DOI: 10.13009/EUCASS2025-017
11TH EUROPEAN CONFERENCE FOR AERONAUTICS AND SPACE SCIENCES (EUCASS)

Accelerating Space Mission Thermal Analysis: a Deep Learning Framework for Rapid Temperature Prediction in the Vega Rocket Roll and Attitude Control System

Giuseppe Bernabei[⋆†] *Carlo Laurenti Argento*[⋆]
[⋆]*AVIO S.p.A*
Colleferro, Italy
giuseppe.bernabei@avio.com carlo.laurentiargento@avio.com
[†]Corresponding author

Abstract

This study introduces a Deep Learning framework for predicting the evolution of the temperatures of the critical components of the Roll and Attitude Control System of the Vega family rockets. This system consists of six thrusters that are fired to control the roll and attitude of the rocket throughout the entire mission. Traditional Lumped Parameter Models simulations require approximately 3 days of computation time to model 12 critical temperatures (6 thrust chambers and 6 flow control valves) across an 8000-second timeline. Our Deep Learning approach directly processes the thrusters firing profiles to predict the mentioned temperature profiles, achieving a reduction of 99.54% in computational time (inference down from 3 days to less than 15 minutes) while maintaining very high accuracy with mean absolute errors below 3°C across the mission timeline, for all temperature predictions. The neural network successfully captures the complex thermal dynamics and heat transfer phenomena, demonstrating that machine learning can effectively replace computationally expensive simulations for thermal mission planning applications in aerospace engineering.

1. Introduction

Thermal analysis is a critical component in space mission planning and systems engineering. For launch vehicles such as the Vega C, accurate temperature predictions of key systems are essential to ensure mission success and overall system reliability. Traditional approaches require significant computational resources and time, creating bottlenecks in the mission design process.

One of such systems is Vega C's Roll and Attitude Control System (RACS), which maintains the orientation and stability of the rocket throughout the mission. In the Vega C launch vehicle, the RACS operates as part of the AVUM+ module (see Fig. 1). Thermal analysis for this system presents a significant computational bottleneck that limits design iteration and optimization, and this paper addresses the challenge of reducing computation time for RACS thermal analysis.

This paper introduces an innovative Deep Learning (DL) framework that leverages modern neural network architectures to learn the complex relationships between thruster firing profiles and the resulting temperature evolution in some critical components of the RACS. This framework enables rapid and accurate predictions that replace traditional simulations that require days of computation time.

2. Background

Accurate prediction of the thermal behavior of the RACS components is paramount to ensure reliable operation during launch sequences, orbital maneuvers, and payload deployment. This section outlines the motivation behind this research, the limitations of current approaches, and the objectives that this paper aims to achieve.

2.1 Motivation

Thermal analysis is a fundamental requirement in space mission design, as extreme temperature variations can severely impact component reliability and mission success. These analyses have traditionally been performed with computa-

DOI: 10.13009/EUCASS2025-017

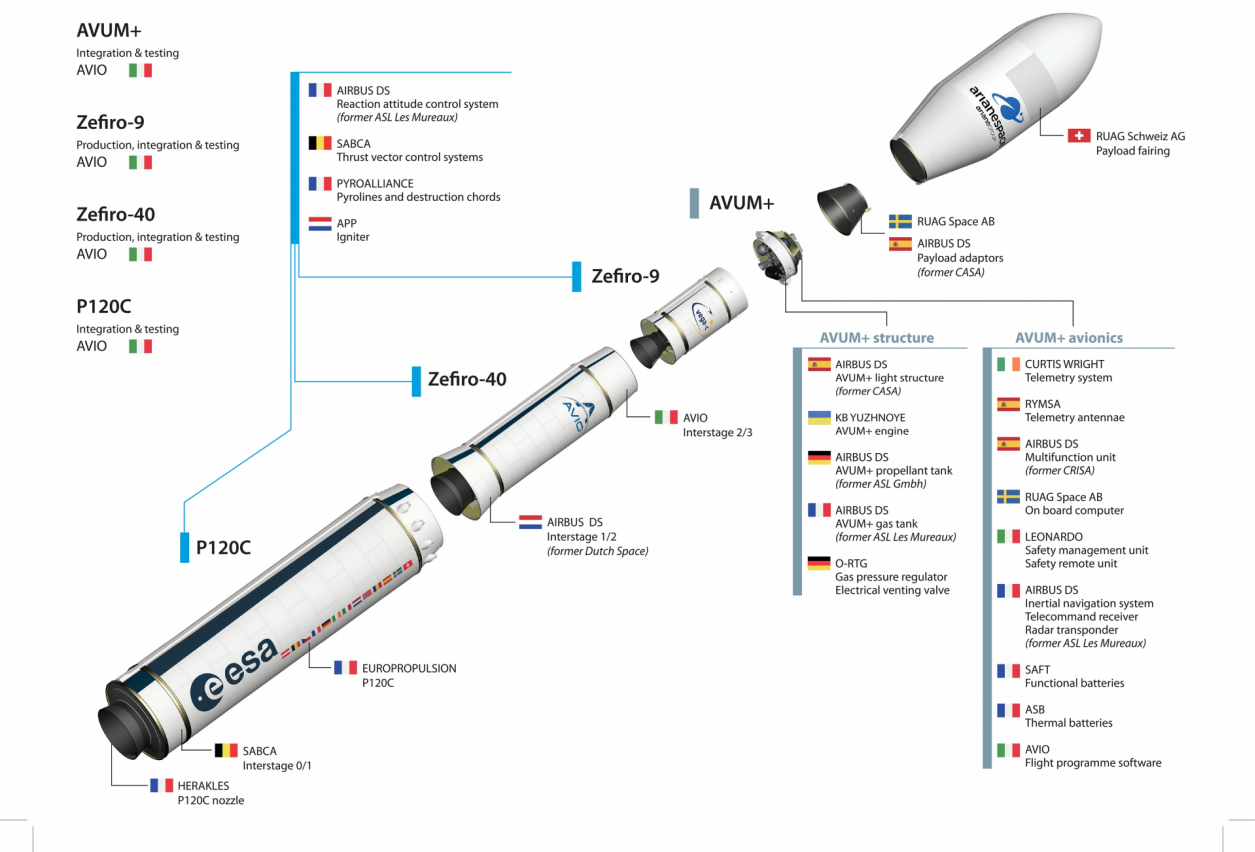

Figure 1: The Vega C Launch Vehicle

tionally intensive Lumped Parameter Models (LPMs) simulations. The current reliance on LPM simulations, while accurate, imposes substantial delays on the iterative design process. With computation times of approximately 3 days for a single mission profile simulation, engineers face significant constraints when evaluating multiple mission scenarios or performing design optimizations. This computational burden limits the number of design iterations and thermal analysis cycles that can realistically be performed during mission planning phases.

2.2 Limitations of Traditional Approaches

The historical approach based on LPMs provides high-fidelity results, but suffers from several notable limitations:

1. Computational expense: current LPM techniques used in Avio require approximately 3 days of computation time to model just 12 critical temperatures across the mission timeline. These simulations also create resource bottlenecks as the computing infrastructure is tied up for extended periods.

2. Iteration constraints: the lengthy simulation times restrict the number of what-if analyses that can be evaluated within practical development time frames.

3. Expertise dependencies: traditional thermal modeling requires specialized expertise in both thermal physics and simulation software, creating potential workflow bottlenecks.

These limitations become particularly problematic during the later stages of mission planning when multiple configuration changes may need rapid evaluation or when unforeseen issues require rapid thermal analysis to support decision making.

DOI: 10.13009/EUCASS2025-017

2.3 Research Objectives

This research aims to address the limitations mentioned above by developing a DL framework that can dramatically accelerate thermal analysis while maintaining prediction accuracy. Specifically, the goal of this paper is to:

1. Reduce computational time: develop a model capable of reducing thermal prediction time from days to minutes, allowing rapid iteration in the design process.
2. Maintain high accuracy: achieve temperature predictions with accuracy levels comparable to traditional LPMs results.
3. Specific Modeling: develop specialized neural network architectures that are tailored to the distinct thermal behaviors of different RACS components.
4. Practical generalizable integration: design a solution that can be easily and generically integrated into existing mission planning workflows, requiring minimal specialized expertise to operate.

By achieving these objectives, this research aims to fundamentally transform the thermal analysis process for space systems, enabling faster iteration cycles and ultimately more reliable and optimized rocket systems.

3. System Analysis and Description

The RACS of the Vega rockets consists of 6 thrusters strategically positioned around the rocket body in two modules. Each module has 2 thrusters dedicated to controlling roll and 1 to controlling pitch, as shown in Fig. 2. The critical temperatures are those inside the thrust chambers and those in proximity of the flow control valves, which control the flow of propellant that goes inside the thrust chambers.

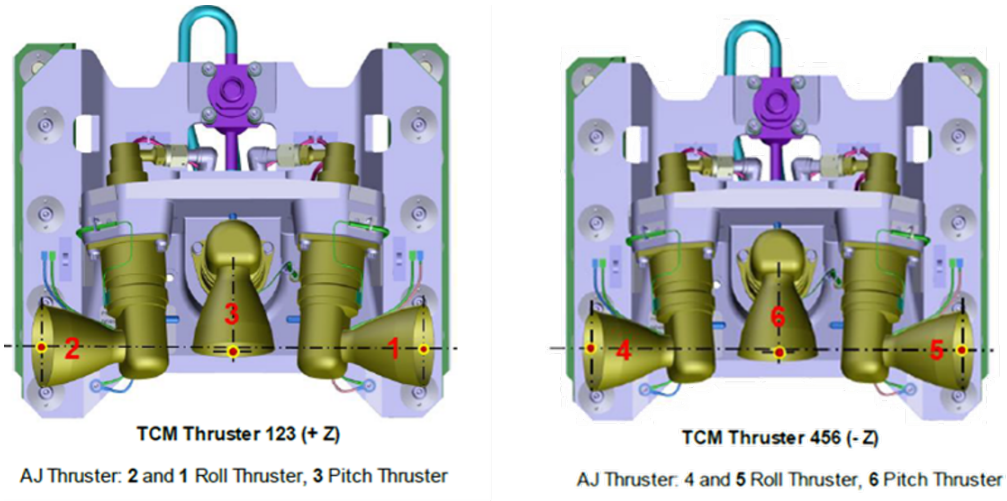

Figure 2: Vega RACS Thrusters

In this study, data from 15 thermal simulations were analyzed in six distinct flights, including both the Vega and Vega C missions. The focus was specifically on modeling the thermal profiles of the thrust chambers, as these components experience the most significant temperature variations during operation. During a typical mission, these thrusters are fired in specific patterns determined by the guidance, navigation, and control (GNC) engineering department.

Each thruster firing generates heat that affects both the operating thruster and neighboring components through thermal conduction and radiation, making accurate temperature monitoring critical for mission safety and component reliability. The thermal behavior of each component is determined by several critical parameters, including:

1. The duration of thruster firings
2. The frequency of thruster firings
3. The varying external thermal environment throughout the mission

DOI: 10.13009/EUCASS2025-017

4. The specific physical properties of the components

The dataset used in this study consists of raw firing sequences provided by the GNC department and the corresponding temperature predictions produced by the thermal department. An important challenge in working with the data was the time representation differences: the firing profiles were provided as a sequence of discrete firing events, with each event specified by a start and end time, while the thermal data were time series which featured varying timesteps depending on the particular mission phases.

An extensive analysis of the dataset revealed several critical characteristics that necessitated careful consideration during the design of the neural network architecture:

1. Thrust chambers exhibit relatively rapid thermal responses to firings, with temperatures raising rapidly in proximity to firing events and then gradually decreasing during rest phases.

2. There are clear correlations between firing patterns and temperature profiles, but these relationships are highly non-linear and component-specific.

3. Temperature profiles contain both short-term responses to individual firings and long-term cumulative effects.

These observations proved crucial in guiding our selection of appropriate deep learning architectures for different components of the system, allowing us to achieve accurate temperature predictions while dramatically reducing computational time compared to traditional LPM simulations.

4. Data Preparation and Deep Learning Model

Analyzing the system alongside the data suggested two possible strategies. The first strategy considers the time series in its entirety and takes into account the effect of past events on the temperature estimation. For this reason, the neural network chosen is a Long Short Term Memory (LSTM) network combined with a data preparation aimed at taking into account the effect of past events according to their position in the timeline. The second approach arises from the assumption that the temperature variation is solely a function of the duration of the fire event. For this approach, a Multi Layer Perceptron (MLP) network combined with data preparation focused on showing the relationship between the duration of the single fire event and the variation it induced into the components' temperature was developed.

4.1 Windowed Approach with LSTM

LSTM networks have emerged as a powerful deep learning architecture for sequential data prediction problems, particularly for time series with complex temporal dependencies. First introduced by Hochreiter and Schmidhuber (1997),[1] LSTMs address the vanishing gradient problem typical of traditional recurrent neural networks, enabling effective capture of long-range dependencies in sequential data without learning stagnation. These networks have demonstrated exceptional performance in time series forecasting applications (Siami-Namini et al., 2018),[4] making them particularly well suited for modeling the thermal behaviors of aerospace components.

Although LSTMs have been successfully applied in engineering domains such as reduced order modeling for turbulent flows (Mohan and Gaitonde, 2020),[2] this research on thermal modeling of rocket thruster components represents a novel application. This approach leverages the LSTM architecture's capacity to simultaneously model both immediate thermal responses and the cumulative effects that characterize the complex heat-transfer mechanisms of Vega's RACS.

4.1.1 Data Preparation

The data preparation process involves several key steps that transform raw thruster firing profiles and raw temperature time series into a format suitable for LSTM training. This preparation is crucial to enable the model to learn the heating patterns of the components.

First, the raw firing data is transformed into time series data. Then, the newly formed time series, along with the temperatures' time series, are split into sequences, where each sequence represents a segment of a rocket mission's timeline. Furthermore, to have a rigid time context, a common timeline is built. Finally, the following features are extracted for each timestep:

1. Cumulative firing durations across multiple time steps

2. Cumulative firing counts across multiple time steps

DOI: 10.13009/EUCASS2025-017

3. Temperature measurements

4. Temperature differentials

A critical aspect of the data preparation methodology is the windowing approach. The sequences in the previous step were extracted as overlapping windows of fixed length (default 120 time steps, corresponding to 120 seconds). These structured data points would then serve as input sequences for the LSTM model. For feature engineering, 6 features were derived from the firing profiles based on 3 distinct time windows: T_1, T_2, T_3. Specifically, at each time step along the common timeline, these features captured both the cumulative firing durations and the cumulative firing counts in the previous T_1, T_2, T_3 seconds, respectively. Through iterative data visualization and model evaluation, one could clearly determine that temperature evolutions exhibited stronger dependencies on events that occurred in the near past, and this led us to establish the time window values at 10, 20 and 50 seconds.

Lastly, each input sequence is paired with the temperature differential $T(t) - T(t+1)$, which constitutes the final target value to predict. Recognizing that this target variable is inherently correlated with the current temperature state, we incorporated $T(t)$ as a seventh feature in the input sequences to provide essential contextual information. To improve training stability and model performance, all features were scaled to map their value to a range of [0,1].

Figure 3 illustrates the comprehensive data processing pipeline: the normalized temperature profile (blue "Temperature"), the normalized variations of consecutive temperature readings (green "Normalized deltaT" stars). The paired vertical red and green lines represent the beginning ("Fire Start") and end ("Fire End") of each thrusters' activation, providing visual context for investigating the thermal response patterns.

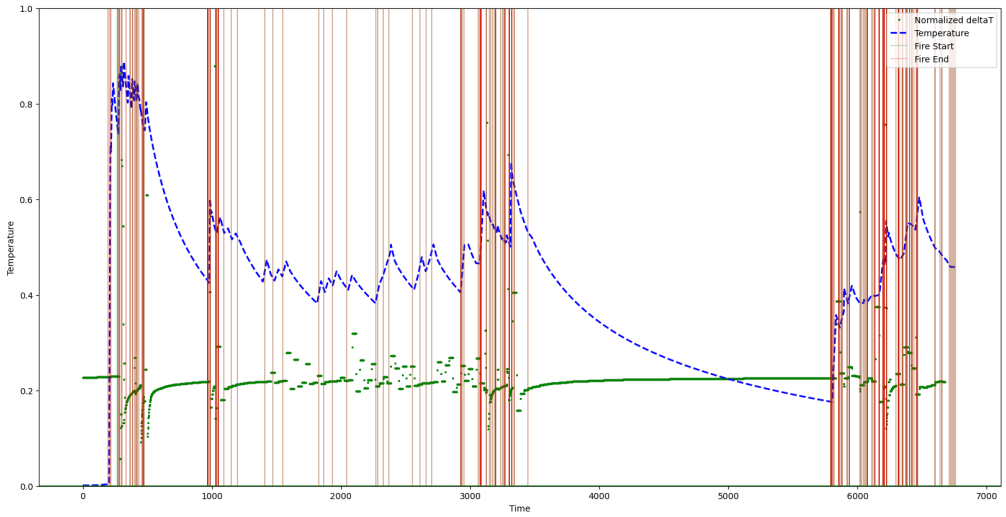

Figure 3: The Evolution of the Temperature of a Thruster Chamber plotted with its corresponding Firing Profile

4.1.2 Deep Learning Model

The model is built on the LSTM architecture, which is particularly well suited for this application due to its ability to capture both short-term and long-term dependencies in time series. The LSTM model is implemented in PyTorch and consists of an LSTM layer with 12 hidden units and 4 fully connected layers with decreasing dimensions: 64, 32, 16 and 1. The activation function is a ReLU between the dense and dropout layers (15% dropout rate).

The network weights are initialized according to the Xavier uniform technique, which was chosen to help maintain proper scaling of gradients throughout the network. For training, we employ Adam optimizer with a small learning rate (10^{-5}) and a step learning rate scheduler that reduced the learning rate by a factor of 0.1 every 15 epochs. The loss function is an error function that computes the sum of squared differences between the predictions and the targets.

The training process incorporates early stopping with a patience of 5 epochs to prevent overfitting. During each epoch, the model is trained on batches of sequences, with gradients updated after each batch. After each epoch, the model is evaluated on the validation set, and the best-performing model is chosen as the best-performing model on the validation set.

DOI: 10.13009/EUCASS2025-017

4.1.3 Results

To train the model, the data was divided into training set (80%) and validation set (20%). Two flights were kept on the side as the final test set. Ultimately, the goal was for the model to focus on distinct prediction horizons and to learn the temporal patterns in the data.

In terms of computational performance, this approach achieved a remarkable 99.54% reduction in computation time compared to traditional LPM techniques. These conventional methods required approximately 3 days to simulate the RACS temperatures on a standard Avio work station across a full mission timeline. On the other hand, the LSTM generates predictions in less than 10 minutes (on the same hardware configuration).

However, the LSTM model demonstrated significant shortcomings in terms of generalization. For mission profiles similar to those in the training dataset, the model performed adequately, capturing temperature spikes during thruster firings and accurately describing cooling between events, but for profiles that deviated substantially from training examples, prediction accuracy deteriorated greatly. This performance gap persisted despite extensive hyperparameter tuning, architecture modifications, and regularization attempts.

Ultimately, the promising evolution of the validation loss did not translate into robust performance on novel mission profiles. This suggests that the complex LSTM architecture, while theoretically capable of modeling temporal dynamics, was capturing noise rather than underlying patterns in the limited dataset.

After multiple test runs, the conclusion was that the LSTM consistently showed signs of overfitting. This can be attributed primarily to two factors:

1. Limited dataset: the computational complexity of physics-based simulations significantly constrains the number of simulations that are run annually. This limitation in the training corpus makes the capture of patterns across diverse mission scenarios harder.

2. High variability: rocket missions are characterized by highly variable thruster firing sequences, flight configurations, and external conditions. This inherent complexity demands a significantly more representative data processing method to make the LSTM more robust to generalization.

These findings led us to conclude that while the LSTM approach offers computational advantages, it is not robust enough for operational deployment. The results point towards the need for a more informed data processing pipeline and simpler model architecture. This led us to think that a smaller multilayer perceptron (MLP) might better generalize from limited training data while maintaining the computational efficiency benefits of neural network approaches.

Our experience with the LSTM implementation demonstrates an important lesson in applied machine learning: more complex models are not always better, particularly when training data is limited and not necessarily representative of the whole phenomenon.

4.2 Single Fire Event Approach with MLP

The results from the previous section point towards the need for a more informed data processing pipeline and simpler model architecture. This led us to the smaller multilayer perceptron (MLP) architecture. Introduced by Rumelhart et al.,[3] the MLP is a neural network that could potentially better generalize from limited training data while maintaining the computational efficiency showcased in the previous section.

This second approach is based on the idea that, under the assumption of constant heat flow, the temperature variation in a given component depends solely on the duration for which the component is exposed to the heat source. This simplifying assumption does not aim to explain the entire process but rather to guide the artificial intelligence tools towards a potential solution. The underlying concept is that these tools will capture all the factors neglected due to the introduced simplifications. For example, no distinction is made between different modes of heat transfer. These same assumptions apply during the cooling phase, when components not exposed to active heat sources gradually approach ambient temperature.

Two distinct models have thus been developed: one for the heating process, which occurs when the component is exposed to the heat source emitted by the RACS thrusters, and one for the cooling process, which takes place when the thrusters are turned off. Beyond the benefits mentioned above, this approach enables effective data augmentation for the training step. In fact, unlike the method in the previous section, which treated fixed-length sequences, this solution decomposes each time series into individual thruster firing events, with each event constituting a separate training sample. This decomposition strategy significantly expands the available distinct data by more than doubling the data points.

DOI: 10.13009/EUCASS2025-017

4.2.1 Data Preparation

The primary objective of the data preparation process is to identify the temperature variation for each fire event and for each rest event. However, this is not feasible for all fire events, because the number of fire events occurring between two consecutive temperature recordings is variable. In fact, the frequency of the temperature measurements does not always match the occurrences of fires.

To systematically identify all thermal events (heating and cooling phases), we developed an algorithm that precisely determines the start and end temperatures for each event. All events for which the start and end temperatures can be defined via linear interpolation were selected, the others were discarded. To correctly apply the linear interpolation method, the temperature gradient between the two interpolation points must remain constant over time. This condition requires both points to be within or outside the same fire event.

Take, for example, the fire event described by the vertical lines at time t_{start} and t_{end} in Fig. 4, where the temperature was registered 3 different times as T_1, T_2 and T_3, at respectively t_1, t_2 and t_3. When extracting a data point from this particular fire event, one would look for the start time, end time, and duration time of this fire event. If the record T_2 did not exist, this firing event could not be claimed, since interpolating linearly T_1 with T_3 would lead to the orange $T_{interpolation}$, which is clearly physically wrong. If instead T_2 were recorded, one could interpolate linearly T_1 with T_2, thus obtaining the purple $T_{interpolation}$. This interpolated record, as opposed to the orange one, would be physically correct. The latter case would lead to a claimable data point where the start time is the purple $T_{interpolation}$, the end time is T_3 and the duration is $t_3 - t_{start}$.

It is understood that the temperature does not vary linearly over time, but tends toward the thermal equilibrium temperature with the external environment, which is unknown. Through this process, we are able to select a subset of events for which the start and end temperatures can be approximated with reasonable accuracy and without breaking any physical law. Finally, knowing the duration of each event, all necessary data are available for training the network.

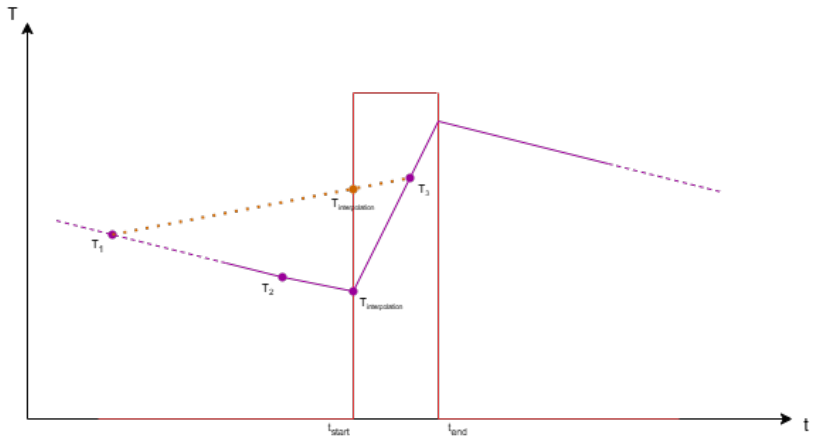

Figure 4: Firing Temperature Interpolation Process

4.2.2 Deep Learning Model

The approach under analysis aims to identify the relationship between the duration of an event, the initial temperature, and the final temperature, with the primary objective of predicting the final temperature given the other two variables. A 2 dimension input neural network with four hidden layers of decreasing size (128, 64, 32, 16), all using ReLU activation, and a single output neuron was chosen. The network is trained using the Adam optimizer and the mean absolute error (MAE) loss function. Training is optimized through callbacks that reduce the learning rate if validation performance plateaus, stop training early to prevent overfitting, and save the best model.

Following the split used in the previous section, the data from a simulation campaign performed for 14 flight configurations were divided into a training set (80%) and a validation set (20%). Two flights were kept on the side as the final test set. Below, the reader can find the training and validation losses for the fire (Fig. 5) and rest (Fig. 6) models.

DOI: 10.13009/EUCASS2025-017

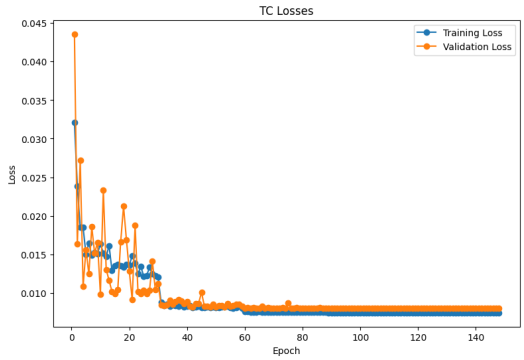

Figure 5: Fire Model Loss Trend

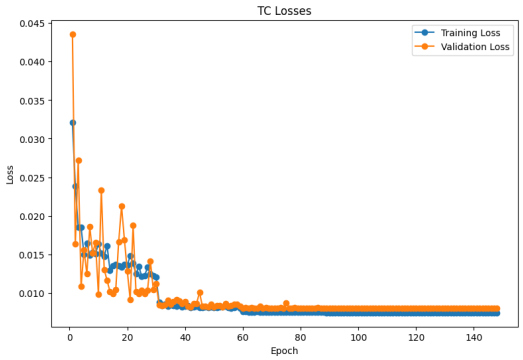

Figure 6: Rest Model Loss Trend

4.2.3 Results

The results obtained for the Roll thrust chambers (TCs) for the 2 test flights are reported below. The following graphs compare the predictions obtained by the LPM method with the prediction obtained by the model introduced in this paper. Note that the input used to predict each profile is always the same: the corresponding raw firing profile of the GNC department for that TC.

In regions without events, especially during long cooling periods, the discrepancy between the prediction and the actual values is simply due to a visualization choice. As shown in the figure, the predicted temperature at the start and end of the event is connected by a straight line segment, whereas in reality, this segment's behavior is asymptotic to the external temperature. As previously explained, the temperature prediction is made only at the start and end points, so what happens between these two points is not taken into account by the network.

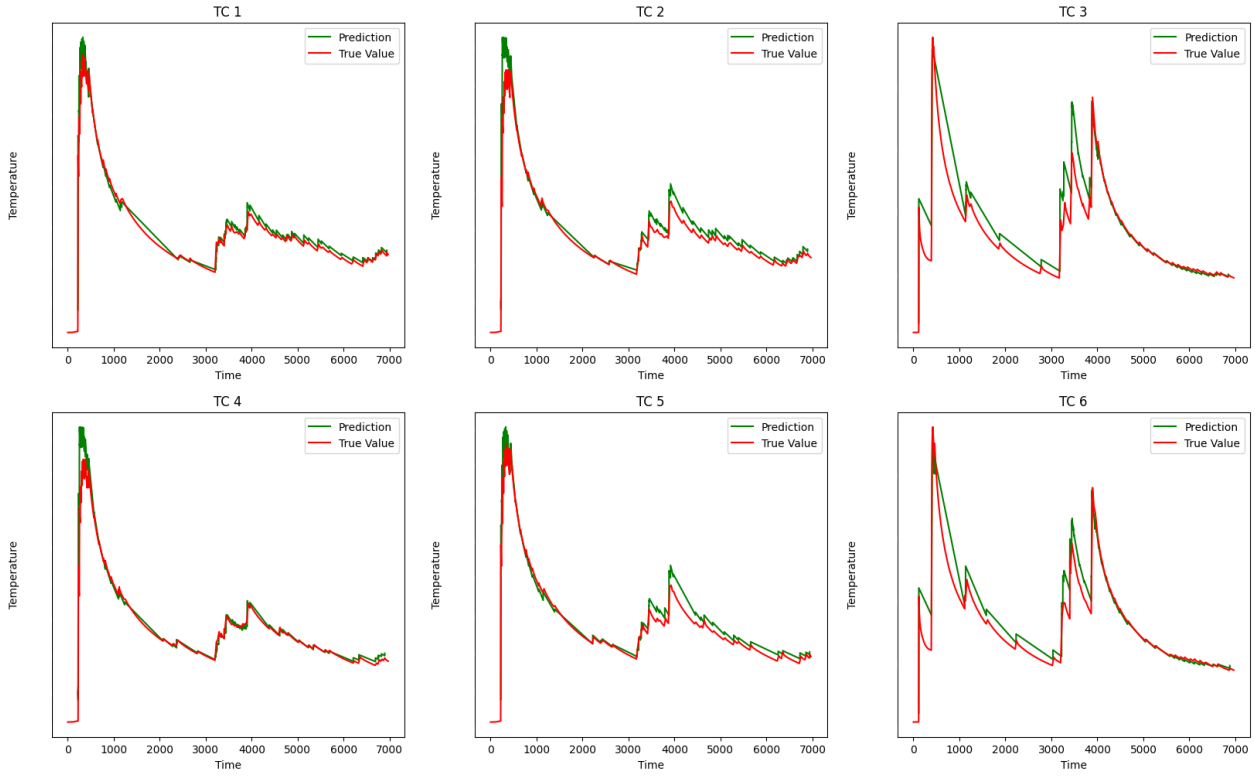

Figure 7: Flight A Results

DOI: 10.13009/EUCASS2025-017

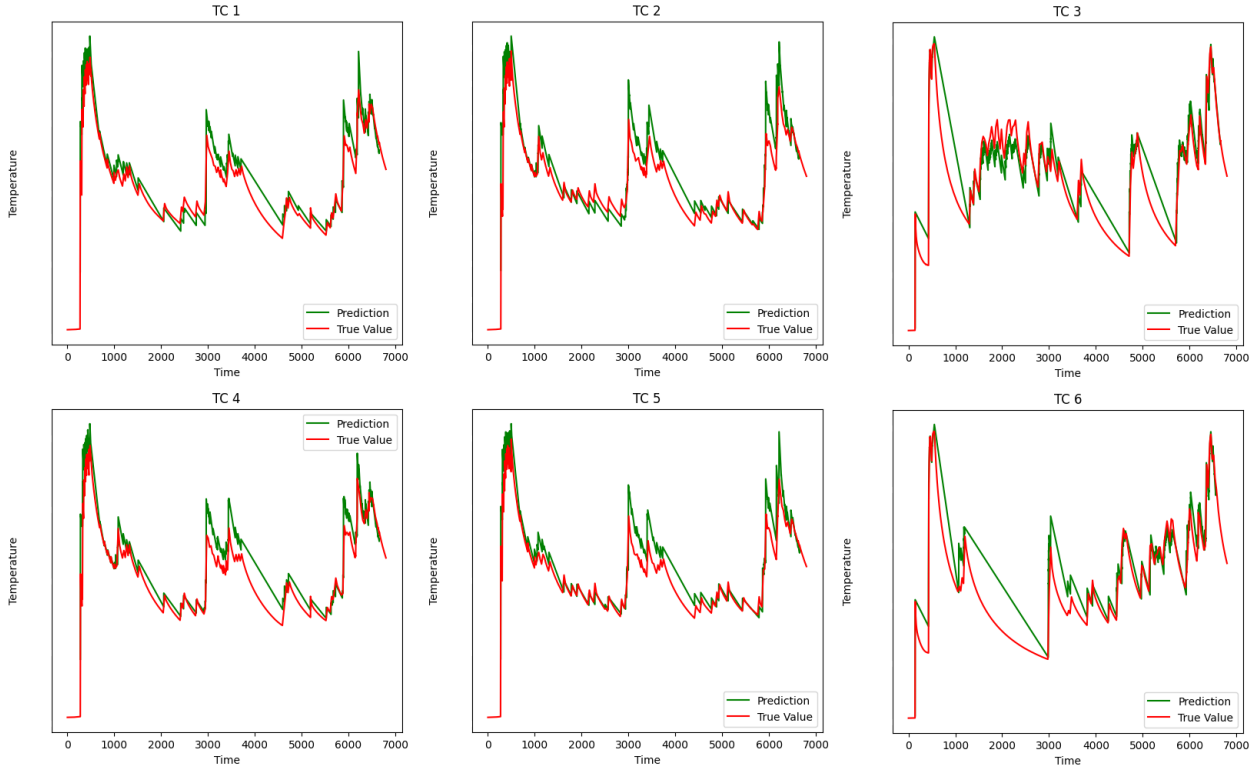

Figure 8: Flight B Results

As can be seen in the graph, the neural network was able to predict the temperature trend with a high degree of accuracy. Another important detail to pay attention to is the network's ability to achieve good results in terms of forecasting for very different temperature profile trends (the 2 test flights do indeed show different profiles overall). The great advantage of using AI is related to the reduction of computational time needed to generate the forecasts. The neural network takes less than 10 minutes to generate the profile compared to about the 3 days needed by the LPM method. Therefore, similarly to the LSTM model, the single fire approach achieved a 99.54% reduction in computation time compared to the LPM simulation.

These results appear to be very promising, especially considering the minimum size of the initial dataset and the reduction of the time needed to obtain the prediction.

5. Conclusions

This study has demonstrated the significant potential of deep learning techniques to accelerate thermal analysis in space mission planning. By developing specialized neural network architectures that were tailored to the thermal behavior of different components of RACS, we achieved a remarkable reduction of 99.54% in computational time while maintaining high prediction accuracy.

The proposed framework enables rapid iteration in mission design and analysis processes, allowing engineers to quickly evaluate the thermal implications of different trajectory and control strategies. This capability is particularly valuable during the early phases of mission planning, where multiple scenarios must be efficiently evaluated.

Future work will focus on extending the framework to additional rocket subsystems, incorporating uncertainty quantification, and potentially developing hybrid models that combine physics-based simulations with deep learning for even greater accuracy and efficiency. The principles and approaches demonstrated in this study have broad applicability in aerospace engineering and may contribute to similar advancements in other fields requiring complex physics simulations.

6. Disclaimer

In accordance with safeguarding Avio's intellectual property, all plots presented in this paper have been appropriately normalized or modified. These adjustments protect proprietary information while preserving the integrity of scientific

DOI: 10.13009/EUCASS2025-017

findings and methodological rigor.

References

[1] Sepp Hochreiter and Jurgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 11 1997.

[2] Arvind T. Mohan and Datta V. Gaitonde. A deep learning based approach to reduced order modeling for turbulent flow control using lstm neural networks, 2018.

[3] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323:533–536, 1986.

[4] Sima Siami-Namini, Neda Tavakoli, and Akbar Siami Namin. A comparison of arima and lstm in forecasting time series. In *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 1394–1401, 2018.