude and longitude at each timestamp of the telemetry frame was checked against the visible range of the ground station. If the satellite was not in the visible range of the ground station, the telemetry frame was discarded. Figure 6.4 depicts the effect of this filtering process on the telemetry data. It can be observed that before filtering, some telemetry data for $\theta_T > 240^\circ$ is present. This is not possible due to the satellite being in the eclipse section of its orbit for these values of θ_T . After filtering, the telemetry data for $\theta_T > 240^\circ$ is removed, without an explicit check for the value of θ_T . Some data in the illuminated sections of the orbit were also removed, which would not be possible if a simple threshold on θ_T was used, demonstrating the robustness of this method.

No time correction was done for Delfi-PQ: as the location data of the ground station was not used for this study, and for FUNcube-1: as it was not necessary.

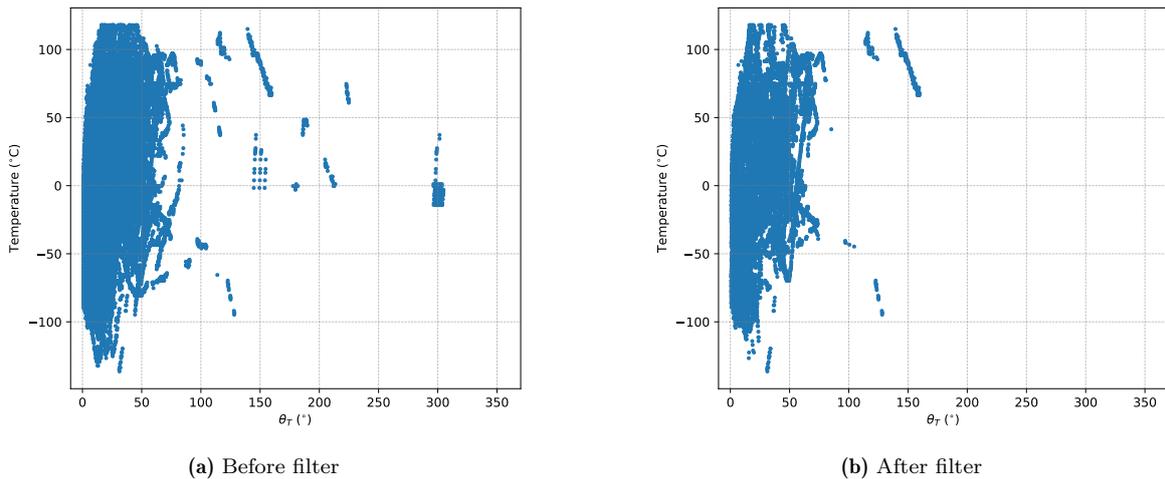


Figure 6.4: Delfi-C³ before and after field of view filtering for TFSC Z+X+ telemetry data in 2012

6.2.3. Constant section removal for FUNcube-1

FUNcube-1 telemetry data contains sections where the temperature sensors report a constant value, as shown in Figure 6.5. This issue was also mentioned by the FUNcube-1 team when providing the data for this study. These sections were removed from the dataset using a simple filter, checking for constant temperature values for three consecutive timestamps. If the temperature values were constant for three consecutive timestamps, all the previous and subsequent constant temperature values, except the first, were removed from the dataset. Figure 6.6 shows the FUNcube-1 black chassis temperature before and after the constant section removal.

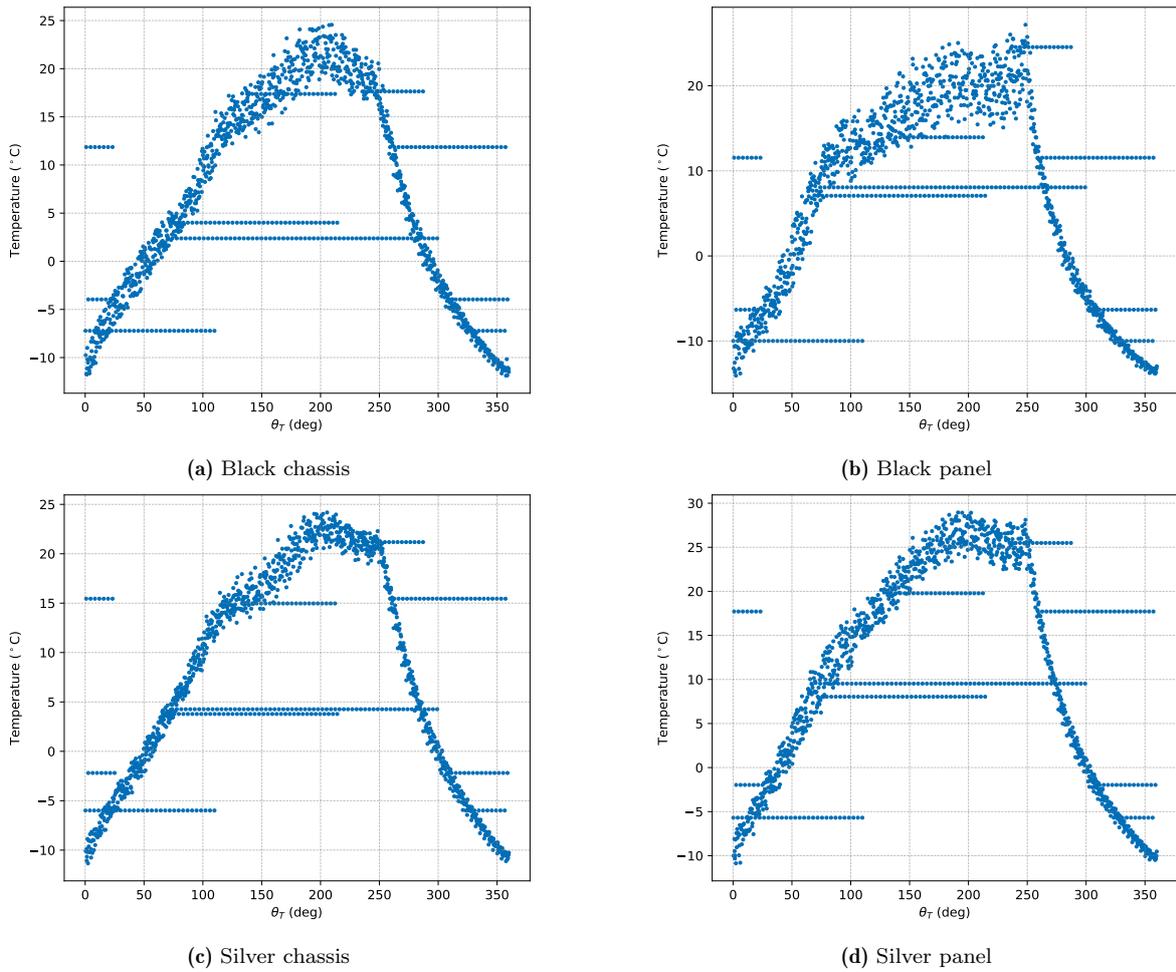


Figure 6.5: FUNcube-1 raw chassis and panel temperatures

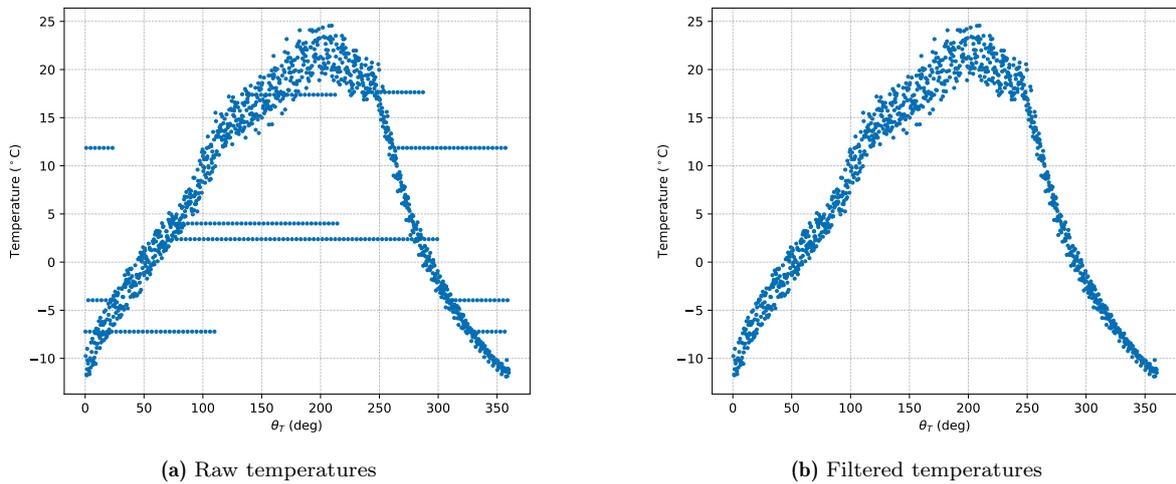


Figure 6.6: FUNcube-1 black chassis temperature before and after constant section removal

6.2.4. Hampel filter

The thresholding process described in [Section 6.2.1](#) removes obvious outliers from the dataset. Outliers that lie within the threshold range, but are anomalous compared to

local data, are not removed by the thresholding process. These outliers can be removed using the Hampel filter [43], which is a robust method for removing outliers from a dataset. The Hampel filter is a three-step process, which is described in this section.

The first step of the Hampel filter is calculating the median of the dataset. Since the temperature of the satellite varies with the satellite's position in its orbit, the median is calculated by binning the data into bins of 5° and calculating the median of each bin. A bin is considered valid if it contains at least 10 data points. The bin size and the minimum number of data points per bin were chosen based on manual inspection of the data.

The amount of solar radiation flux received by the satellites can also be expected to have an influence on the temperatures of the satellites. Thus, the first step of the Hampel filter is also performed by binning the data per three months.

The second step of the Hampel filter is calculating the median absolute deviation (MAD) of the dataset, done using Equation 6.10.

$$\text{MAD} = \text{median}(|x_i - \text{median}|) \quad (6.10)$$

In Equation 6.10, x_i is the i th value of the data in the bin. The MAD is calculated for each bin of the dataset.

The third step of the Hampel filter is calculating the Hampel identifier (HI), a measure of how far each value of the dataset is from the median, of the dataset. The HI is calculated using Equation 6.11.

$$\text{HI} = \frac{x_i - \text{median}}{1.4826 \cdot \text{MAD}} \quad (6.11)$$

In Equation 6.11, x_i is the i th value of the dataset. The factor of 1.4826 is used to make the MAD equivalent to the standard deviation of a normally distributed dataset. MAD is used instead of directly using the standard deviation because the MAD is a robust measure of dispersion, not sensitive to outliers unlike the standard deviation.

Removing the outliers is the final step of the Hampel filter. The outliers are the values of the dataset for which the HI is greater than a threshold value. The threshold value can be changed to remove more or less outliers from the dataset. The threshold value used for this study was 3, which is equivalent to removing data points that are more than 3 standard deviations away (assuming a normal distribution) from the median of the bin. Figure 6.7 depicts the effect of the Hampel filter applied to the Delfi-PQ solar panel Y_p temperatures. All the outliers between $150^\circ < \theta_T < 250^\circ$ are removed by the Hampel filter. The filtering is not perfect, as some outliers remain in the dataset, especially some data points from the cluster at $\theta_T \approx 100^\circ$. However, the Hampel filter is able to remove most of the outliers from the dataset. This filter was considered sufficient for the purposes of the machine learning models developed in this thesis, and no further filtering was performed.

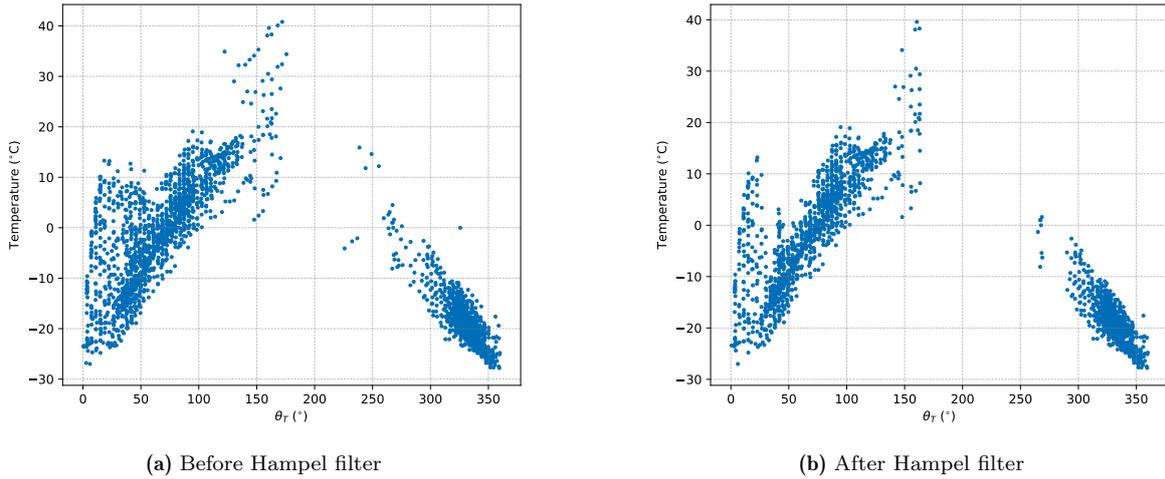


Figure 6.7: Hampel filter applied to Delfi-PQ solar panel Yp temperatures for the period of 2022-01-18 to 2022-03-29

6.3. Training and testing split

The train-test split is an important concept in creating machine learning models that involves dividing the dataset into two subsets: the training set, which is used by the model to learn relationships in the data, and the test set, which is used to evaluate the model's performance.

The test set serves as a never-before-seen dataset that the model has not encountered during training. Evaluating the model's performance on the test set can help us understand the effectiveness of the model to generalize to new, unseen data.

To simulate the requirements of a real space mission, where it might be necessary to employ existing data to train the models to predict future behavior, a sequential split was used instead of a random split. The satellite degradation can significant deviations in the temperatures of the satellite over time. Thus, a sequential split could also be used to test the robustness of the models to adapt to the varying satellite behavior.

After outlier removal, the first 80% of the total available dataset was used as the training set, with the remaining 20% being used as the test set to evaluate the prediction performance of the models.

6.4. Data normalization

Data normalization, also known as feature scaling or standardization, is an essential preprocessing step in machine learning. It involves transforming the numerical features of a dataset to a common scale or range.

When features in a dataset have different scales or measurement units, it can negatively affect the performance of optimization algorithms used to train neural networks. Features

with larger scales can dominate the learning process and lead to biased results [36]. Furthermore, optimization algorithms often have greater efficiency and better convergence when the features are of a similar size [19]. Normalization of data ensures that the optimization process is not hindered by features with widely different ranges, resulting in a more stable and reliable model training process.

For this study, a simple scaling normalization, described in Equation 6.12, was used to normalize the data. The minimum and maximum values of the dataset used to scale the data were calculated only on the training set to avoid data leakage from the test set.

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (6.12)$$

6.5. Data selection

After using the filtering process described in Section 6.2.1, Section 6.2.2 and Section 6.2.4, the Delfi-C³ telemetry data was still found to be noisy. This is hypothesized to be due to the low thermal mass of the TFSCs, causing large temperature fluctuations due to tumbling. Since Delfi-C³ no data from the attitude sensors, it proved to be difficult to compensate for the tumbling. A possible solution to this problem would be to use the mean temperature of all the TFSCs as the input to the model, instead of using the temperatures of each panel separately, resulting in the fluctuations due to tumbling being averaged out, as depicted in Figure 6.8. It was observed that the mean temperature of the TFSCs, while still containing some noise, followed a similar trend to the temperatures from Delfi-PQ and FUNcube-1 data. Due to the low thermal mass of the TFSCs, the temperatures also begin to saturate at smaller values of θ_T compared to Delfi-PQ and FUNcube-1.

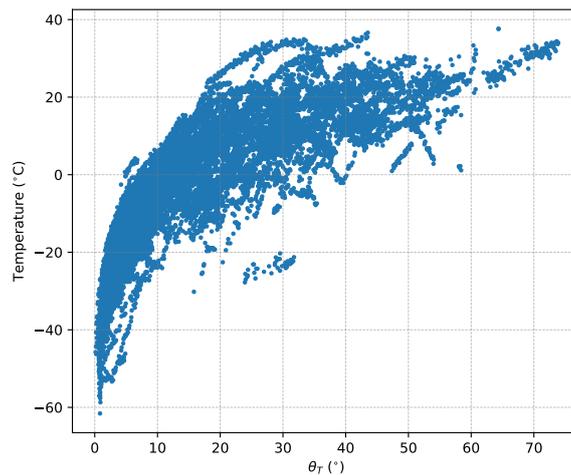


Figure 6.8: Mean TFSC temperature of Delfi-C³ for 2012

The averaging process, however, reduces the available payload temperature channels from

four to one. Thus, the Delfi-C³ data was not used for the analysis in this study. The limitation of no data availability in the eclipse phase, making it difficult to model the cooling cycle, was also a factor in not using the Delfi-C³ data.

Telemetry from FUNcube-1 is available for the whole orbit. Thus, the FUNcube-1 telemetry data was used to validate the performance of purely data-driven models, considered as the baseline models in this study. The Delfi-PQ telemetry, being sparse and having sections of orbit with no data, provides a challenge for purely data-driven models. This was used to demonstrate the limitations of data-driven models when no fitting data is available, and to demonstrate the advantages of using a physics-based model to fill in the gaps in the data.

7

Data-driven models

Purely data-driven models serve as a preliminary reference point for this study, to understand the potential accuracy of temperature predictions under ideal circumstances. The thermal modeling can be formulated as a time series prediction problem, where the primary objective is to forecast future temperatures based on past measurements within a defined input window size w .

Upon examining the telemetry data, it becomes evident that for a given position in the orbit (maintaining the same θ_T), the temperatures tend to be similar. As outlined in the state-of-art, LSTM based neural networks have demonstrated to be effective in time series prediction applications.

By using LSTM networks, which account for temporal dependencies in the data, the models developed are potentially well-suited for predicting future temperature values based on past observations. However, it's also important to remember that these models are purely data-driven, and they can be more challenging to interpret due to their "black box" nature.

7.1. FUNcube-1 temperature prediction

The initial data-driven models were developed utilizing FUNcube-1 data, owing to the availability of full-orbit telemetry. The first phase of model training was focused solely on predicting the temperature of the black panel. This panel was selected due to its significant temperature fluctuations induced by tumbling, thus offering a challenging scenario for the models to address.

7.1.1. Temperature-based models

The black panel exhibits a steady thermal behavior with no abrupt temperature shifts under standard conditions. This stability potentially allows for future temperature

Table 7.1: LSTM model architecture

Layer	Number of cells
Input layer	Based on input window size
Hidden layer 1	256
Hidden layer 2	256
Hidden layer 3	128
Hidden layer 4	128
Hidden layer 5	64
Hidden layer 6	32
Hidden layer 7	16
Output layer	Based on the number of output channels

prediction based purely on prior temperature readings. Models trained on a range of input step sizes, from one minute to 120 minutes, were tested. Since the telemetry available is sampled at one minute intervals, the size of the input window in minutes directly translates into the number of temperature readings within the input window.

[Table 7.1](#) summarizes the number of layers and LSTM cells in each layer for the models used in this study. The networks were sized conservatively, most likely being larger than necessary, to establish the possibility of the models to achieve the desired accuracy.

Since a large number of models were evaluated during the study, a short training time was advantageous in rapid prototyping and iterations. Therefore, the selection of activation functions was confined to the default tanh and sigmoid functions described in [Equation 2.2](#) and [Equation 2.3](#) respectively. This constraint enabled running the models on a GPU using the TensorFlow and cuDNN frameworks.

The models were trained using the Adam optimizer [31] with a learning rate of 10^{-4} . The learning rate was chosen based on the smoothness of the loss curve during training. The loss function used was the Mean Squared Error (MSE) loss function, defined in [Equation 7.1](#). The MSE loss function was chosen since it is a differentiable function, which is a requirement for the Adam optimizer. It also penalizes large errors more than small errors. To prevent overfitting, the loss function value was monitored on the test set during training, which was stopped when the loss on the test set stopped decreasing.

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (7.1)$$

Two distinct behaviours were observed in the models trained. For step sizes less than five, the models were unable to capture the temperature fluctuations due to tumbling. For a small section of step sizes, the models were able to capture the temperature fluctuations. As the step sizes increased, the models tended to smooth out the temperature fluctuations, predicting the average temperature in the training data.

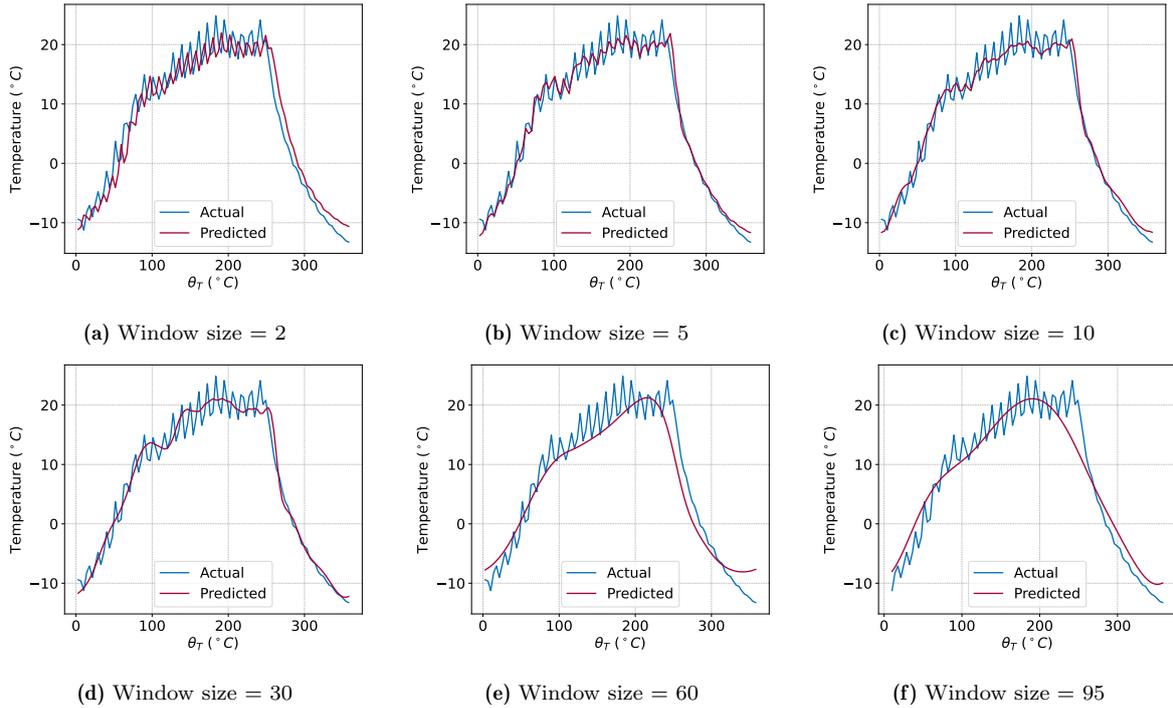


Figure 7.1: Predicted and actual black panel temperature for FUNcube-1 (2016-02-04 20:52:00 to 2016-02-04 22:28:00) using a temperature-based LSTM model

The aforementioned models excel at predicting temperatures under nominal conditions. Nevertheless, they present certain limitations during anomalous data segments, particularly when the sensors relay constant temperature readings. For small step sizes, model predictions closely mirror the anomalous constant temperature, yielding an error margin nearly equal to the mean error on the test set. This small error may complicate anomaly detection. Conversely, for larger step sizes, due to the broader input window, the model predictions respond more slowly to the anomaly, generating considerable error in the initial phase. This also results in a prolonged recovery period from the anomaly, maintaining significant errors even after the anomaly has concluded. A larger input window would facilitate anomaly detection, but concurrently hinder the precise isolation of the anomalous section. This behavior of the temperature-based LSTM models is illustrated in [Figure 7.2](#), with error statistics summarized in [Table 7.2](#).

Table 7.2: Prediction error in black panel temperature for FUNcube-1 during anomalous orbit (2016-02-04 07:56:00 to 2016-02-04 09:30:00) using temperature based LSTM model

Step size	Error μ ($^{\circ}C$)			Error σ ($^{\circ}C$)		
	Before anomaly	During anomaly	After anomaly	Before anomaly	During anomaly	After anomaly
2	-0.03	-0.42	-0.88	3.03	0.27	3.74
10	-0.04	-0.12	-0.63	1.89	0.93	3.46
60	1.37	-5.54	-8.03	4.03	1.53	12.22

A possible solution to this problem is to decouple the predicted temperatures from the

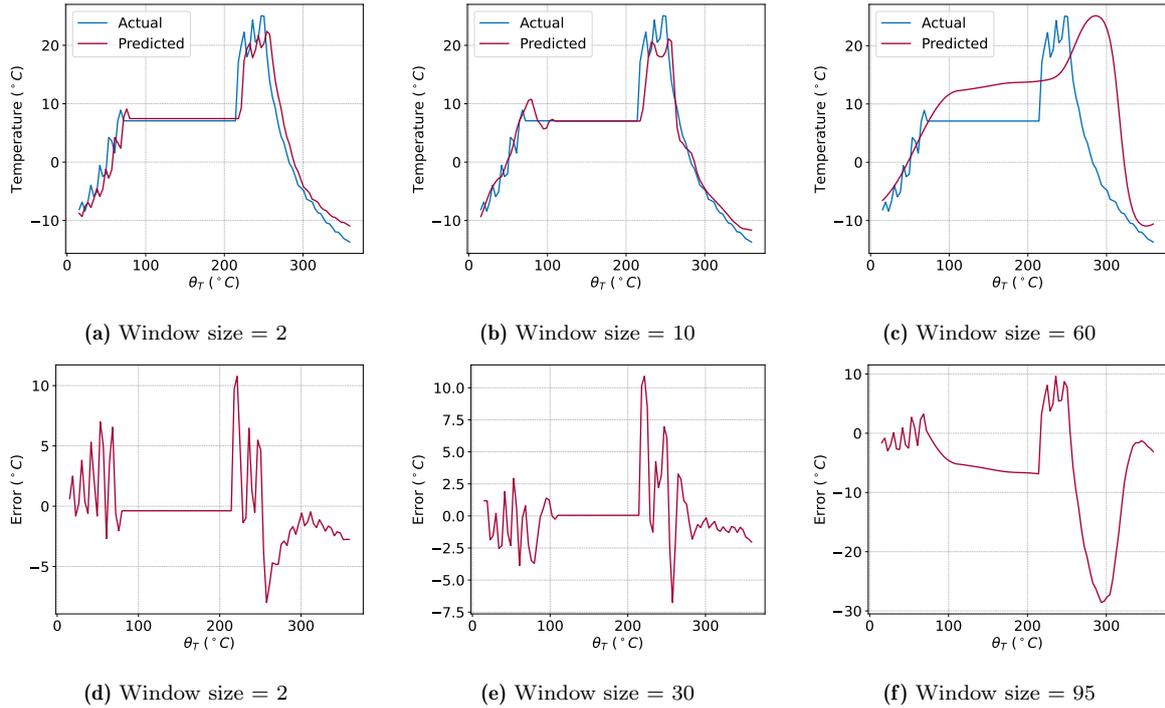


Figure 7.2: 7.2a, 7.2b, 7.2c: Predicted and actual black panel temperature for FUNcube-1 during anomaly (2016-02-04 07:56:00 to 2016-02-04 09:30:00) using temperature-based LSTM model 7.2d, 7.2e, 7.2f: Difference between actual and predicted black panel temperature for FUNcube-1 during anomaly

previous measured temperatures. Thus, the anomalies would not affect the predicted temperatures, allowing the detection and isolation of anomalous sections of temperature data. Such models are discussed in the next section.

7.1.2. Illumination-based models

One-channel temperature prediction

As outlined in Section 7.1.1, the temperature-based models are susceptible to sensor anomalies, demonstrating limitations in simultaneously detecting and isolating anomalous data segments. The relatively constrained range in the temperature data after orbit overlaying (as shown in Figure 6.6b for the black chassis temperatures) indicates the potential for predicting temperatures based on illumination data. Acting as a stand-in for the satellite's orbital position, the illumination data can be readily inferred from solar panel voltages. Moreover, solar radiation is a principal contributor to the satellite's temperature, reinforcing the potential utility of this approach.

The illumination data was considered as a binary input, being one if the satellite was in sunlight, and zero if the satellite was in eclipse. The models maintained identical architecture, as well as the same training and testing dataset, as the temperature-based models highlighted in Section 7.1.1. The requisite minimum time steps to resolve the temperature data accurately was found to correspond to the duration of the satellite's illuminated orbital phase. The illuminated phase represents approximately 70% of the

orbital period, and for step sizes smaller than the illuminated phase, a fraction of the illuminated phase will exhibit identical input illumination data. Consequently, the model regresses to the mean temperature of this section rather than the actual temperature, as illustrated in Figure 7.3. This phenomenon can also be observed in the fluctuation of the prediction error standard deviation, represented in Figure 7.4. The error standard deviation remains high for smaller step sizes, decreasing as the step size increases to match the duration of the illuminated phase. For step sizes exceeding the illuminated phase, the error standard deviation stabilizes and remains almost constant.

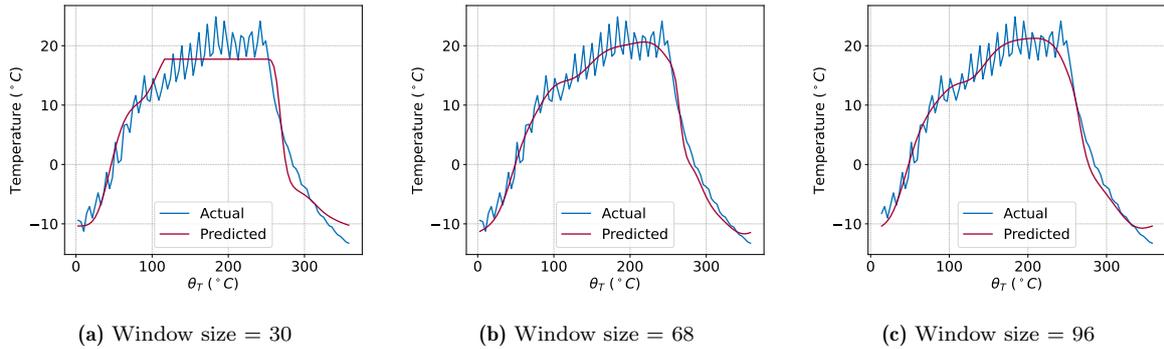


Figure 7.3: Predicted and actual black panel temperature for FUNcube-1 using an illumination-based LSTM model

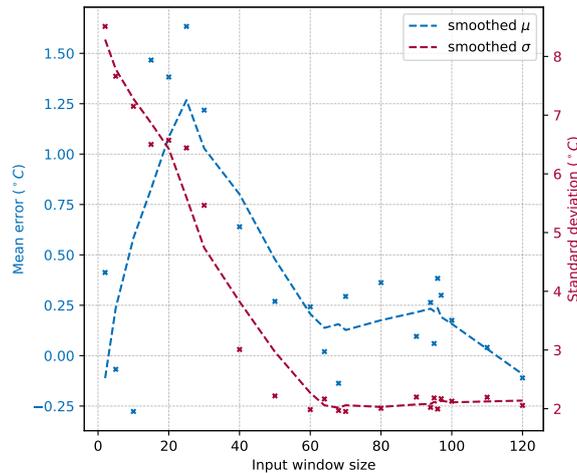


Figure 7.4: FUNcube-1 black panel temperature prediction error statistics for different window sizes

The illumination model regresses to the mean temperature trend of the overlaid orbital temperature data, failing to account for temperature variations caused by tumbling. To reincorporate the temperature measurements into the model, while maintaining the benefits of illumination-based models, a residual network was developed. This network adjusts the base illumination model prediction to more precisely correspond with measured temperatures.

Given that the network is founded on LSTM cells and the inputs are sequentially fed into the model, the input layer remains unaltered across all step sizes. Given the absence of

drawbacks associated with increasing the input step size for the base model, all subsequent analyses adopted an input step size of 96, approximately equating to one orbit.

The residual model utilizes a compact input window comprising temperature measurements and the base illumination model's predicted temperature for the subsequent time step. It then generates a corrective prediction for the base model. An MLP was chosen for the residual model, instead of an LSTM network, as the input window size, varying from one to ten time steps, was relatively small in comparison to the networks examined in [Section 7.1.1](#). The number of hidden layers and nodes in each layer was identical to those detailed in [Table 7.1](#), but the LSTM cells were supplanted with dense feed-forward neurons. The activation function for the hidden layers was the Rectified Linear Unit (ReLU) function, as described in [Equation 7.2](#), while a linear activation function was applied for the output layer. The schematic of the combined model is depicted in [Figure 7.5](#)

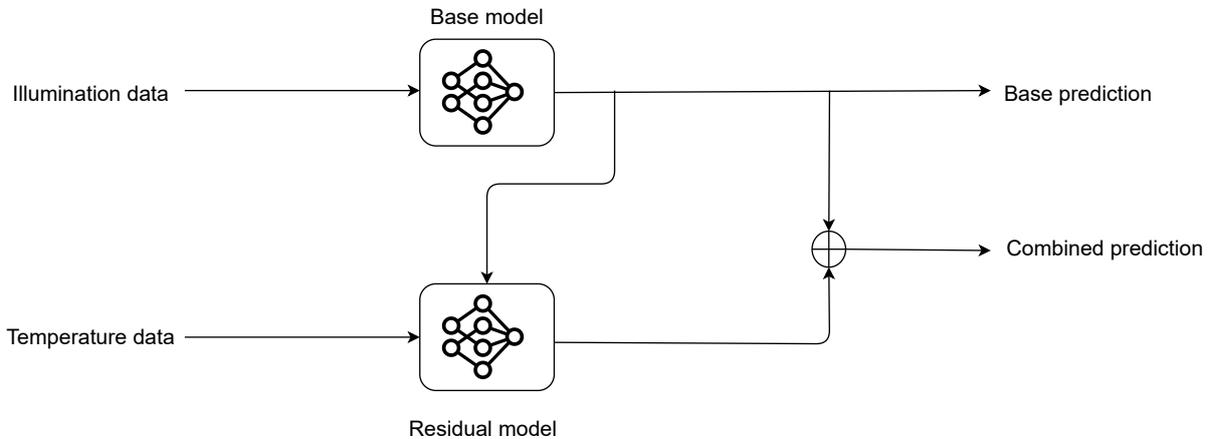


Figure 7.5: Schematic of the combined illumination-based and temperature-based residual models

$$f(x) = \max(0, x) \quad (7.2)$$

[Figure 7.6](#) depicts the performance of the combined model for residual model input window sizes of 2, 4 and 8. For small step sizes, the model is unable to resolve the temperature variations due to tumbling, and the prediction error is similar to the base model. The prediction error reduces with an increase in input step size. For step sizes approximately 5, the improvement in prediction error is small. This trend can also be seen in the prediction error statistics depicted in [Figure 7.7](#).

As illustrated in [Figure 7.8](#), the combined model exhibits notable performance in predicting the anomalous temperature segment. Because the base model does not consider temperature measurements, unlike the models discussed in [Section 7.1.1](#), the abnormal constant temperature measurements do not influence the base model's prediction. Due to the temperature inputs used by the residual model, the combined model aligns closely with the anomalous sensor readings.

Although the prediction error of the combined model is less than that of the base model, it is still higher than the prediction error for the non-anomalous part of the orbit. Additionally, the combined model's prediction error rapidly reverts to non-anomalous prediction error levels as soon as the anomalous section concludes, owing to the relatively

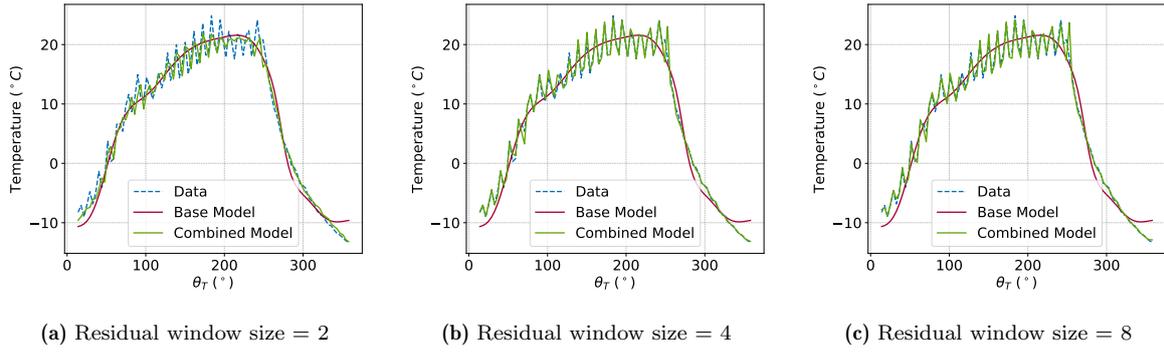


Figure 7.6: Predicted and actual black panel temperature for FUNcube-1 using illumination-based LSTM model with residual network

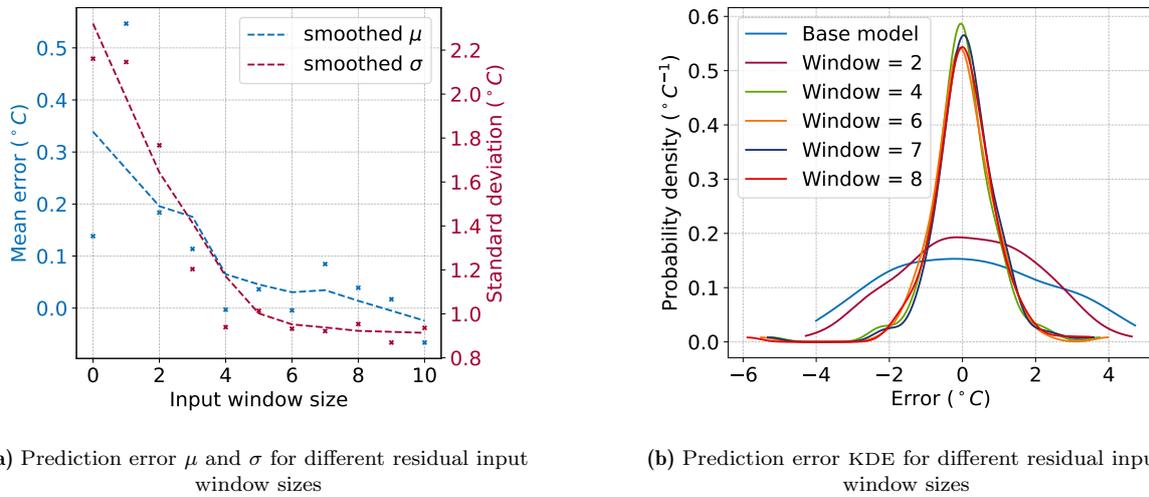


Figure 7.7: FUNcube-1 black panel temperature prediction error statistics for different residual input window sizes

small input window size of the residual model. Therefore, by comparing the prediction errors of the base and combined models, the anomalous section can be identified and distinguished from the non-anomalous section.

Multi-channel temperature prediction

The models detailed in the preceding sections were all trained to forecast only one of the eight accessible temperature channels in the FUNcube-1 telemetry data. Ideally, a single model would be employed to predict all eight temperature channels, as the model would have the potential to learn the thermal couplings between the different sections of the satellite. To evaluate the possibility of concurrently predicting all temperature channels, an illumination-based model with the architecture described in Table 7.1 was trained to predict all eight temperature channels. The base model employed an input window size of 96.

The measured and predicted temperatures for all eight channels are portrayed in Figure 7.9. The model manages to predict the temperature of all eight channels with a prediction error analogous to the base model mentioned above, which predicted solely black panel temperatures. Given that the model can predict all eight temperature channels with the

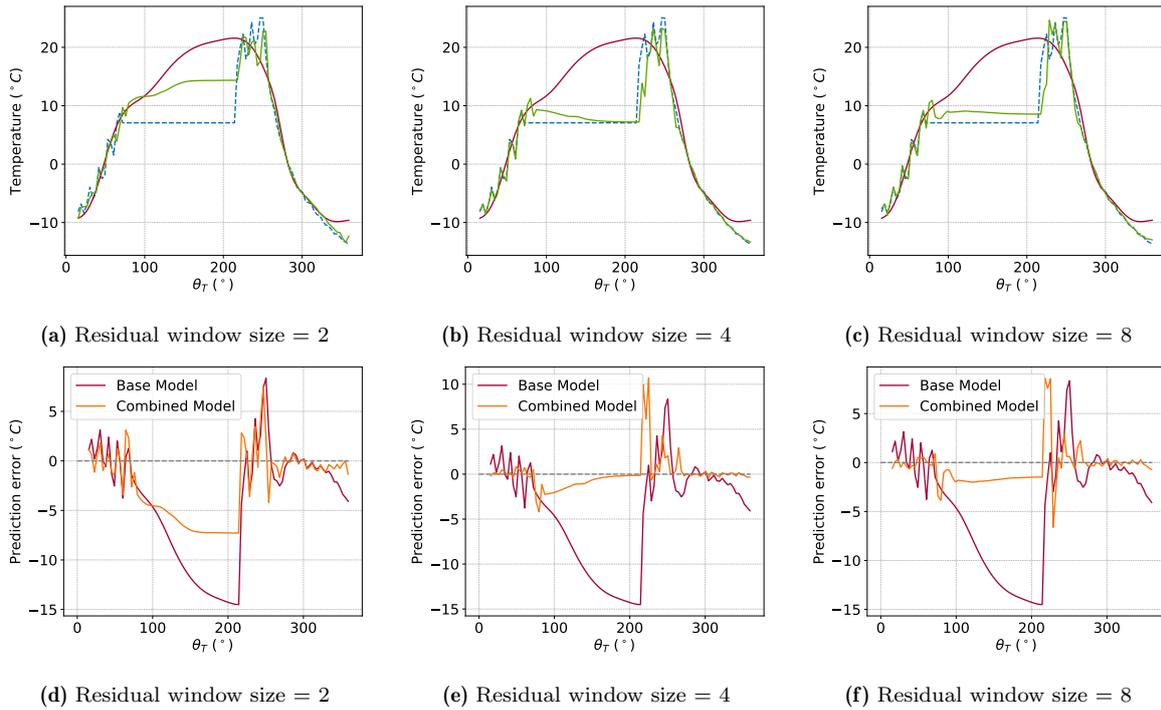


Figure 7.8: 7.8a, 7.8b, 7.8c: Predicted and actual black panel temperature for FUNcube-1 during anomaly (2016-02-04 07:56:00 to 2016-02-04 09:30:00) using an illumination-based LSTM model with an input window size of 96 time steps and temperature-based residual model
 7.8a, 7.8b, 7.8c: Difference between actual and predicted black panel temperature by the base and combined models for FUNcube-1 during anomaly

same architecture as the model forecasting just one temperature channel, it is plausible that the size of the single-channel model could be reduced.

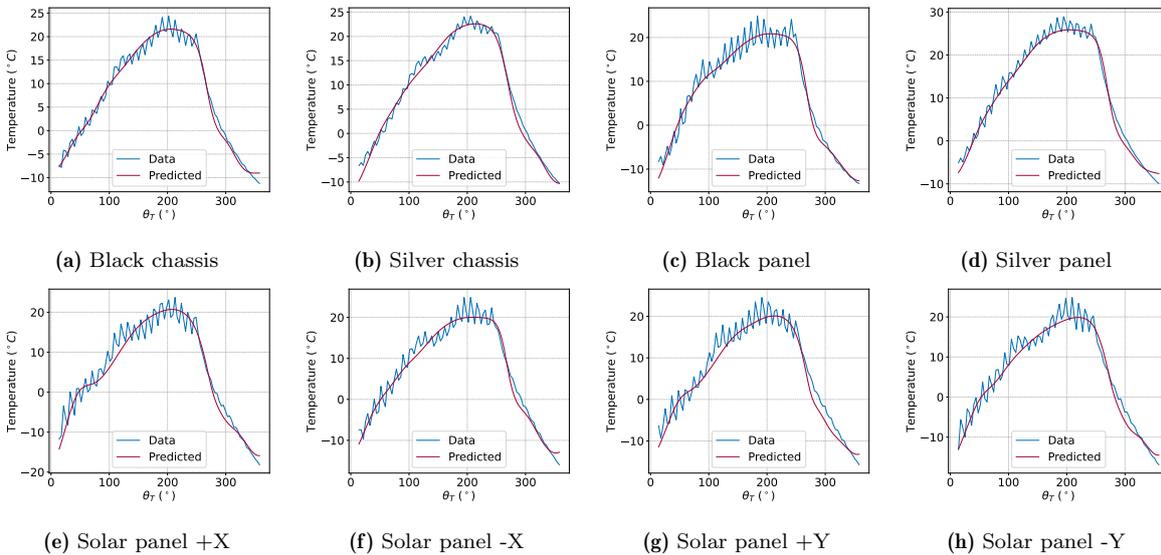


Figure 7.9: Measured and predicted temperatures of all 8 temperature channels in FUNcube-1 telemetry data using an illumination-based LSTM model with an input window size of 96 time steps

The residual models similar to the residual models for the one channel base model were

trained for the eight channel base model, for input window sizes ranging from one to ten time steps. The error statistics for the combined models for different input window sizes are depicted in Figure 7.10. The reduction in standard deviation of the prediction errors with increasing input window size is small after an initial reduction compared to the base model.

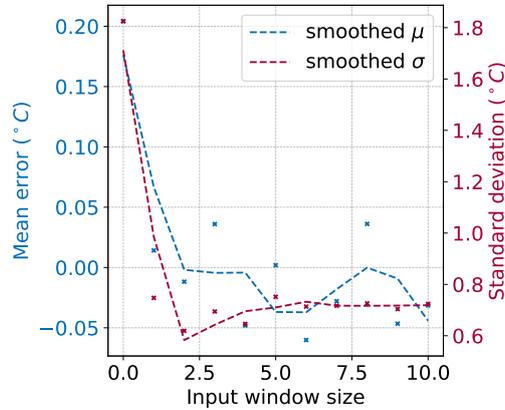


Figure 7.10: Prediction error μ and σ for different residual input window sizes of the combined model for all 8 temperature channels in FUNcube-1 telemetry data

The residual models were observed to capture the temperature fluctuations due to tumbling at smaller time steps compared to one channel models. This is possibly due to information present in the additional temperature channels. For example, it can be expected that the temperatures of opposing solar panels have opposite deviation from the base model prediction. Thus, with even a single time step of temperatures, the residual model is able to predict the deviation from the base model due to tumbling. The predictions of the combined model with input window sizes for the residual models of one and five time steps are depicted in Figure 7.11 and Figure 7.12 respectively, from which it can be observed that the predictions are very similar.

Similar to the one channel model, for an anomalous section, the base model can be used to identify and isolate the anomalous section. The combined model predictions follow the anomalous measurements more closely due to the input anomalous temperature values to the residual model. The prediction error of the combined model, however, is still higher for the anomalous section compared to the non-anomalous. The error statistics for the combined models are shown in Table 7.3.

Table 7.3: Prediction error (all channels) for FUNcube-1 during anomalous orbit (2016-02-04 07:56:00 to 2016-02-04 09:30:00) using combined illumination-based model ($w = 96$) and residual models

Residual window size	Error μ (°C)			Error σ (°C)		
	Before anomaly	During anomaly	After anomaly	Before anomaly	During anomaly	After anomaly
0 (Base model)	0.39	-11.25	-0.14	1.99	5.42	1.42
2	0.05	-0.08	0.4	0.44	2.02	2.72
5	0.01	-0.86	0.17	0.37	1.98	2.88
10	0.11	-0.81	-0.23	0.46	2.60	2.9

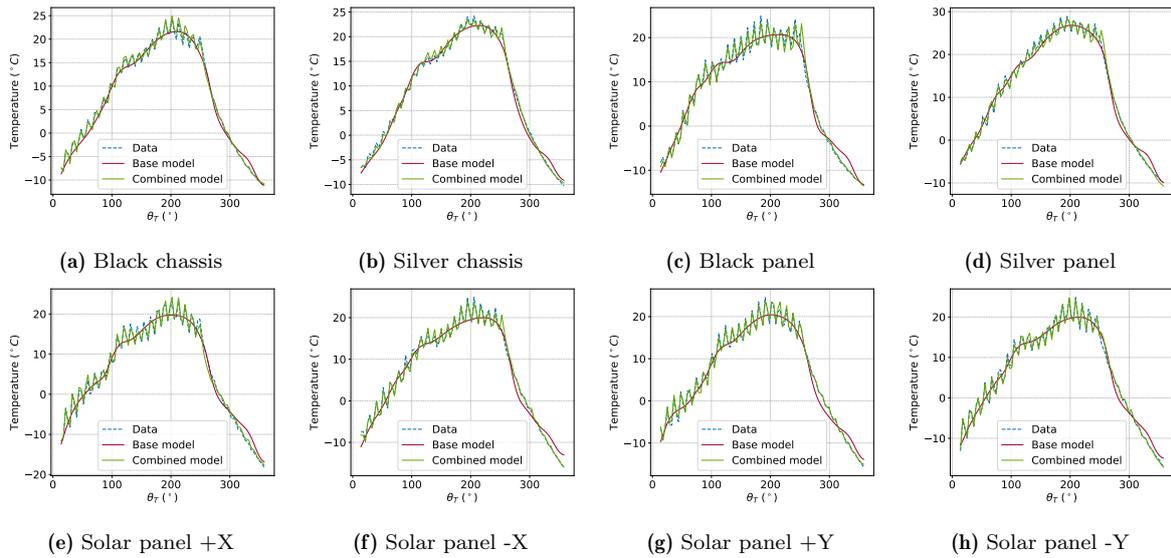


Figure 7.11: Measured and predicted temperatures of all 8 temperature channels in FUNcube-1 telemetry data using an illumination-based LSTM model with an input window size of 96 time steps and a residual model with an input window size of 1 timestep

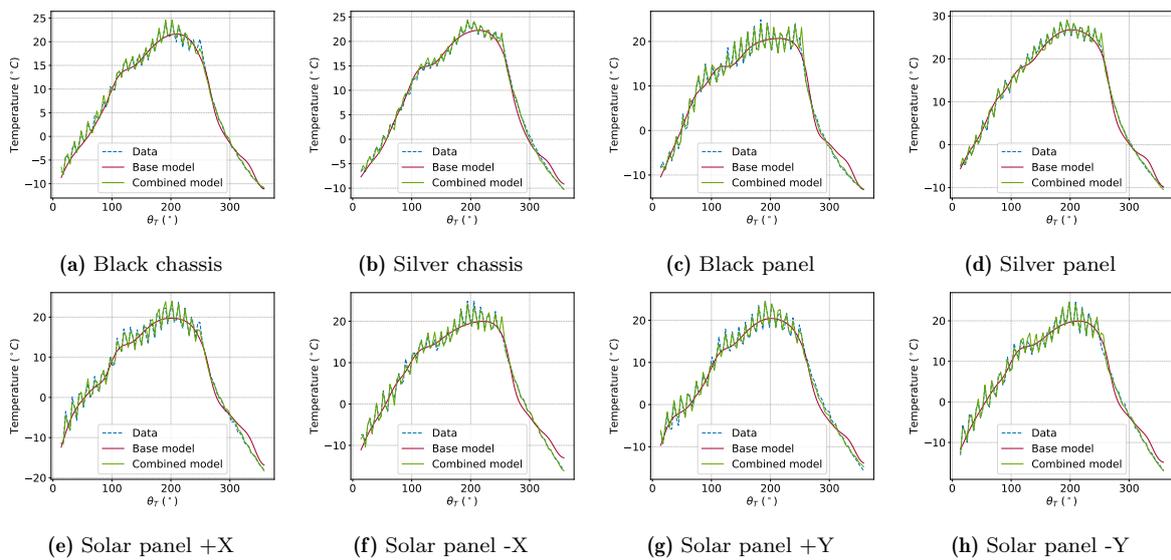


Figure 7.12: Measured and predicted temperatures of all 8 temperature channels in FUNcube-1 telemetry data using an illumination-based LSTM model with an input window size of 96 time steps and a residual model with an input window size of 5 timesteps

Several models with different input data have been described above, with the key results summarized in [Table 7.4](#)

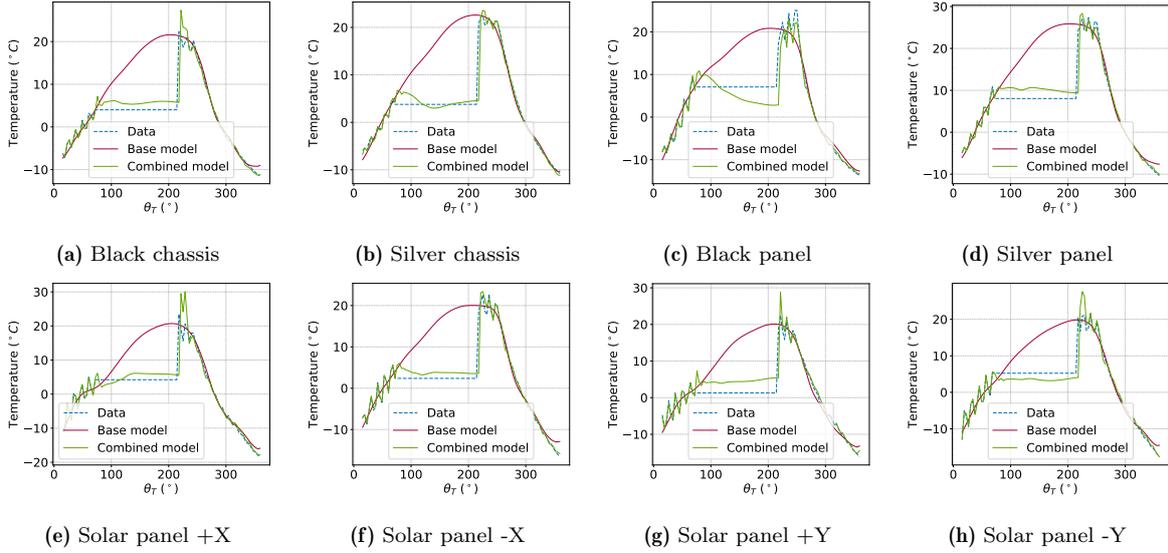


Figure 7.13: Predicted and actual temperatures of all 8 temperature channels in FUNcube-1 telemetry data during anomaly (2016-02-04 07:56:00 to 2016-02-04 09:30:00) using an illumination-based LSTM model with an input window size of 96 time steps and a residual model with an input window size of 5 timesteps

Table 7.4: Summary of FUNcube-1 models

Model	Error μ ($^{\circ}C$)	Error σ ($^{\circ}C$)
Temperature-based model ($w = 96$)	-0.13	2.9
Illumination-based model ($w = 96$)	0.53	1.91
Illumination-based model ($w = 96$) + temperature-based residual model ($w = 5$)	0.02	0.69
Illumination-based model ($w = 96$) + temperature-based residual model ($w = 10$)	0.06	0.76

7.2. Delfi-PQ temperature prediction

Unlike FUNcube-1, Delfi-PQ telemetry data is sparse. Telemetry data for Delfi-PQ is available only when it is over a receiving ground station, therefore the data is sparse and sections of orbit, especially over the South Pole, have no telemetry data. This makes it challenging to train a purely data-driven model for Delfi-PQ as the confidence of the model predictions for sections of orbit with no data is hard to quantify. An illumination-based model, with the same architecture as the illumination-based models described in [Section 7.1.2](#) with an input window size of 94 time steps (which roughly corresponds to one orbit) was trained to predict the six available temperature channels.

[Figure 7.14](#) shows the measured and predicted temperatures, from the illumination-based model, of the six temperature channels for the test set of Delfi-PQ. It was observed that the prediction error is larger than the FUNcube-1 model. This is possibly due to FUNcube-1 data spanning only one day, while the Delfi-PQ data spans over about one year. The thermal behavior of the satellite can be expected to change with time due to

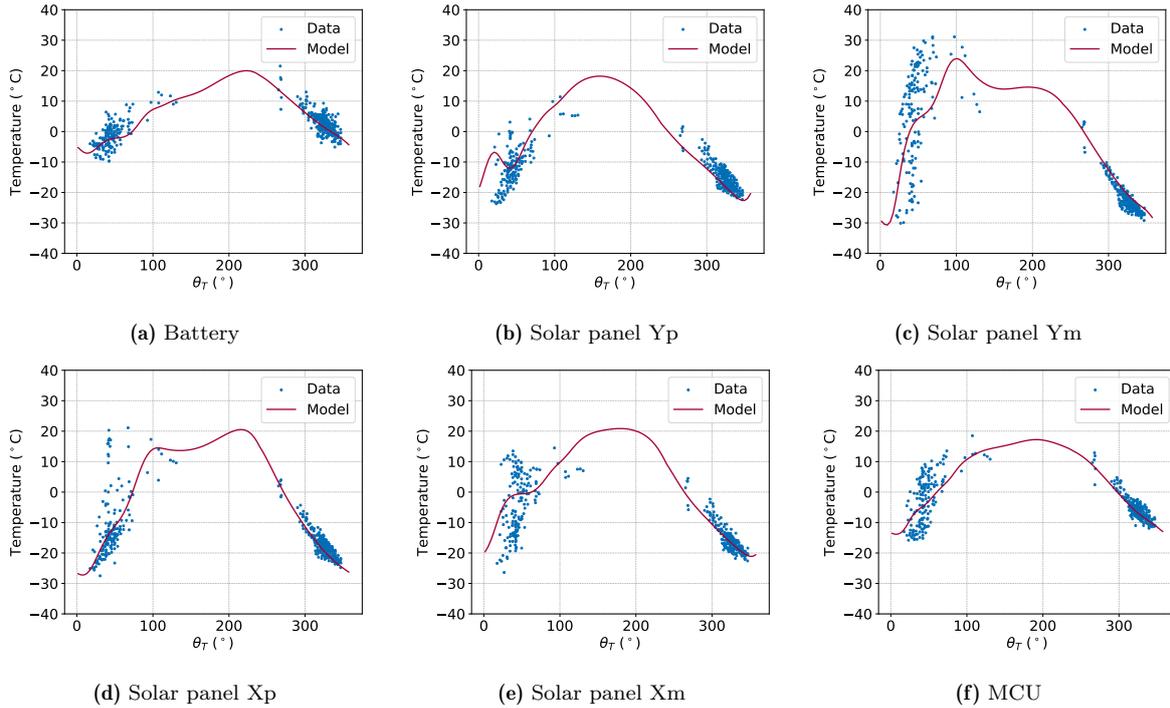


Figure 7.14: Predicted and measured temperatures of Delfi-PQ using an illumination-based LSTM model with an input window size of 94 time steps

aging and seasonal variation in the input solar flux. Combining data from 11 months to create the training set can be seen to increase the spread of the data. Since the model fits to the mean of the data, the prediction error is higher compared to the FUNcube-1 model. The KDE of the prediction errors is shown in [Figure 7.15](#).

The prediction error might potentially be reduced by constructing different base models for smaller data segments. However, this approach was not pursued in this thesis. It's evident that the absence of telemetry data for certain orbit sections impacts the predictions of the purely data-driven model. We would generally anticipate the satellite to keep warming up during the orbit's illuminated section, a pattern observable in the FUNcube-1 data. However, the model's predictions do not consistently capture this trend across all channels. From [Figure 7.14](#), the model's predictions do not accurately reflect physical conditions. This highlights the limitations of a purely data-driven model and introduces the potential of employing a hybrid, physics-informed model to enhance prediction accuracy, even when telemetry data is not available.

Temperature based residual models similar to the ones trained for the FUNcube-1 were explored for Delfi-PQ. The step sizes of the residual models had to be limited to a maximum of four time steps due to the availability of test data.

[Figure 7.16](#) shows the error statistics of the combined base and residual models of various step sizes. It can be observed that the trends in the error statistics are similar to the FUNcube-1 residual models depicted in [Figure 7.10](#), with the standard deviation of the prediction error reducing with increasing step size. The prediction error mean and standard deviations for step size of four, however, were observed to have large variation

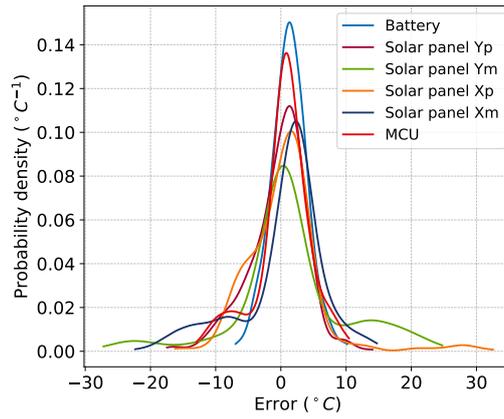


Figure 7.15: KDE of the prediction error on the test set of Delfi-PQ using an illumination-based LSTM model with an input window size of 94 time steps

between different training runs. This is possibly due to the small amount of training data available for the step size of four.

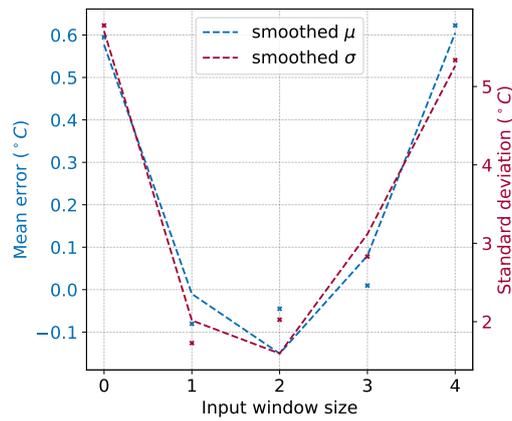


Figure 7.16: Error statistics of the combined illumination base and residual models of various step sizes for Delfi-PQ

8

First-principles models

To supplement telemetry data in sections where there is no available telemetry information, a basic first-principles thermal model of the satellite can be employed. This utilization of a first-principles model can aid in bolstering the reliability of the predictions made by the data-driven model by constraining the predictions using known physics. This section delves into two first-principles models: one is a curve-fit model built upon the lumped capacitance method, and the other is a thermal network model.

8.1. Curve-fit model

The curve-fit model is predicated on the lumped capacitance method [6]. Under the assumption that conduction is the only mode of heat transfer, the satellite's temperature can be characterized using two exponential curves for the heating and cooling phases. The equation describing the heating phase is as follows:

$$T(\theta_T) = \Delta T_h \cdot \left[1 - \exp\left(-\frac{\theta_T}{\tau_h}\right) \right] + T_0 \quad (8.1)$$

where ΔT_h is the temperature difference between the temperature at the beginning of the heating phase and the temperature the satellite would reach if illuminated constantly, τ_h is the time constant of the heating phase and T_0 is the temperature at the beginning of the heating phase.

The cooling phase is described by the following equation:

$$T(\theta_T) = T_{sat,c} + (T_e - T_{sat,c}) \cdot \exp\left(-\frac{\theta_T - 2\pi(1 - f_e)}{\tau_c}\right) \quad (8.2)$$

where $T_{sat,c}$ is the temperature the satellite would reach if it was in eclipse indefinitely, T_e is the temperature of the satellite at the end of the heating phase, f_e is the fraction of eclipse for the orbit and τ_c is the time constant of the cooling phase.

The following boundary conditions are imposed:

- The temperature at the end of the heating phase is equal to the temperature at the beginning of the cooling phase, i.e. $T(\theta_T = 2\pi(1 - f_e)) = T_e$.
- The temperature at the end of the cooling phase is equal to the temperature at the beginning of the heating phase. To ensure this, the value of $T_{sat,c}$ is assumed to be lower than T_0 by ΔT_0 . Using this condition, the value of τ_c can be calculated using [Equation 8.3](#).

$$\tau_c = \frac{-\ln\left(\frac{\Delta T_0}{T_e - T_{sat,c}}\right)}{2\pi f_e} \quad (8.3)$$

These conditions can be used to represent both the heating and cooling cycle of the satellite using the piecewise function shown in [Equation 8.4](#). The temperature data is fitted to this function by varying the parameters ΔT_h , τ_h , T_0 and ΔT_0 .

$$T(\theta_T) = \begin{cases} \Delta T_h \cdot \left[1 - \exp\left(-\frac{\theta_T}{\tau_h}\right)\right] + T_0 & \text{if } 0 \leq \theta_T \leq 2\pi(1 - f_e) \\ T_{sat,c} + (T_e - T_{sat,c}) \cdot \exp\left(-\frac{\theta_T - 2\pi(1 - f_e)}{\tau_c}\right) & \text{if } 2\pi(1 - f_e) \leq \theta_T \leq 2\pi \end{cases} \quad (8.4)$$

8.1.1. Validation of curve-fit model using FUNcube-1 data

The feasibility of the curve-fit model based on the lumped capacitance method was evaluated by fitting the model on the training dataset for FUNcube-1 and comparing its performance against the illumination-based model described in [Section 7.1.2](#). The results of the curve-fit model are shown against the test dataset for FUNcube-1 in [Figure 8.1](#) and the comparison of the KDEs of the base illumination model and the curve-fit model are shown in [Figure 8.2](#). The prediction error of the curve-fit model ($\mu = 0.00$ °C; $\sigma = 2.06$ °C) has a similar distribution to that of the illumination-based model ($\mu = -0.02$ °C; $\sigma = 1.91$ °C), being only slightly higher. This can be attributed to the limitation of the lumped capacitance model, which assumes only conductive heat transfer.

8.1.2. Hybrid networks for Delfi-PQ using curve-fit model

The results of the curve-fit for Delfi-PQ are shown in [Figure 8.3](#) and the fitting parameters are summarized in [Table 8.1](#). [Figure 8.4](#) depicts the KDE of the prediction error of the illumination-based model described in [Section 7.2](#) and the curve-fit model on the test dataset. Similar to the validation case for FUNcube-1, the prediction error of the curve-fit model ($\mu = 0.87$ °C; $\sigma = 5.96$ °C) is slightly higher than that of the illumination-based model ($\mu = 0.61$ °C; $\sigma = 5.72$ °C), but has the advantage of having better predictions for sections of the orbit where there is no telemetry data.

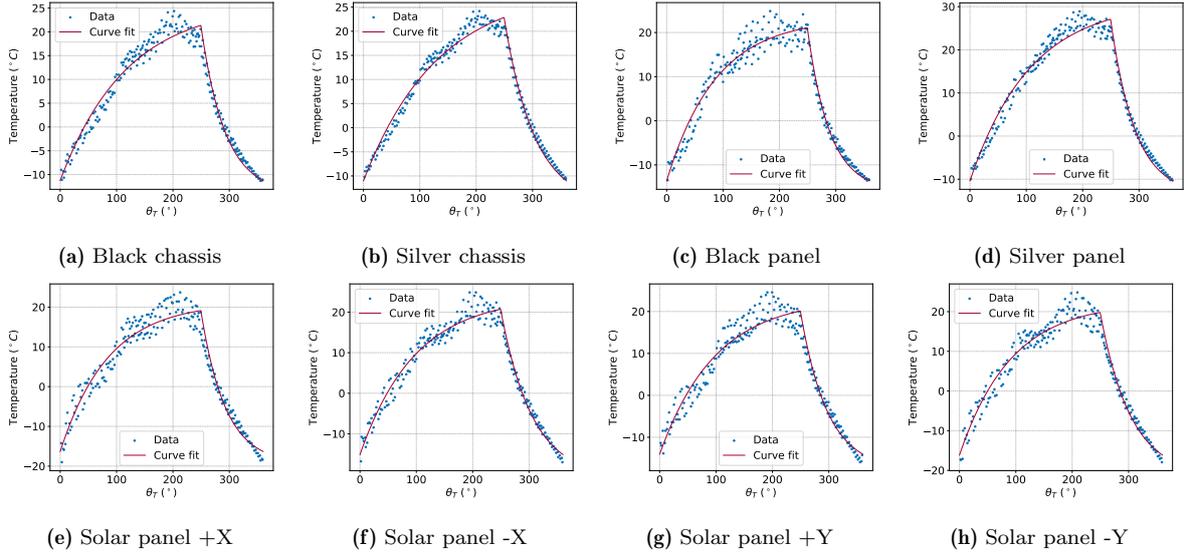


Figure 8.1: Curve-fit model prediction on FUNcube-1 test dataset

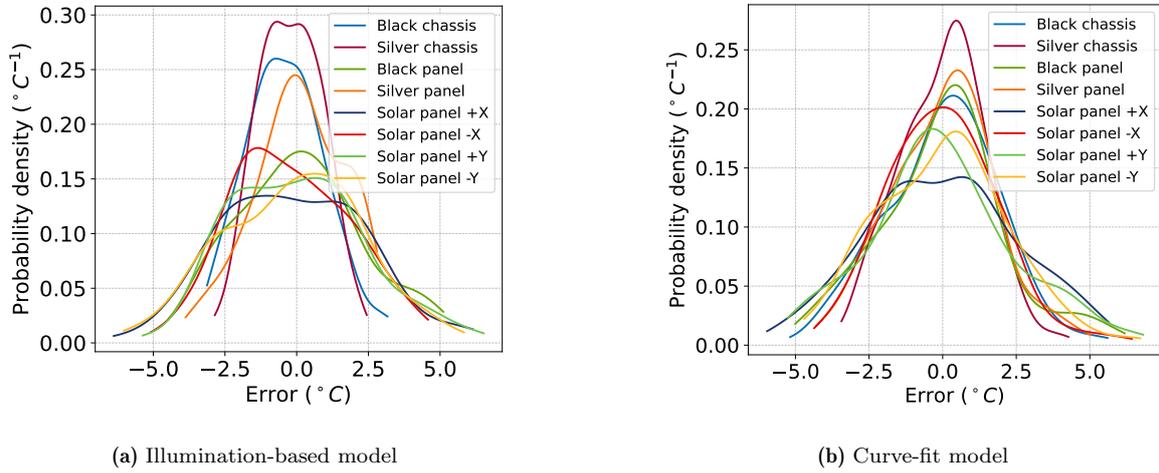


Figure 8.2: KDE of the prediction error of the illumination-based model and the curve-fit model on the FUNcube-1 test dataset.

The battery temperature has different heating and cooling trends than the other temperature channels. This is possibly due to the battery being an internal node, with the heat transfer to the battery being through conduction from the outer panels, along with the high heat capacity (C_p) of the battery, causing the temperature rise and fall to be almost linear over the timescale of one orbit. This can also be seen in the quality of the curve-fit for the battery temperature, where the fitted value for ΔT_h and ΔT_0 hitting the upper bound and lower bound of 300°C and 75°C respectively, and the heating rate τ_h being at least an order of magnitude lower than the other channels.

To improve the curve-fit model, a hybrid dataset was created by combining the existing training dataset for Delfi-PQ and simulated temperatures obtained using the curve-fit model. The number of simulated orbits ranged from one to ten orbits. Illumination-based models were trained using the hybrid datasets, and their prediction error was compared

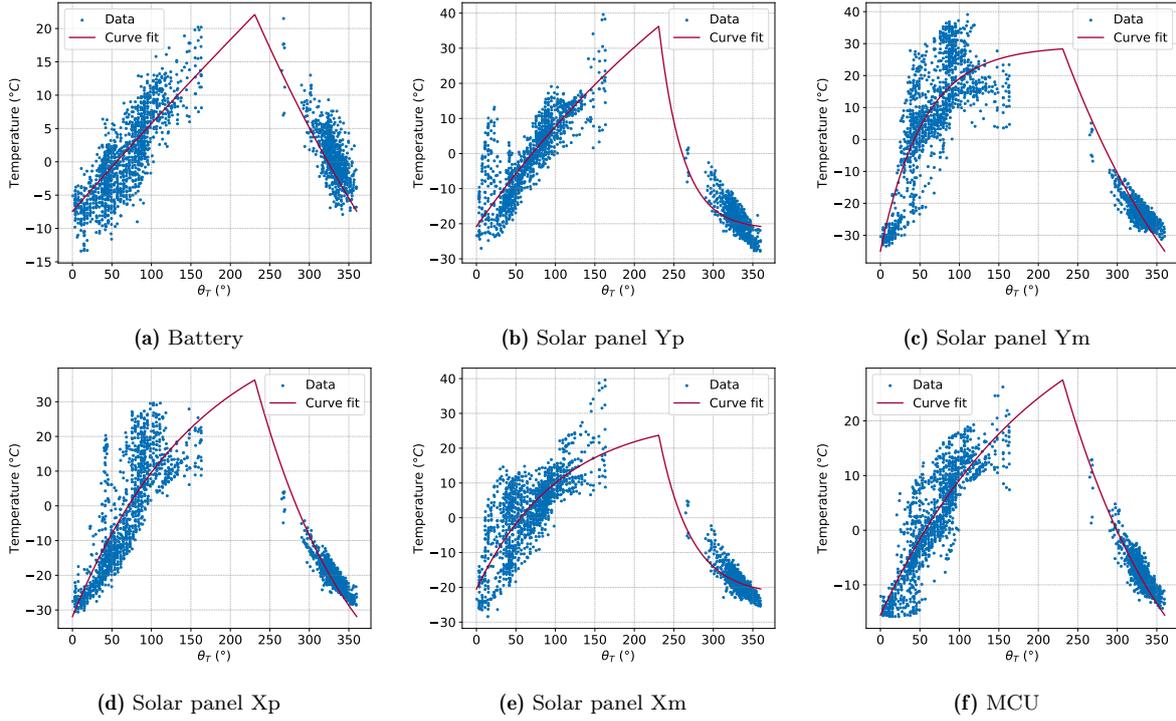


Figure 8.3: Exponential curve-fit on Delfi-PQ training dataset

Table 8.1: Curve-fit model parameters for Delfi-PQ.

Channel	ΔT_h (°C)	τ_h (°)	T_0 (°C)	ΔT_0 (°C)
Battery	300	4.41×10^{-4}	-6.93	75
Solar panel Yp	192.73	1.61×10^{-3}	-21.35	2.25
Solar panel Ym	62.20	1.94×10^{-2}	-34.09	44.97
Solar panel Xp	89.94	6.04×10^{-3}	-31.41	37.94
Solar panel Xm	48.77	1.02×10^{-2}	-21.22	4.61
MCU	58.72	5.52×10^{-3}	-15.64	41.52

to the prediction error of the model created using the original training dataset. [Figure 8.5](#) shows the hybrid model predictions for Delfi-PQ with different number of simulated orbits used for training the hybrid model.

The prediction error was calculated for both the test dataset and a simulated test orbit. The model prediction error against the test dataset can be used to evaluate the performance of the hybrid model for regions of the orbit with available data, while the prediction error with the simulated orbit can be used to gain insights on the theoretical performance of the model in regions with no available data. The results of this analysis are shown in [Figure 8.6](#). It was observed that the prediction error on the test dataset remained relatively constant, $\approx 5.8^\circ\text{C}$ with the addition of simulated orbits, while the prediction error on the simulated test orbit decreased with increase in number of simulated orbit, reducing from 5.5°C for no simulated orbit to 2.4°C for nine simulated orbits in

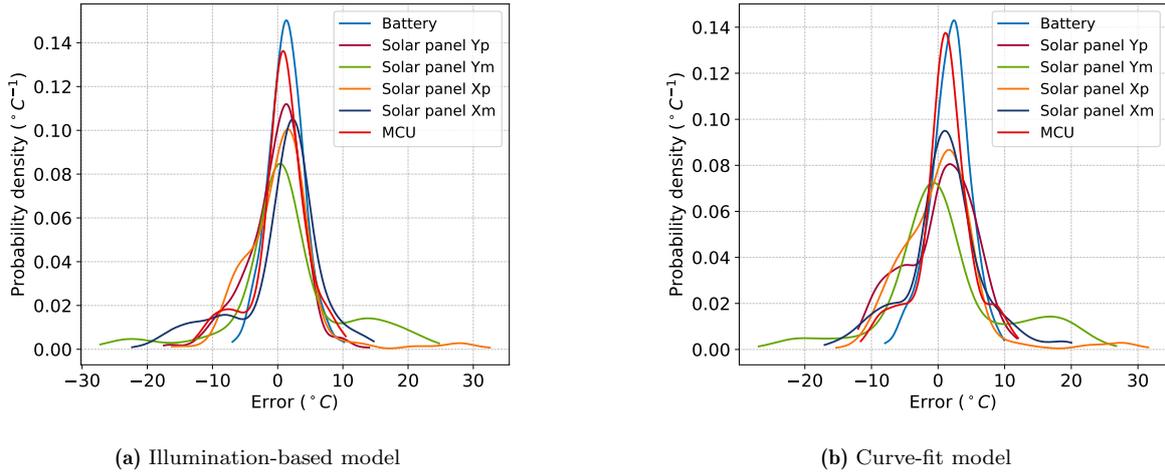


Figure 8.4: KDE of the prediction error of the illumination-based model and the curve-fit model on the Delfi-PQ test dataset

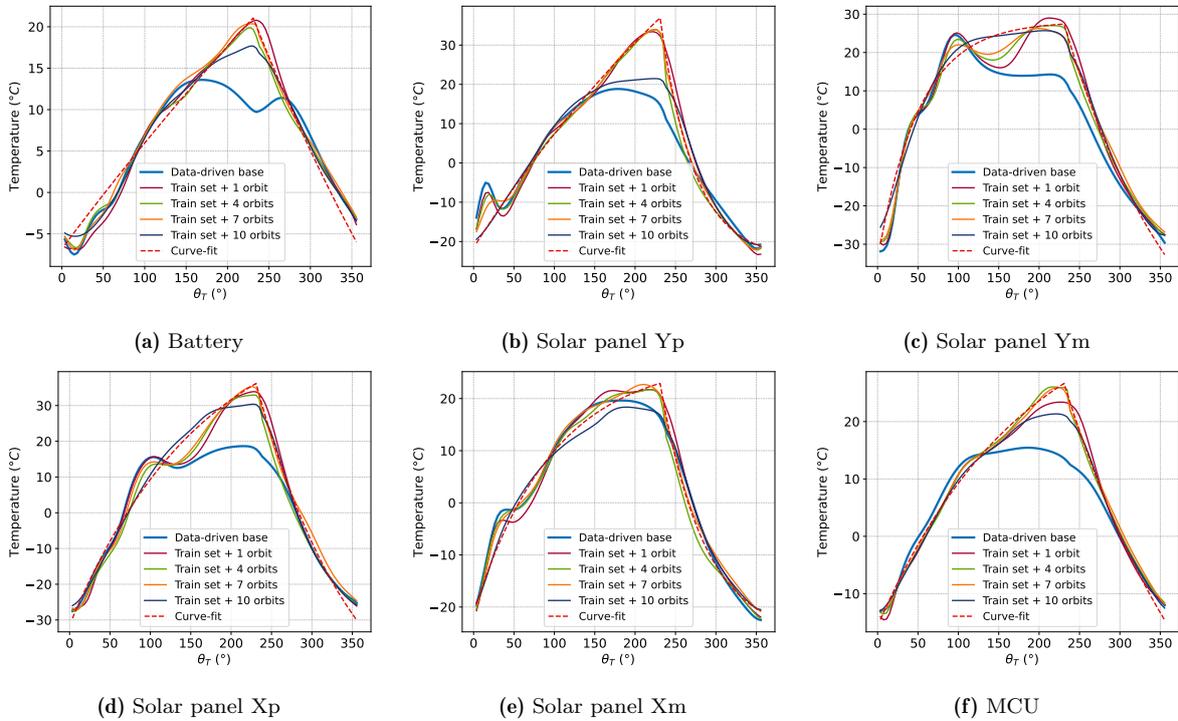
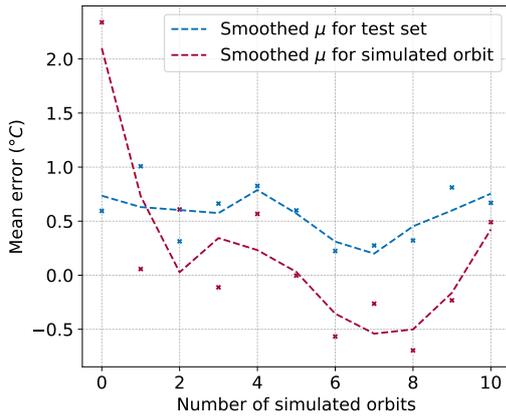


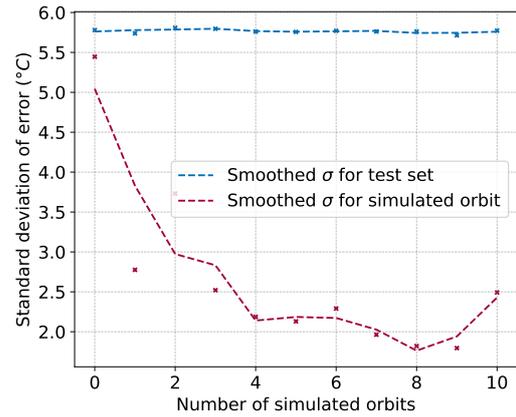
Figure 8.5: Hybrid model predictions for Delfi-PQ for different number of simulated orbits used for training.

the training data. From the validation test results described in [Section 8.1.1](#), it can be concluded that the hybrid model has a similar performance to the illumination-based model for regions of the orbit with available data, while having predictions closer to the probable true temperatures for the regions of the orbit with no available data.

Since the hybrid model predictions are similar to the illumination-based model, the possibility to reuse the temperature based residual models described in [Section 7.2](#) can be used without modification was tested, and the error statistics of the combined hybrid



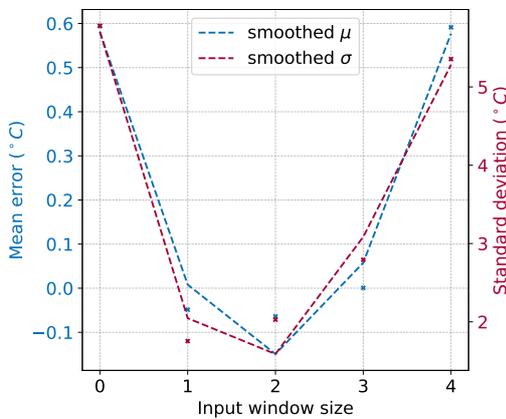
(a) Prediction error μ for different number of simulated orbits used for training



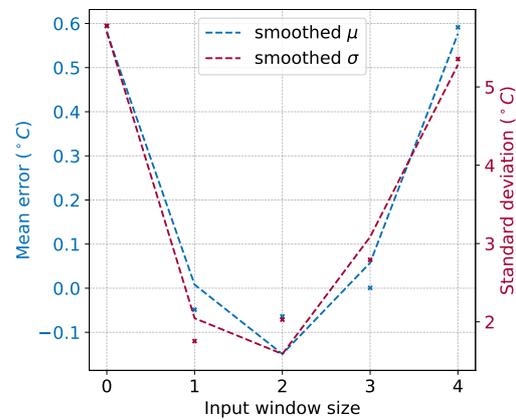
(b) Prediction error σ for different number of simulated orbits used for training

Figure 8.6: Prediction error of the hybrid model for Delfi-PQ for different number of simulated orbits used for training.

model and the temperature based residual models are shown in Figure 8.7. Comparing the error statistics to the ones depicted in Figure 7.16, which have very similar error distribution and variation with input window sizes, with the standard deviation difference for all the different input window sizes being less than $0.001\text{ }^{\circ}\text{C}$ between the illumination-based model and the two hybrid models. This implies that the temperature based residual models can be reused without modification for the hybrid model.



(a) Hybrid model with 2 simulated orbits



(b) Hybrid model with 8 simulated orbits

Figure 8.7: Error statistics of the combined hybrid base and residual models of various training simulated orbits and input temperature input window sizes for Delfi-PQ

Using a curve-fit model as the first-principles model in the hybrid model has some major drawbacks. The curve model investigated above considers each node separately and does not model the heat transfer between nodes. This is a limitation of the approach in this study and not in the lumped capacitance model. The heat loss from the satellite to space is in the form of infrared radiation, which is varies with T^4 instead of being linear in temperature as considered by the lumped capacitance model. The model used here is also a quasi-steady state model, thus the transient effects, such as the fluctuations caused

by the tumbling of the satellite, are difficult to model.

8.2. Thermal network model

The conductive heat flow for a node is given by the following equation:

$$\dot{Q} = \sum_{i=1}^n \frac{T_i - T}{R_i} + Q_{in} \quad (8.5)$$

where \dot{Q} is the thermal power flowing into the node, T_i is the temperature of the i th node connected to the node, T is the temperature of the node, R_i is the thermal resistance between the node and the i th node, and Q_{in} is the internal heat generated by the node.

The above equation closely resembles an electrical circuit, where the temperature is analogous to voltage, the heat flow is analogous to current, and the thermal resistance is analogous to electrical resistance. The inverse of the thermal resistance, named thermal conductance, is from here on referred to as the thermal coupling.

If the mass and the heat capacitance of the node is known, the temperature of the node can be described by the following differential equation:

$$\frac{dT}{dt} = \frac{\dot{Q}}{mC_p} = \frac{1}{mc_p} \left(\sum_{i=1}^n \frac{T_i - T}{R_i} + Q_{in} \right) \quad (8.6)$$

8.2.1. Thermal network model for Delfi-PQ

An 11 node thermal network model was developed for Delfi-PQ by Ruiz [44] using an inverse modeling approach, by fitting the thermal couplings to match simulated temperatures to the experimentally measured temperatures with a known heat input. The 11 nodes are: one node for each face of the satellite (six in total), one for each metal ring (three in total) and one each for the payload and battery. This model was expanded by Cinotti, Lusvarghi, Marchese, *et al.* [13] by adding more nodes to the payload and battery, creating a 13 node model. The thermal couplings between the nodes are described in Table 8.2. The physical properties of the nodes, consisting absorptivity (α), emissivity (ϵ), C_p and panel areas, are described in Table 8.3.

Since temperature measurements are available only for six nodes, comprising the four faces with the solar panels, the battery and the MCU, the 13 node network was reduced to an eight node network by removing nodes without temperature measurements from the model. The nodes for the Zp and Zm panels were retained, even though they do not have any temperature sensors, to properly calculate the input solar power to the satellite. The reduction in number of nodes was carried out using the star-point reduction technique [37], depicted in Figure 8.8. The new additional couplings between the nodes can be calculated using Equation 8.7

Table 8.2: Thermal couplings (W/K) between nodes in the 13 node thermal network model for Delfi-PQ

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	0	0	0	0	0	0	0	0	0	0	2.8×10^{-4}	1.2×10^{-1}	2.9×10^{-4}
2	0	0	0	0	0	0	0	0	0	0	2.8×10^{-4}	1.2×10^{-1}	2.9×10^{-4}
3	0	0	0	0	0	0	0	0	0	0	2.8×10^{-4}	1.2×10^{-1}	2.9×10^{-4}
4	0	0	0	0	0	0	0	0	0	0	2.8×10^{-4}	1.2×10^{-1}	2.9×10^{-4}
5	0	0	0	0	0	0	0	0	0	0	2.1×10^{-1}	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	7.9×10^{-2}
7	0	0	0	0	0	0	0	0	0	0	3.2×10^{-1}	3.2×10^{-1}	0
8	0	0	0	0	0	0	0	0	4.3×10^{-1}	0	0	3.2×10^{-1}	0
9	0	0	0	0	0	0	0	4.3×10^{-1}	0	9.7×10^{-1}	0	0	0
10	0	0	0	0	0	0	0	0	9.7×10^{-1}	0	0	0	9.7×10^{-1}
11	2.8×10^{-4}	2.8×10^{-4}	2.8×10^{-4}	2.8×10^{-4}	2.1×10^{-1}	0	3.2×10^{-1}	0	0	0	0	0	0
12	1.2×10^{-1}	1.2×10^{-1}	1.2×10^{-1}	1.2×10^{-1}	0	0	3.2×10^{-1}	3.2×10^{-1}	0	0	0	0	0
13	2.9×10^{-4}	2.9×10^{-4}	2.9×10^{-4}	2.9×10^{-4}	0	7.9×10^{-2}	0	0	0	9.7×10^{-1}	0	0	0

Table 8.3: Physical properties of nodes

Property	Node	Value
α	Panels Xp/Xm/Yp/Ym	0.91
	Panels Zp/Zm	0.1517
ϵ	Panels Xp/Xm/Yp/Ym	0.85
	Panels Zp/Zm	0.1355
Area (m^2)	Panels Xp/Xm/Yp/Ym	0.0089
	Panels Zp/Zm	0.0025
C_p (J/K)	Panels Xp/Xm/Yp/Ym	28.9784
	Panels Zp/Zm	8.14
	MCU	36.3
	Battery	66.65

$$R_{new} = \frac{R_i R_j}{\sum_{k=1}^n R_k} \quad (8.7)$$

Since the internal heat dissipations of the nodes being removed are small compared to the incident solar power and the heat flow from the outer panels to the internal nodes, the

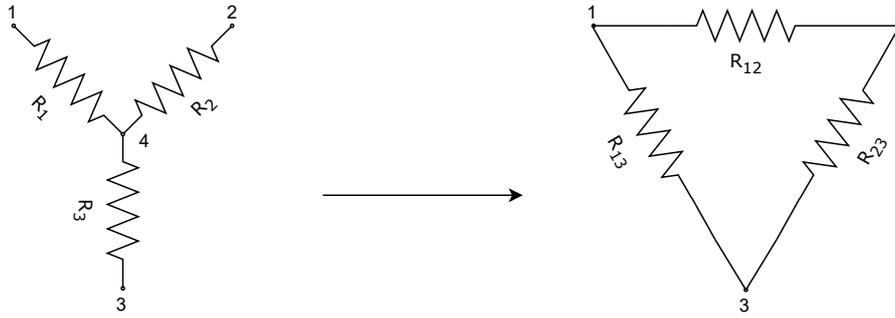


Figure 8.8: Star-point reduction

effect of removing these nodes on the temperature of the remaining nodes is negligible. The new thermal couplings are described in [Table 8.4](#).

Table 8.4: Thermal couplings (W/K) between nodes in the reduced 8 node thermal network model for Delfi-PQ

	1	2	3	4	5	6	7	8
1	0	1.7×10^{-2}	1.7×10^{-2}	1.7×10^{-2}	1.3×10^{-2}	4.1×10^{-5}	4.4×10^{-2}	2.5×10^{-4}
2	1.7×10^{-2}	0	1.7×10^{-2}	1.7×10^{-2}	1.3×10^{-2}	4.1×10^{-5}	4.4×10^{-2}	2.5×10^{-4}
3	1.7×10^{-2}	1.7×10^{-2}	0	1.7×10^{-2}	1.3×10^{-2}	4.1×10^{-5}	4.4×10^{-2}	2.5×10^{-4}
4	1.7×10^{-2}	1.7×10^{-2}	1.7×10^{-2}	0	1.3×10^{-2}	4.1×10^{-5}	4.4×10^{-2}	2.5×10^{-4}
5	1.3×10^{-2}	1.3×10^{-2}	1.3×10^{-2}	1.3×10^{-2}	0	0	3.3×10^{-2}	0
6	4.1×10^{-5}	4.1×10^{-5}	4.1×10^{-5}	4.1×10^{-5}	0	0	0	6.8×10^{-2}
7	4.4×10^{-2}	4.4×10^{-2}	4.4×10^{-2}	4.4×10^{-2}	3.3×10^{-2}	0	0	4.3×10^{-1}
8	2.5×10^{-4}	2.5×10^{-4}	2.5×10^{-4}	2.5×10^{-4}	0	6.8×10^{-2}	4.3×10^{-1}	0

8.2.2. Assumptions for the thermal simulation

The thermal simulation was carried out using the thermal network model described in the previous section, with the following assumptions:

- The Sun is assumed to be at infinity. This simplifies the calculation of varying heat input due to the rotation of the satellite. A more complicated model, which incorporates the β angle of the Sun vector, can be used to properly model the heat input from the Sun. This, however, would be important only when the satellite thermal behavior over a short period of time is of interest. Since the mean thermal

behavior of the satellite over a period of one year is of interest, where the fluctuations due to tumbling are averaged out, the assumption of the Sun at infinity is valid.

- The solar heat flux is assumed to be constant, set at 1400 W m^{-2} . The variation of the solar flux due to the eccentricity of the orbit is not considered.
- The satellite is assumed to rotate at a constant rate, with respect to an Earth-fixed non-rotating frame. Since no attitude data is available for Delfi-PQ, the spin rates estimates from Cinotti, Lusvarghi, Marchese, *et al.* [13] are used. The sparse attitude data available for Delfi-PQ makes it difficult to properly resolve the fluctuations due to tumbling, using which an estimate of the spin rate can be obtained. The spin rate is assumed to be constant over the entire mission duration.
- Earth's albedo is assumed to be constant, set at 0.3. The variation of the Earth's albedo over the orbit is not considered. The reflected solar flux is also assumed to be constant over the illuminated section of the orbit, hence no view factor calculations are required.
- Earth is assumed to be a perfect black body with a temperature of 300 K.
- All heat transfers between the nodes of the satellite are assumed to be only conductive transfers. This assumption can be made due to the relatively small temperature differences between the nodes of the satellite. The surface properties of the internal components of the satellite are also not characterized, making it difficult to model the radiative heat transfer between the nodes.
- All heat transfers from the satellite to space assumed to be radiative and only happening from the surfaces of the satellite facing space.
- The outer panels Xp, Xm, Yp and Ym are assumed to be fully covered in solar cells. The surface properties of these panels are assumed to be the same as the solar cells.

8.2.3. Physical property estimation

Using the thermal couplings and physical properties described in [Table 8.4](#) and [Table 8.3](#) respectively, the temperatures of the nodes were simulated by numerically integrating [Equation 8.6](#) using the fourth order Runge-Kutta method to obtain the temperatures for three orbits. The results of the simulation, depicted in [Figure 8.9](#), show that the satellite attains equilibrium temperatures after 3-4 orbits, with the temperatures at the same θ_T for different orbits being within 1°C .

Since there is no attitude data available for Delfi-PQ, the initial Euler angles were assumed to be 0 rad for all the thermal simulations. Though the choice of initial angles was found to have no effect on the general temperature trends and the amplitude of fluctuations in the temperatures, the choice of initial angles was found to have an effect on the phase of the fluctuations. Even with a perfect thermal model, the prediction errors can be large if the phases of the simulated temperature fluctuations and the actual satellite temperatures are not the same. To avoid this, the temperatures were smoothed using a third order Savitzky-Golay filter [45] with a window size of 100 s. The smoothed temperatures show a delay in the heating cycle for the internal nodes of the battery

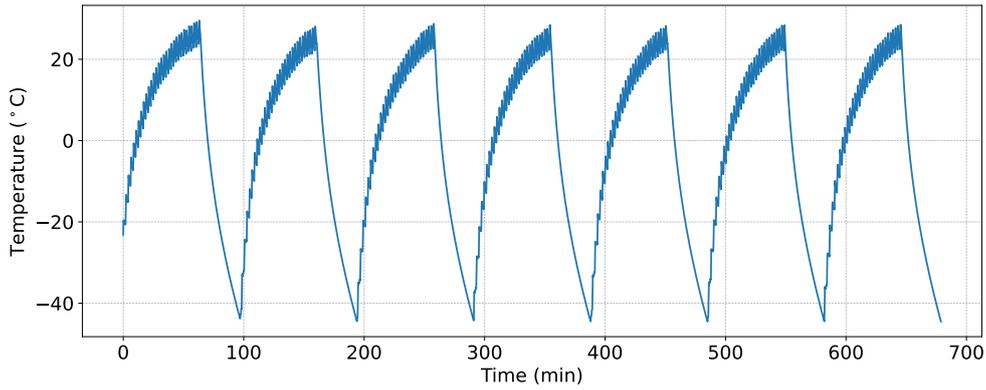


Figure 8.9: Simulated temperature of Xp solar panel for 680 minutes

and payload, compared to outer panels, which can be expected due to the delayed heat transfer from the outer panels to the internal nodes. This delay is more pronounced for the battery, which has a larger C_p compared to the payload.

Overlaying the simulated temperatures against the available temperature data, depicted in Figure 8.10, shows that the simulated temperatures are not a perfect fit to the measured temperatures. The KDE of the errors between the simulated and measured temperatures, containing not only the information about the mean and, depicted in Figure 8.11, shows that the errors are not centered around 0, indicating a bias in the model. The standard deviations of the errors are also large, depicted in the large width of the KDE, show that the model is not very accurate.

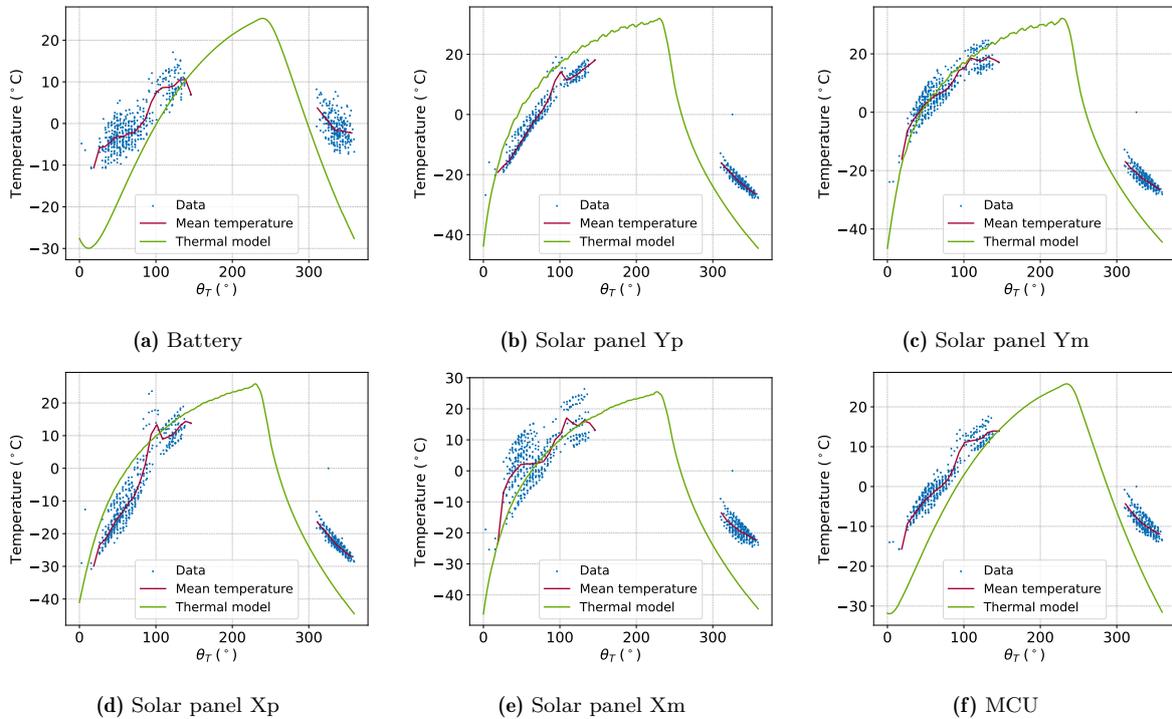


Figure 8.10: Base thermal network simulation

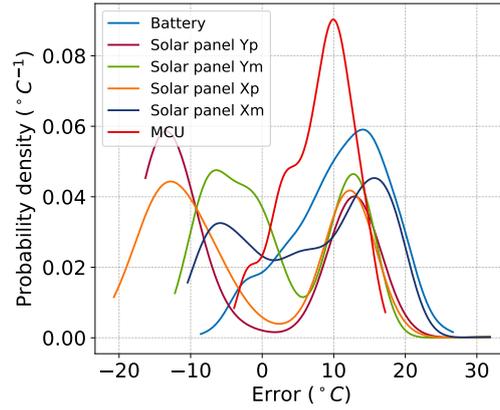


Figure 8.11: KDE of errors between simulated temperatures and telemetry data

The large errors might be due to the inaccuracies introduced by the assumptions made in the model, especially the assumptions about the heat input to the satellite from the Sun and Earth. As a possible solution, the physical properties of the satellite, such as the surface α and ϵ , and the C_p of the battery were varied to obtain a better fit to the measured temperatures.

The effect of the different parameters on the simulated temperatures was studied to identify suitable parameters to use as design variables for the optimization process. A selection of parameters, namely the coupling matrix, C_p , surface α and ϵ , initial Euler angles and the spin rate of the satellite were scaled by $\pm 20\%$ and the effect of the perturbations on the maximum and minimum temperatures of the simulated temperatures were analyzed.

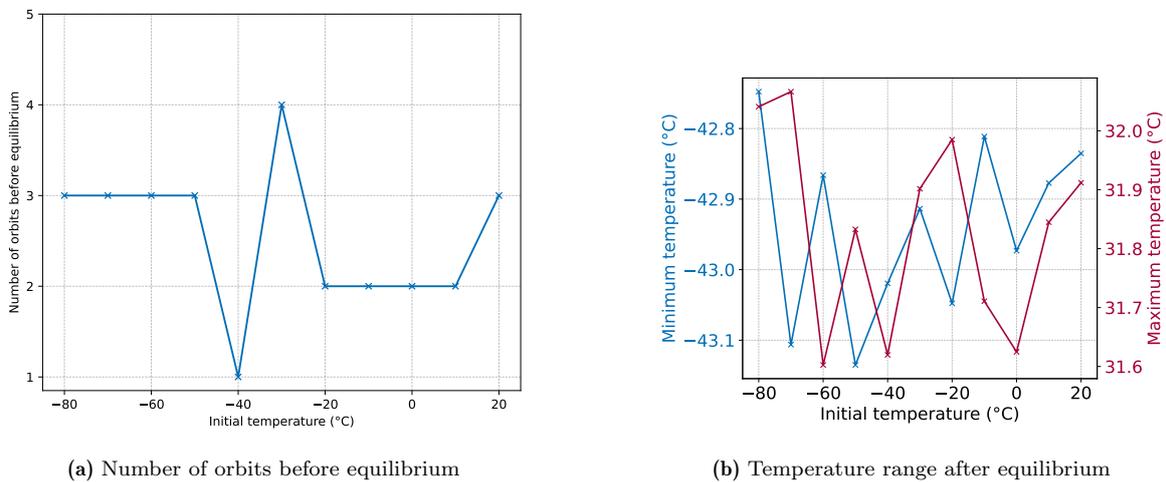


Figure 8.12: Effect of initial temperature on the thermal network simulation for solar panel Xp

The effect of the initial temperature of the nodes on the convergence of the temperatures over the orbits is depicted in Figure 8.12a. The temperatures over an orbit were considered to have converged when the minimum and maximum temperature, occurring at the end of the eclipse and illuminated sections of the orbit respectively, was within 1°C of the previous orbit. It was observed that for all initial temperatures studied, the number

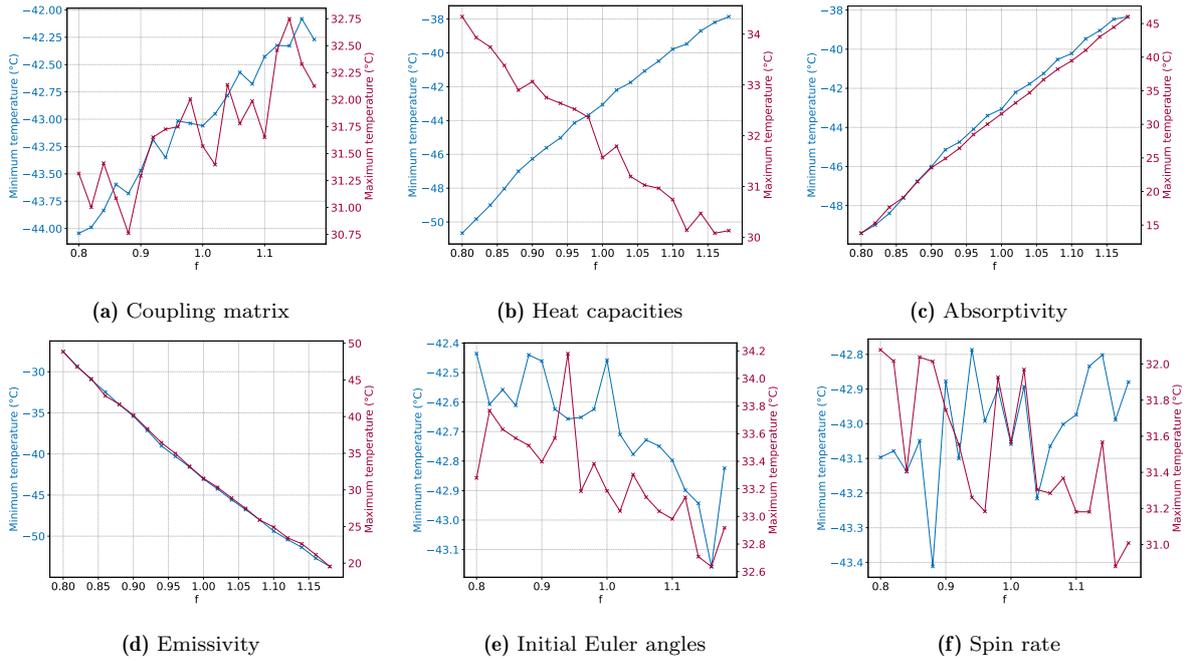


Figure 8.13: Effect of perturbations on the thermal network simulation for solar panel Xp

of orbits before equilibrium was less than five. Thus, for all further simulations, the temperatures of the fifth simulated orbit were selected as the representative temperatures for the satellite. The effect of the initial temperatures on the minimum and maximum temperatures of the satellite, depicted in Figure 8.12b, was found to be within 1°C for all initial temperatures studied. Thus, the initial temperatures were not considered as design variables for the optimization process.

The effect of the coupling matrix on the minimum and maximum temperatures (Figure 8.13a) was also found to be small for all perturbations studied, being less than 2.5°C . From Equation 8.6, it can be observed that increased C_p increases the energy required to change the temperature of the node. For a constant heat input, the increase in C_p would be expected to reduce the range of temperatures of the node over the orbit. This expected behavior was observed in the simulation results, as depicted in Figure 8.13b. The surface α and ϵ of the outer panels determines the amount of heat absorbed and emitted by the satellite. Increasing α and reducing ϵ both have the same effect of increasing the heat available to the satellite for increasing the temperatures. This effect is depicted in Figure 8.13c and Figure 8.13d. The effect of the initial Euler angles and the Euler spin rate of the satellite used for the simulation was found to have an effect of similar magnitude as the effect of the coupling matrix on the simulated temperatures. This is depicted in Figure 8.13e and Figure 8.13f respectively. The effect of changing initial Euler angles cannot be integrated into the thermal network simulations used in this study, and thus was set to zero for all simulations.

The effect of the various parameters on the simulated temperatures, having the same trend as expected from the governing equations, increases the confidence in the thermal network simulation despite the simplifications made.

Several optimization algorithms were explored to minimize the RMSEs between the

simulated temperatures and the telemetry data, namely Nelder-Mead simplex algorithm [18], a zeroth-order method, trust region [15] and Sequential Quadratic Programming (SQP) [7], two first-order methods. The order of the method refers to the order of the derivative information required for the optimization algorithm. The first-order optimization algorithms require gradient information, analytical calculations of which were not explored in this study, and only finite difference methods were investigated. The time taken for simulating temperatures for 5 orbits was about 30 s, making it computationally expensive to run first-order optimization algorithms using finite-difference methods for a large number of design variables. Thus, the Nelder-Mead simplex optimization algorithm was chosen for this study.

The optical properties of the outer panels, the C_p of the nodes and the thermal resistances between the nodes were chosen as possible design variables for the optimization algorithm. As depicted in Figure 8.13, several of the design variables have similar effects on the simulated temperatures. Thus, the inverse modeling problem of fitting the design variables to the telemetry data was found to be under-determined. No convergence of the optimization algorithm was observed when all the design variables were fitted simultaneously. An alternative approach of a step-wise optimization process is thus proposed.

First, the α and ϵ of the outer panels were fitted by formulating a 12 design variable minimization problem. The large errors in the predicted temperatures were found to cause bad convergence in the optimization algorithm. Thus, a simple brute-force search was used to minimize the RMSE between the simulated and measured temperatures using a scaling factor f for the panel α and ϵ as the design variable. The design space was constrained to $0 < f < 1.5$ and a search was conducted for all values of f with a input window size of 0.1. The best fit was found to be for $f = 0.7$. The initial point for the Nelder-Mead simplex algorithm was set as the values of the α and ϵ described in Table 8.3, scaled by $f = 0.7$. Figure 8.14 depicts the simulated temperatures for the scaled initial α and ϵ values. Comparing this with Figure 8.10, it can be seen that the simulated temperatures are a better fit to the measured temperatures.

The selection of the cost function to minimize was found to have a large impact on the optimization results. The smoothing of the simulated temperatures using the Savitzky-Golay filter removes the spread in temperatures due to tumbling, it was considered logical to try to remove the spread of the temperatures from the telemetry data as well. Thus, similar to the method described in Section 6.2.4, the telemetry data was binned into θ_T ranges of 5° , and the mean of the temperature in each bin was calculated, depicted in Figure 8.15. All prediction errors were calculated against the mean telemetry data, instead of the raw telemetry data. This was found to improve the convergence of the optimization algorithm.

Since the amount of data for the illuminated side is greater than the eclipse side, if the RMSE was calculated over the whole dataset, as described in Equation 8.8, it was found that the optimization algorithm converged to fit to the heating phase, while worsening the cooling phase predictions.

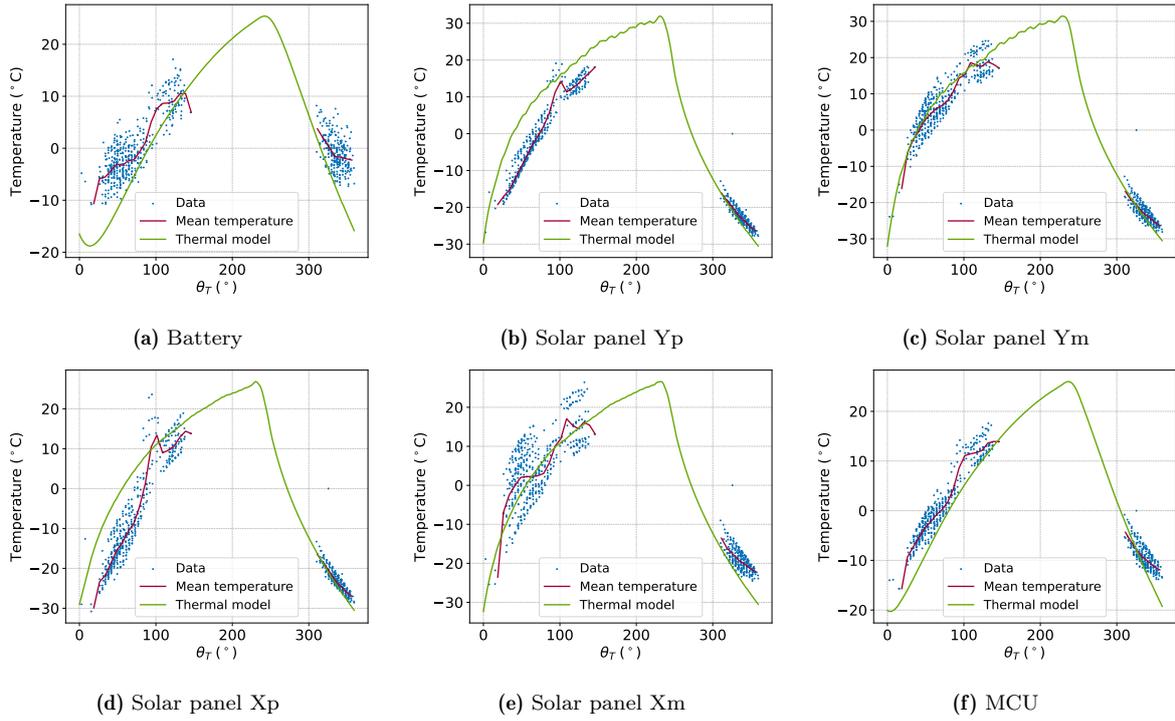


Figure 8.14: Thermal network simulation for $f = 0.7$

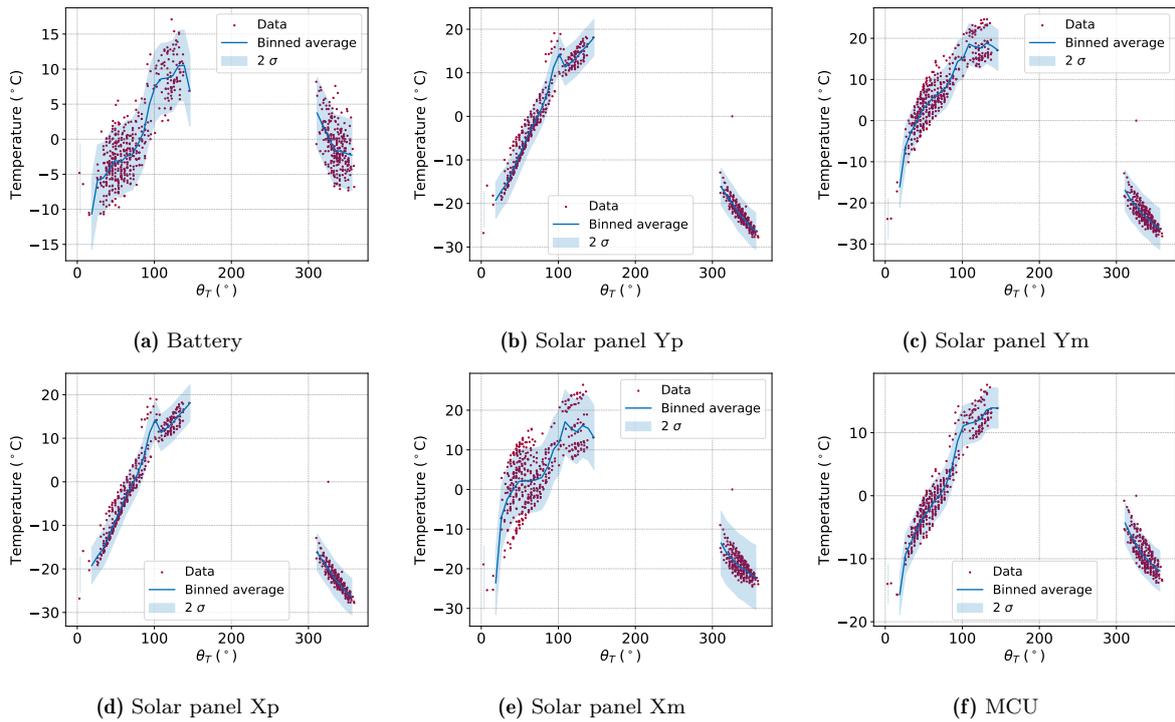


Figure 8.15: Mean temperatures of the Delfi-PQ telemetry between 2022-01 and 2022-04

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (T_{sim,i} - T_{mean\ data,i})^2} \quad (8.8)$$

As a possible solution, the errors were calculated separately for the heating and cooling phase of the temperatures, and the final cost function was defined as the mean of the two RMSEs, described in Equation 8.9. Even after the filtering steps described in Chapter 6, and the compensating for spread in telemetry data as described previously, a small region of noisy data was observed in the solar panel temperatures around $\theta_T \approx 100^\circ$. Thus, the RMSE of the heating phase was calculated for $\theta_T < 80^\circ$ to avoid the noisy data.

From here onwards in the report, the mean RMSE between the heating and cooling phases described in Equation 8.9 is referred to as the RMSE.

$$\overline{RMSE} = \frac{RMSE_{heating} + RMSE_{cooling}}{2} \quad (8.9)$$

The Nelder-Mead optimization algorithm was run for a maximum of 2500 iterations with a tolerance of $10^{-4} \text{ }^\circ\text{C}$ for the reduction in RMSE and 10^{-4} for the change in design variables. The design space was constrained to 0.1 to 1 for the values of α and ϵ .

The optimum values for the values of α and ϵ for the panels are summarized in Table 8.5.

Table 8.5: Optimized values of α and ϵ for the panels

Panel	α		ϵ	
	Original	Optimized	Original	Optimized
Xp	0.91	0.6215	0.85	0.5894
Xm	0.91	0.6698	0.85	0.5223
Yp	0.91	0.5935	0.85	0.5983
Ym	0.91	0.6504	0.85	0.5828
Zp	0.1517	0.1108	0.1355	0.0994
Zm	0.1517	0.1142	0.1355	0.0974

The simulated temperatures for the optimized values of α and ϵ are depicted in Figure 8.16. The RMSE between the simulated and measured temperatures was found to be $4.36 \text{ }^\circ\text{C}$, which is an improvement over the RMSE of $11.14 \text{ }^\circ\text{C}$ for the original values of α and ϵ .

Next, the C_p of the battery and MCU nodes were optimized, while allowing for small changes to the optimized values of α and ϵ . This was done to compensate for the possible reduction in panel temperatures due to increased C_p of the internal nodes. The optimization problem was formulated as a 14 design variable minimization problem, with the design variables being the C_p of the battery and MCU nodes, and the α and ϵ of the panels. The initial values for α and ϵ were set to the results of the previous optimization results. The design space was constrained to 0.1 to 5 times the original values of the C_p of the battery and MCU nodes, and 0.75 to 1.25 times the optimized values for α and ϵ described in Table 8.5. The results of the optimization are summarized in Table 8.6. The RMSE of the solution of this optimized run was $3.89 \text{ }^\circ\text{C}$, a small improvement from the previous optimization run.

From Figure 8.17, it can be seen that the thermal network simulation results in temperatures closer to the observed temperatures for Delfi-PQ; however, it is still not a perfect

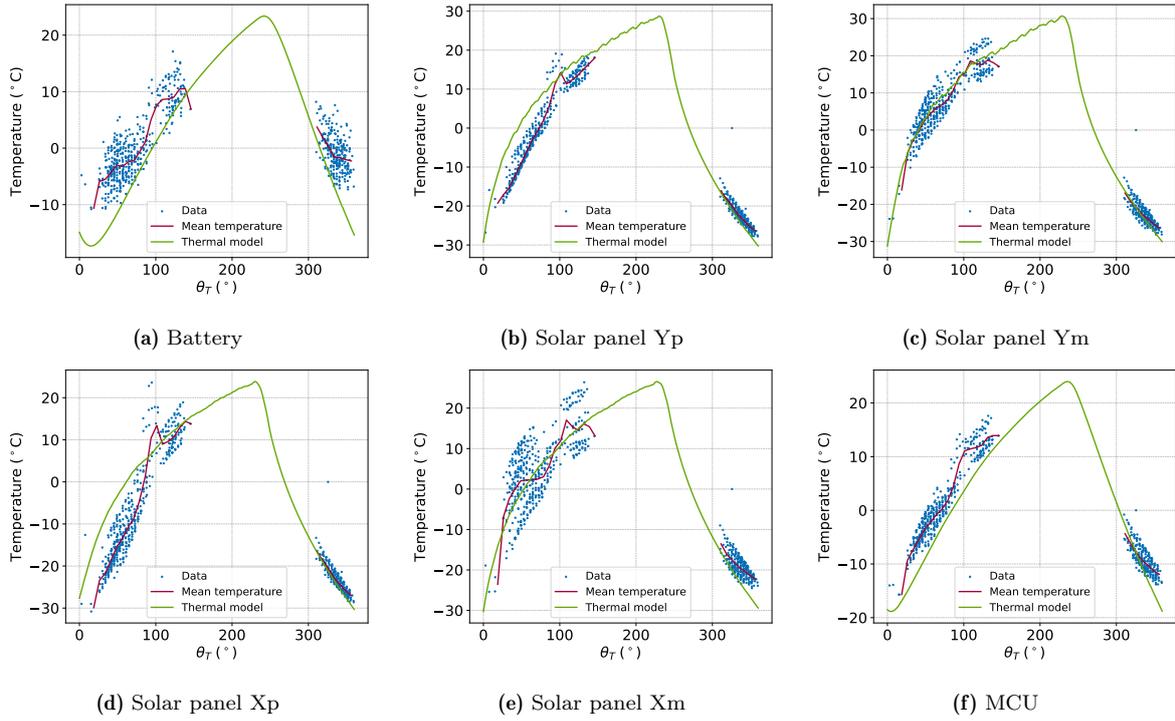


Figure 8.16: Simulated temperatures of the thermal network with optimized values of α and ϵ

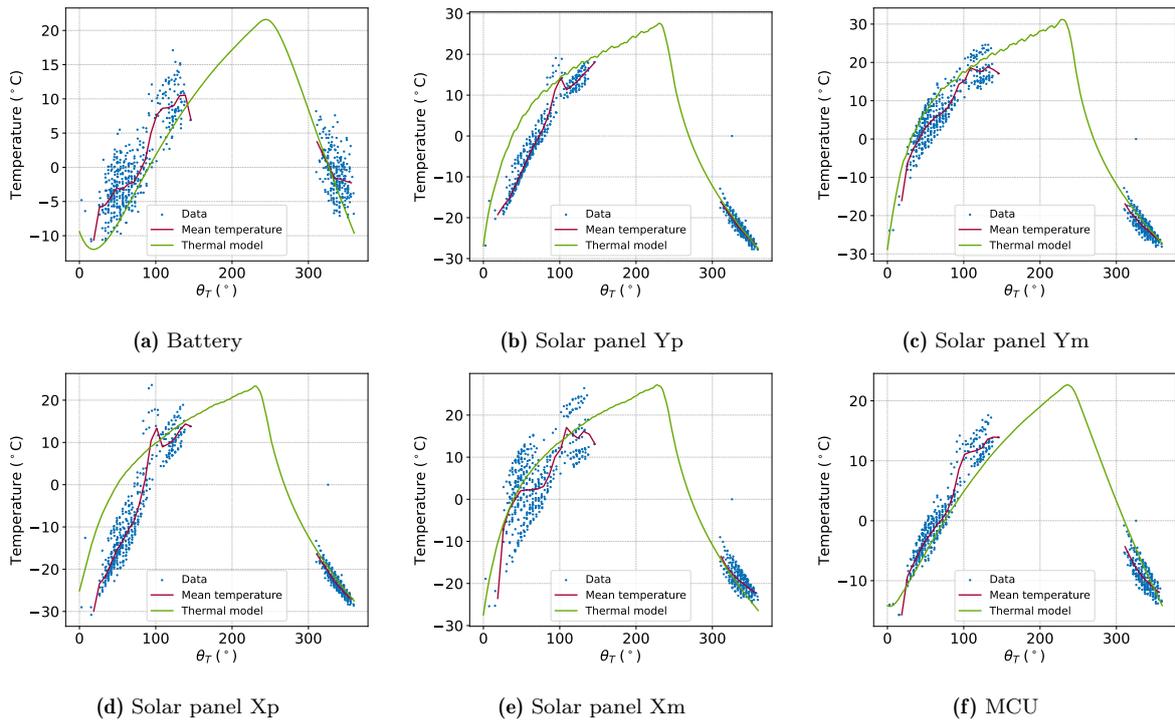


Figure 8.17: Simulated temperatures of the thermal network with optimized values of α , ϵ and C_p

match. Attempts to improve the match by further optimizing the thermal couplings between the nodes were attempted. If all thermal couplings described in Table 8.4 are used as the design variables for the optimization, it results in a 28 design variable

Table 8.6: Optimized values of α , ϵ for the panels, and C_p for battery, and MCU

Node	α		ϵ		C_p (J/K)	
	Initial	Optimized	Initial	Optimized	Original	Optimized
Xp	0.6215	0.6087	0.5894	0.5907	-	-
Xm	0.6698	0.6773	0.5223	0.5106	-	-
Yp	0.5935	0.5745	0.5983	0.5961	-	-
Ym	0.6504	0.6547	0.5828	0.5823	-	-
Zp	0.1108	0.1119	0.0994	0.0999	-	-
Zm	0.1142	0.1157	0.0974	0.0969	-	-
MCU	-	-	-	-	36.3	35.6975
Battery	-	-	-	-	66.65	109.799

minimization problem, not considering the inclusion of the panel surface properties and the C_p of the internal nodes, resulting in a 32 design variable minimization problem. Both variations of the optimization problem were attempted; however, the optimization runs did not converge to a usable solution. For the first formulation with 28 design variables, the optimization run did not converge to a solution within 2500 iterations, taking over 12 hours to finish. For the second formulation, the optimization run converged after ≈ 1700 iterations, but the optimized thermal couplings had changed by less than 5% of their initial values. The RMSE of the optimized solution was 3.82°C , which is not a large improvement over the previous optimization run for the computation time required. Thus, the optimization process was stopped at the previous optimization run.

An important caution to note is that the optimized parameters described in the results obtained above are only valid for the specific assumptions made in the thermal network model. The most important assumption is the heat input assumptions made for the outer panels. A constant albedo of 0.3 for the illuminated section of the orbit, which is a reasonable assumption only when $\theta = 0^\circ$ with it being lower for other parts of the illuminated section of the orbit. This can also be observed in the results of the optimization run, where the α of the panels are considerably lower than the initial values. This is possibly caused as a result of the overestimation of the heat input to the panels, which is compensated by reducing the α of the panels. The heat input to the panels can be more accurately modeled by using a time varying albedo factor to account for the viewing angle, which is a function of the angle between the Sun and the orbit plane. This would require the use of a more complex thermal model, which would be more computationally expensive to solve. Thus, the thermal network model is a good compromise between accuracy and computational cost.

The RMSEs of the thermal model at the different optimization stages are summarized in [Table 8.7](#).

Table 8.7: RMSE of the thermal network at different optimization stages

Thermal model	RMSE ($^{\circ}C$)
Base thermal network	11.1357
Thermal network with $f = 0.7$	5.5748
α, ϵ optimized thermal network	4.3567
α, ϵ, C_p optimized thermal network	3.8864

8.2.4. Residual networks

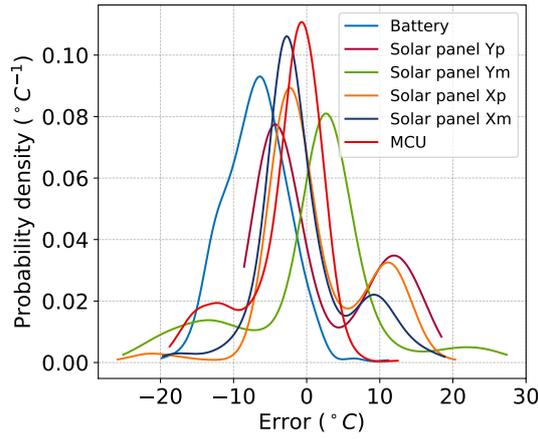
Using the approach described in [Section 7.2](#), residual networks were trained to correct for the errors in the thermal network model. Instead of using an illumination-based model (output of which is of very low confidence in the no-data regions) or a curve-fit based model (does not model any thermal interactions between the different sections of the satellite), the base model used here is the thermal network model with the values of α and ϵ for the panels and the C_p of the battery and MCU optimized to fit the telemetry data as described above. Residual networks of input window sizes ranging from 1 to 4 previous time steps of temperature data were trained. The KDE of the prediction error of the thermal network model and the combined models are shown in [Figure 8.18](#), and the prediction error statistics with varying input window sizes are depicted in [Figure 8.19](#). It can be observed that the standard deviation of the prediction error of the thermal network model drops by around $5^{\circ}C$ with the addition of just one time step of temperature data using the residual models. This is also evident in the reduction in spread of the KDE in [Figure 8.18](#). As seen for temperature based residual networks built for FUNcube-1 and other base models for Delfi-PQ, the standard deviation of the prediction errors reduce with an increase in input window size. Similar to the previous residual models built for Delfi-PQ described in [Section 7.2](#) and [Section 8.1.2](#), the small number of training and testing samples available for input window size = 4 results in an increase in standard deviation of the errors.

The prediction error metrics of the various data-driven and hybrid models for Delfi-PQ are summarized in [Table 8.8](#).

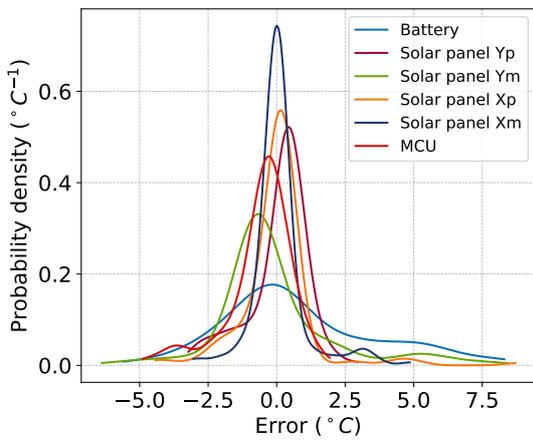
8.2.5. Batched optimization

Even though the optimized parameters do not have direct correlation to their physical counterparts, attempts to gain insights into the thermal evolution of the satellite with time were made. With the variation of the input solar radiation over the course of a year due to changing distance between the Sun and the Earth, the heat input for the satellite changes over the course of the year. Since the thermal network model assumes a constant solar input, we can expect the optimized surface parameters of the panels to mirror this trend in the heat input. We can also expect the satellite to run hotter with time due to degradation, so we should also be able to see a general increase in α and reduction in ϵ after the optimization process.

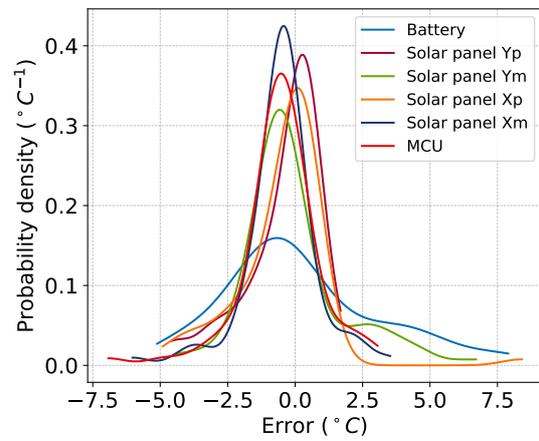
To try to extract these trends, the telemetry data was batched for durations of 30, 45,



(a) α, ϵ, C_p optimized thermal network



(b) α, ϵ, C_p optimized thermal network + temperature-based residual network with $w = 1$



(c) α, ϵ, C_p optimized thermal network + temperature-based residual network with $w = 2$

Figure 8.18: KDE of the prediction error of the thermal network model and the combined models

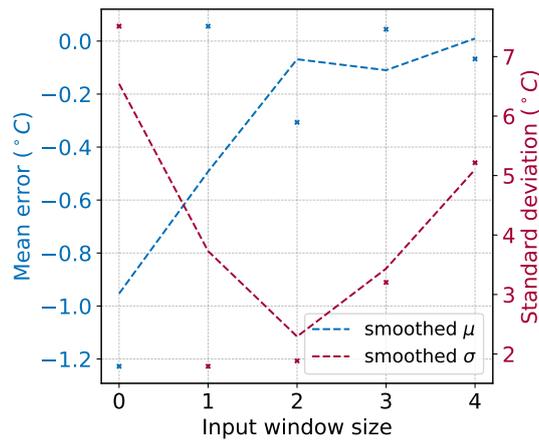
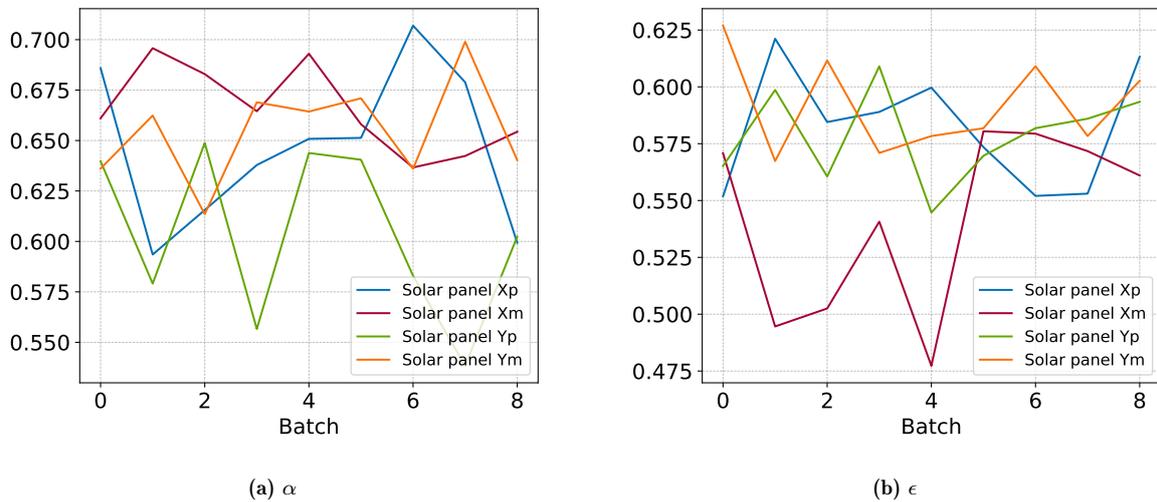


Figure 8.19: Prediction error statistics of the thermal network combined model for different input window sizes

Table 8.8: Summary of FUNcube-1 models

Model	Error μ ($^{\circ}C$)	Error σ ($^{\circ}C$)
Illumination-based model ($w = 94$)	0.61	5.72
Illumination-based model ($w = 94$) + temperature-based residual model ($w = 1$)	0.00	1.64
Illumination-based model ($w = 96$) + temperature-based residual model ($w = 2$)	-0.05	2.03
Curve-fit model	0.87	5.96
Base thermal network	10.24	10.07
Thermal network with $f = 0.7$	4.24	7.94
α, ϵ optimized thermal network	2.03	7.75
α, ϵ, C_p optimized thermal network	-1.27	7.59
α, ϵ, C_p optimized thermal network + temperature-based residual model ($w = 1$)	0.06	1.79
α, ϵ, C_p optimized thermal network + temperature-based residual model ($w = 2$)	-0.31	1.88

60, 75 and 90 days and the two-step optimization process described above was performed on each of the batches. For a batch size of 45 days, the variation of the optimum α and ϵ to minimize the RMSE is depicted in Figure 8.20. The expected seasonal pattern in the fitted values were not observed in the results of the optimization process, and appears to be mostly dominated by noise.

**Figure 8.20:** Variation of optimized α and ϵ for batches of 45 days

To understand the reason for this, the variation of the RMSE for different batches and the spread of the data (calculated as the mean of the standard deviations of the telemetry data in every 5° bin of θ_T) were analyzed, and the same is depicted in Figure 8.21. It was observed that there was a rough between the RMSE and the spread in the data. The method of compensating for the spread of the data by considering only the mean trend

of the temperatures does not appear to be completely effective, although it helped in the convergence of the optimization algorithm. The implications of these results is that the thermal network model is susceptible to the temperature variation due to tumbling. Even though a full rotation model is incorporated in the thermal network model, the rotation calculations are not constrained properly due to the absence of attitude data, and relying on the mean trend of the temperatures does not completely compensate for this.

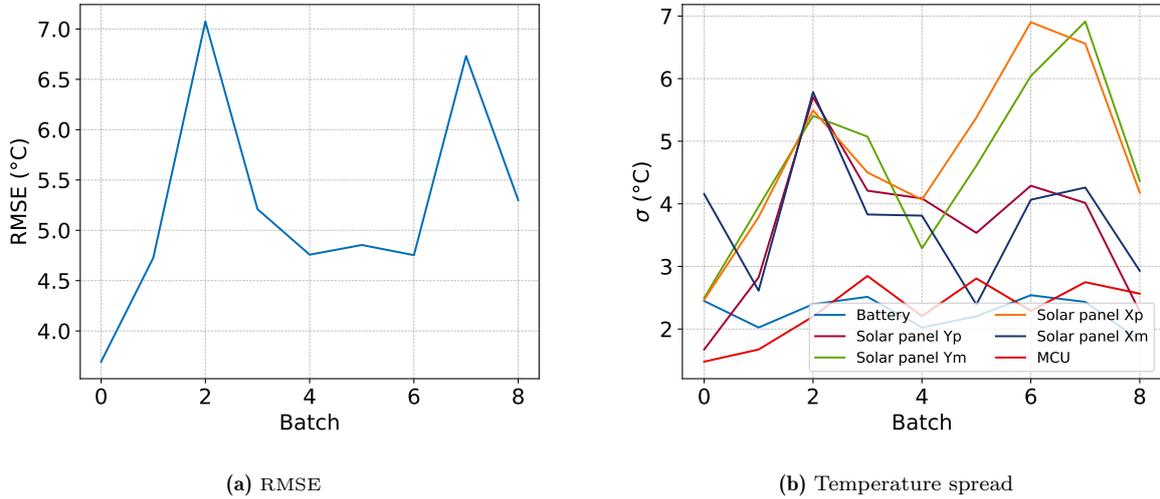


Figure 8.21: Variation of RMSE of the optimization results and the spread of the data for batches of 45 days

9

Conclusion

The research presented in this study sought to answer the three main research questions defined in [Chapter 3](#). The answers to these questions are summarized in this chapter, along with recommendations for future work.

9.1. Answers to research questions

How effective are the data-driven neural networks in predicting the temperatures of the satellites analyzed in this study.

The applicability of LSTM models for time series prediction tasks, including those relevant to space applications, has been demonstrated by various prior research (as per the references cited in the literature review section).

For the data-driven models developed in this study, the FUNcube-1 data set was selected as the base case. This was due to the availability of full orbit data, at a regular sampling rate of 1 min and the presence of distinct anomalies. In regions of nominal operation, the models that were solely reliant on preceding temperature data were able to effectively predict future temperatures. However, these models failed to detect and identify anomalies concurrently in regions characterized by anomalous conditions. For small input window sizes, the models followed the anomalous temperature trend, with the prediction error having a similar magnitude as the nominal regions. For larger input window sizes, it was possible to detect anomalies as regions with large prediction error, but these errors persisted for several time steps after the anomaly had ended.

As a possible solution, an approach was adopted that decoupled temperature prediction from previous temperature data using an illumination-based model. For sufficiently long input window size, long enough for each sample in the orbit to have a unique previous illumination sequence, the illumination regressed to the mean temperature trend of the satellite. This removed the fluctuations due to tumbling from the predictions.

Since the illumination model did not have temperature data as an input, the measured

temperatures did not affect the model predictions, enabling the model to simultaneously detect and isolate anomalies by monitoring the prediction error. A temperature-based secondary network was developed to correct for the residuals of the illumination-based model. When used with the illumination model, the combined model resolved fluctuations due to tumbling, reducing the prediction error to $\sigma < 1^\circ C$.

However, these purely data-driven models have limitations when applied to telemetry from Delfi-PQ, which has gaps in coverage over the orbit. The model predictions have high accuracy in regions with available training data, but cannot be trusted in regions without data.

For the residual networks developed for Delfi-PQ, the input window size was constrained by the availability of suitable contiguous training and testing data. From the prediction error variation for different step sizes, it is possible that the residual network performance is close to saturation at an input window size of 3 steps.

How does the performance of first-principles model compare to the data-driven models?

This study focused on two first-principles models for Delfi-PQ, namely the exponential curve fitting based on the lumped capacitance model, and a thermal network model. It was found that merely relying on the exponential curve-fit as the base model might result in a higher error in comparison to models that are purely data-driven. As discussed before, the base model error is important in identifying and isolating anomalies.

To enhance the performance of the base model, a synthetic training dataset was constructed by merging existing training data with simulated data originating from the curve-fit. This resulted in an improvement in the performance of the curve-fit model, with similar prediction error as compared to the illumination-based model.

The inclusion of residual networks into the curve-fit based model resulted in a decrease in prediction error. The residual network developed for the data-riven model could be reused with the curve-fit model due to similar predictions in data-available regions.

In the curve-fit model, thermal coupling between the nodes was not considered. Some coupling may have been inherently captured when the synthetic data was used to train the model, but this might not be accurate. This limitation can be overcome by employing a thermal network model as the base model.

The sparsity of the telemetry and no available attitude data for Delfi-PQ posed a considerable challenge in fully constraining the physical parameters of the satellite used for the thermal model. The heat input from the Sun and Earth, a key component of the thermal model, was highly simplified to improve computation time, resulting in a large error in the model predictions. An optimization process is proposed to estimate model parameters to better align with the assumptions used.

Optimization presented further complications due to the high correlation among various parameters, such as the surface optical properties, heat capacities, and thermal couplings. The close relationship between these parameters rendered simultaneous optimization a complex task, as changes in one variable would invariably affect the others. To work around this issue, a step-wise optimization process was implemented. However, this

approach brought about its difficulties, with the optimized parameters not being physically accurate.

Due to the large computation cost of the thermal network model, the optimization process was only performed on the surface optical properties of the external panels and the heat capacities of the internal nodes. The optimization of the thermal coupling, while yielding small improvements in the model predictions errors, had a disproportionately high computation cost, and was not pursued further.

As in previous models, residual networks could also be utilized to enhance the performance of the combined model. The combined first-principles model and residual network, when compared to the purely data-driven models, have similar prediction error in regions with available data. However, the first-principles model can provide predictions in regions without data with a higher degree of confidence than the purely data-driven models.

Can the evolution of the physical parameters of the satellite be estimated from the thermal network optimization process?

The possibility of using the optimization process developed for improving the thermal network model predictions to estimate the evolution of the physical parameters of the satellite was also explored in this study. Despite the difficulty of directly associating the optimized parameters with their physical equivalents, tracking the evolution of the optimized parameters could provide insights into the physical changes of the satellite.

The available telemetry data for Delfi-PQ, treated as one dataset for creating the thermal network model, was batched for different time periods and the optimization process was performed on each batch. The results of the optimization process were then compared to possibly identify any trends in the evolution of the physical parameters.

One notable difficulty was discerning the effects of changing solar radiation and degradation over time from other potential influences, given the sparse data and changing deviations from the mean temperature trend due to tumbling. This highlights the need for more data to better distinguish between these effects in future studies.

9.2. Future work recommendations

- The study conducted on FUNcube-1 was based solely on a single day's worth of data. This approach might limit the effectiveness of the base model over a more extended period due to the fluctuating mean temperature trend. Future studies could aim to validate the base model's performance over more extended periods, strengthening the reliability of the results and further reinforcing the model's application.
- Currently, illumination is classified as a binary input, categorizing conditions as either sunlight or eclipse. This method of classification could be improved by considering illumination as a continuous variable instead, thus enabling the incorporation of the changing solar flux effect in the data-driven models.
- In parallel with the enhancement of the illumination variable, the parameters of the

first-principles models for Delfi-PQ could be adjusted over time to accommodate changing conditions. However, this would necessitate more comprehensive data to constrain these parameters effectively and, thus, underline the significance of extensive and precise data collection.

- Future work into improving the code efficiency of the thermal network model would enable the inclusion of more accurate heat input models, improving the correlation between the optimized parameters and the physical parameters.
- If full orbit data is available for Delfi-PQ, the possibility of obtaining an attitude estimate based on the generated voltages of the solar panels can be explored. If this is possible, the attitude estimate can be used to improve the first-principles models.
- Similar to Delfi-PQ, the feasibility of inferring attitude data from solar panel voltages for Delfi-C3 can be explored. With attitude data available, the fluctuations due to tumbling can be accurately characterized, allowing for more advanced filtering of the available data. Due to the large amount of available data, it might be possible to implement the models developed in this thesis on Delfi-C3.

References

- [1] S. W. Asmar and S. Matousek, “Mars cube one (marco) shifting the paradigm in relay deep space operation,” in *14th International Conference on Space Operations*, 2016, p. 2483.
- [2] M. S. F. Bangi and J. S.-I. Kwon, “Deep hybrid modeling of chemical process: Application to hydraulic fracturing,” *Computers and Chemical Engineering*, vol. 134, pp. 106 696–106 696, 2020.
- [3] L. Bauersfeld, E. Kaufmann, P. Foehn, S. Sun, and D. Scaramuzza, “Neurobem: Hybrid aerodynamic quadrotor model,” *arXiv preprint arXiv:2106.08015*, 2021.
- [4] A. Bejan and A. D. Kraus, *Heat Transfer Handbook*. John Wiley & Sons, 2003, ISBN: 978-0-471-39015-2.
- [5] T. L. Bergman, T. L. Bergman, F. P. Incropera, D. P. Dewitt, and A. S. Lavine, *Fundamentals of heat and mass transfer*. John Wiley & Sons, 2011.
- [6] T. L. Bergman and A. S. Lavine, *Fundamentals of heat and mass transfer*. Wiley, 2019.
- [7] P. T. Boggs and J. W. Tolle, “Sequential quadratic programming,” *Acta numerica*, vol. 4, pp. 1–51, 1995.
- [8] J. Bouwmeester, A. Menicucci, and E. Gill, “Improving cubesat reliability: Subsystem redundancy or improved testing?” *Reliability Engineering & System Safety*, vol. 220, p. 108 288, Apr. 2022, ISSN: 09518320. DOI: [10.1016/j.ress.2021.108288](https://doi.org/10.1016/j.ress.2021.108288).
- [9] J. Bouwmeester and J. Guo, “Survey of worldwide pico-and nanosatellite missions, distributions and subsystem technology,” *Acta Astronautica*, vol. 67, no. 7-8, pp. 854–862, 2010.
- [10] J. Cao, Z. Li, and J. Li, “Financial time series forecasting model based on ceemdan and lstm,” *Physica a: Statistical Mechanics and its Applications*, vol. 519, pp. 127–139, 2019.
- [11] D. Chaffart and L. A. Ricardez-Sandoval, “Optimization and control of a thin film growth process: A hybrid first principles/artificial neural network based multiscale modelling approach,” *Computers and Chemical Engineering*, vol. 119, pp. 465–479, 2018.
- [12] V. Chandola, A. Banerjee, and V. Kumar, “Anomaly detection: A survey,” *Acm Computing Surveys (CSUR)*, vol. 41, no. 3, pp. 1–58, 2009.
- [13] V. Cinotti, M. Lusvarghi, F. M. Marchese, and C. Rapisarda, “Delfi-pq thermal model verification,” *AE4S10: Microsat Engineering Report, TU Delft*, 2022.
- [14] J. Colwell, J. Brisset, A. Dove, L. Roe, and J. Blum, “Q-pace: The cubesat particle aggregation and collision experiment,” in *12th Low Cost Planetary Missions Conference*, 2017.
- [15] A. R. Conn, N. I. Gould, and P. L. Toint, *Trust region methods*. SIAM, 2000.

- [16] M. Doyle, R. Dunwoody, G. Finneran, *et al.*, “Mission testing for improved reliability of cubesats,” Z. Sodnik, B. Cugny, and N. Karafolas, Eds., SPIE, Jun. 2021, p. 237, ISBN: 9781510645486. DOI: [10.1117/12.2600305](https://doi.org/10.1117/12.2600305).
- [17] F. Gao and L. Han, “Implementing the nelder-mead simplex algorithm with adaptive parameters,” *Computational Optimization and Applications*, vol. 51, no. 1, pp. 259–277, 2012.
- [18] Y. Gao, T. Yang, M. Xu, and N. Xing, “An unsupervised anomaly detection approach for spacecraft based on normal behavior clustering,” ser. 2012 Fifth International Conference on Intelligent Computation Technology and Automation, 2012, pp. 478–481.
- [19] A. Géron, *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow.*” O’Reilly Media, Inc.”, 2022.
- [20] F. A. Gers, J. Schmidhuber, and F. Cummins, “Learning to forget: Continual prediction with lstm,” *Neural Computation*, vol. 12, no. 10, pp. 2451–2471, 2000.
- [21] E. Gill, G. L. E. Monna, J. M. A. Scherpen, and C. J. M. Verhoeven, “Misat: Designing a series of powerful small satellites based upon micro systems technology,” ser. 58th International Astronautical Congress, vol. 6, 2007, pp. 1–6.
- [22] R. Hamann, J. Bouwmeester, and G. Brouwer, “Delfi-c3 preliminary mission results,” 2009.
- [23] H. Heidt, J. Puig-Suari, A. Moore, S. Nakasuka, and R. Twiggs, “Cubesat: A new generation of picosatellite for education and industry low-cost space experimentation,” 2000.
- [24] S. Hochreiter, “The vanishing gradient problem during learning recurrent neural nets and problem solutions,” *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 6, no. 02, pp. 107–116, 1998.
- [25] K. Hundman, V. Constantinou, C. Laporte, I. Colwell, and T. Soderstrom, “Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding,” ser. Proceedings of the 24th acm SIGKDD International Conference on Knowledge Discovery and Data Mining, 2018, pp. 387–395.
- [26] E. Imre. “Satmad.” (2021), [Online]. Available: <https://github.com/egemenimre/satmad> (visited on 06/09/2023).
- [27] D. Iverson, “Data mining applications for space mission operations system health monitoring,” ser. SpaceOps 2008 Conference, 2008, pp. 3212–3212.
- [28] D. L. Iverson, “Inductive system health monitoring,” ser. International Conference on Artificial Intelligence, 2004.
- [29] C. C. Johnson, T. Quackenbush, T. Sorensen, D. Wingate, and M. D. Killpack, “Using first principles for deep learning and model-based control of soft robots,” *Frontiers in Robotics and AI*, vol. 8, pp. 654 398–654 398, 2021.
- [30] Z. Karevan and J. A. K. Suykens, “Transductive lstm for time-series prediction: An application to weather forecasting,” *Neural Networks*, vol. 125, pp. 1–9, 2020.
- [31] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.
- [32] E. Kulu. “Nanosats database.” (2023), [Online]. Available: <https://www.nanosats.eu/> (visited on 06/09/2023).

- [33] P. Malhotra, L. Vig, G. Shroff, P. Agarwal, *et al.*, “Long short term memory networks for anomaly detection in time series,” ser. Proceedings, vol. 89, 2015, pp. 89–94.
- [34] B. K. Malphrus, K. Z. Brown, J. Garcia, *et al.*, “The lunar icecube em-1 mission: Prospecting the moon for water ice,” *IEEE Aerospace and Electronic Systems Magazine*, vol. 34, no. 4, pp. 6–14, 2019.
- [35] L. R. Medsker and L. C. Jain, “Recurrent neural networks,” *Design and Applications*, vol. 5, pp. 64–67, 2001.
- [36] K. P. Murphy, *Machine learning: a probabilistic perspective*. MIT press, 2012.
- [37] J. Nilsson and S. Riedel, *Electric Circuits, vol. 10*. Prentice Hall International Edition, New Jersey, 2015.
- [38] J. Pinto, M. Mestre, J. Ramos, R. S. Costa, G. Striedner, and R. Oliveira, “A general deep hybrid model for bioreactor systems: Combining first principles with deep neural networks,” *Computers and Chemical Engineering*, pp. 107 952–107 952, 2022.
- [39] C. Poly. “Cubesat design specification.” (2023), [Online]. Available: <https://www.cubesat.org/cubesatinfo> (visited on 06/09/2023).
- [40] D. C. Psychogios and L. H. Ungar, “A hybrid neural network-first principles approach to process modeling,” *AIChE Journal*, vol. 38, no. 10, pp. 1499–1511, 1992.
- [41] D. Racelis and M. Joerger, “High-integrity tle error models for meo and geo satellites,” in *2018 AIAA SPACE and Astronautics Forum and Exposition*, 2018, p. 5241.
- [42] D. Robert and G. D. Hager, “Chapter 21 - deep learning: Rnns and lstm,” in (Handbook of Medical Image Computing and Computer Assisted Intervention), Handbook of Medical Image Computing and Computer Assisted Intervention. Academic Press, Jan. 2020, pp. 503–519. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/B9780128161760000260>.
- [43] P. J. Rousseeuw and A. M. Leroy, *Robust regression and outlier detection*. John wiley & sons, 2005.
- [44] C. Ruiz, “Satellite thermal model for delfipq,” *Research Internship Report, TU Delft*, 2020.
- [45] A. Savitzky and M. J. Golay, “Smoothing and differentiation of data by simplified least squares procedures.,” *Analytical chemistry*, vol. 36, no. 8, pp. 1627–1639, 1964.
- [46] P. Shah, M. Z. Sheriff, M. S. F. Bangi, *et al.*, “Deep neural network-based hybrid modeling and experimental validation for an industry-scale fermentation process: Identification of time-varying dependencies among parameters,” *Chemical Engineering Journal*, vol. 441, pp. 135 643–135 643, 2022.
- [47] D. T. Shipmon, J. M. Gurevitch, P. M. Piselli, and S. T. Edwards, “Time series anomaly detection; detection of anomalous drops with limited features and sparse examples in noisy highly periodic data,” *arXiv preprint arXiv:1708.03665*, 2017.
- [48] S. Siami-Namini, N. Tavakoli, and A. S. Namin, “The performance of lstm and bilstm in forecasting time series,” ser. 2019 IEEE International Conference on Big Data (Big Data), 2019, pp. 3285–3292.
- [49] X. Song, Y. Liu, L. Xue, *et al.*, “Time-series well performance prediction based on long short-term memory (lstm) neural network model,” *Journal of Petroleum Science and Engineering*, vol. 186, pp. 106 682–106 682, 2020.

-
- [50] B. Sun, C. Yang, Y. Wang, W. Gui, I. Craig, and L. Olivier, “A comprehensive hybrid first principles/machine learning modeling framework for complex industrial processes,” *Journal of Process Control*, vol. 86, pp. 30–43, 2020.
 - [51] S. Sun, C. C. de Visser, and Q. Chu, “Quadrotor gray-box model identification from high-speed flight data,” *Journal of Aircraft*, vol. 56, no. 2, pp. 645–661, 2019.
 - [52] T. Villela, C. A. Costa, A. M. Brandão, F. T. Bueno, and R. Leonardi, “Towards the thousandth cubesat: A statistical overview,” *International Journal of Aerospace Engineering*, vol. 2019, pp. 1–13, Jan. 2019, ISSN: 1687-5966. DOI: [10.1155/2019/5063145](https://doi.org/10.1155/2019/5063145).
 - [53] Z. Zhang, “Orbital temperature model construction for delfi-pq,” [Manuscript in preparation].

A

Telemetry fields

All available telemetry fields for Delfi-PQ and FUNcube-1 are listed below:

Delfi-PQ

Field	Name
1	timestamp
2	groundstation
3	TotalUptime
4	BootCounter
5	BatteryGGStatus
6	BatteryINASStatus
7	InternalINASStatus
8	UnregulatedINASStatus
9	Bus1INASStatus
10	Bus2INASStatus
11	Bus3INASStatus
12	Bus4INASStatus
13	PanelYpINASStatus
14	PanelYpTMPStatus
15	PanelYmINASStatus
16	PanelYmTMPStatus
17	PanelXpINASStatus
18	PanelXpTMPStatus
19	PanelXmINASStatus
20	PanelXmTMPStatus
21	InternalINACurrent
22	InternalINAVoltage

23	UnregulatedINACurrent
24	UnregulatedINAVoltage
25	BatteryINAVoltage
26	BatteryINACurrent
27	BatteryGGTemperature
28	BatteryTMP20Temperature
29	Bus4Current
30	Bus4Voltage
31	Bus3Current
32	Bus3Voltage
33	Bus2Current
34	Bus2Voltage
35	Bus1Current
36	Bus1Voltage
37	PanelYpCurrent
38	PanelYpVoltage
39	PanelYmCurrent
40	PanelYmVoltage
41	PanelXpCurrent
42	PanelXpVoltage
43	PanelXmCurrent
44	PanelXmVoltage
45	PanelYpTemperature
46	PanelYmTemperature
47	PanelXpTemperature
48	PanelXmTemperature
49	MCUTemp

FUNcube-1

Field	Name
1	Satellite Date/Time UTC
2	Black Chassis deg. C
3	Silver Chassis deg. C
4	Black Panel deg. C
5	Silver Panel deg. C
6	Solar Panel +X deg. C
7	Solar Panel -X deg. C
8	Solar Panel +Y deg. C
9	Solar Panel -Y deg. C
10	Solar Panel X mV
11	Solar Panel Y mV
12	Solar Panel Z mV
13	Tot. Photo Curr. mA
14	Battery mV
15	Tot. System Curr. mA

B

Delfi-PQ thermal network

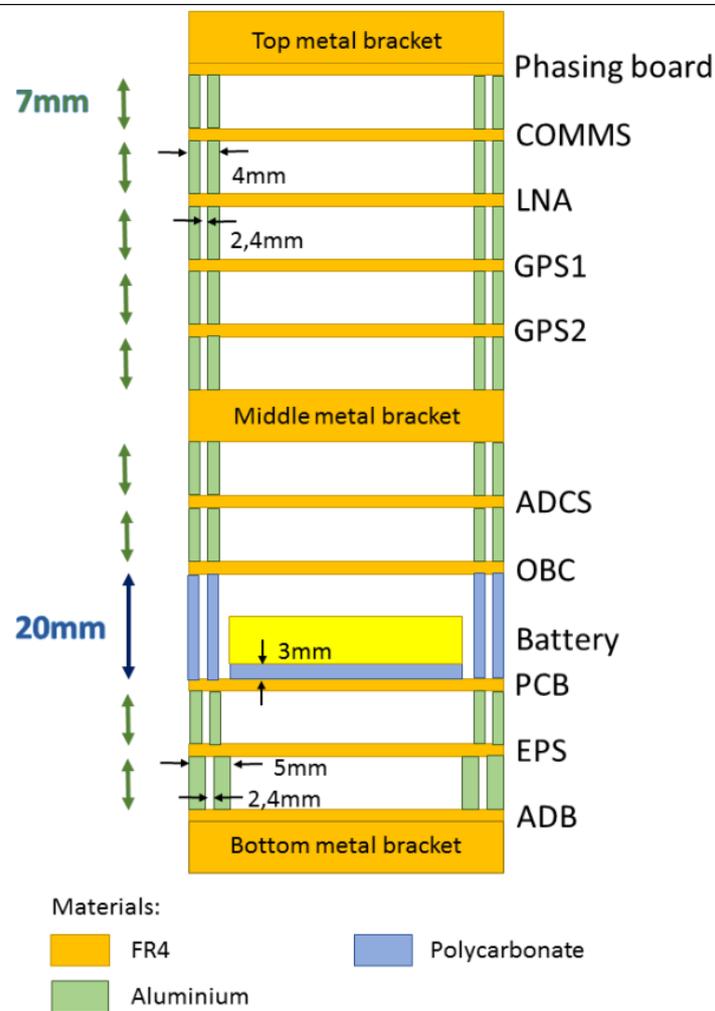


Figure B.1: Internal stack of Delfi-PQ [44]

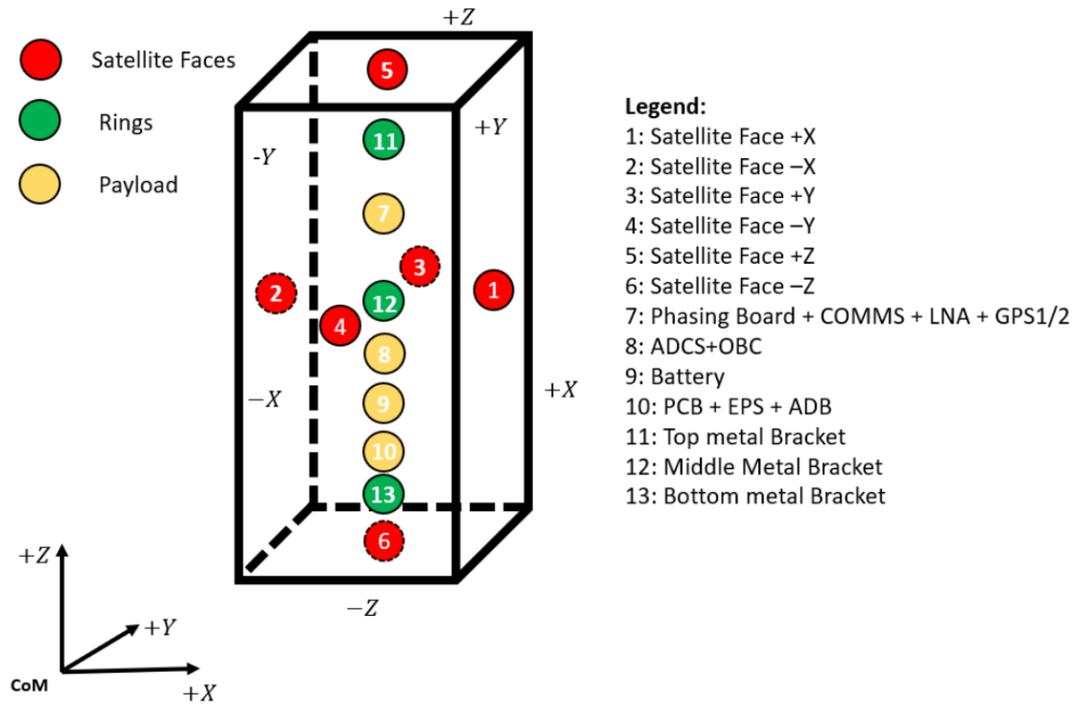


Figure B.2: Location of nodes of the 13-node thermal network model [13]

C

Code

All Python scripts and Jupyter notebooks used for this thesis are available on GitHub at <https://github.com/ullas-bhat/master-thesis>.

The installable Python package `SSA-Calc`, useful for calculating the angles required for orbit overlaying is available at <https://github.com/ullas-bhat/master-thesis/tree/main/ssa-calc>