AESim: A Data-Driven Aircraft Engine Simulator

Abdellah Madane[1,2], **Florent Forest**[4], **Hanane Azzag**[3], **Mustapha Lebbah**[1], **Jérôme Lacaille**[2]
[1]David Lab, UVSQ, Paris-Saclay University, 78035 Versailles, France
[2]DataLab, Safran Aircraft Engines, 77550 Moissy-Cramayel , France
[3]LIPN, UMR CNRS 7030, Sorbonne Paris Nord University, 93430 Villetaneuse, France
[4]IMOS Lab, EPFL, Lausanne, Switzerland
abdellah.madane@uvsq.fr, florent.forest@epfl.ch, azzag@univ-paris13.fr, mustapha.lebbah@uvsq.fr, jerome.lacaille@safrangroup.com

Abstract

We present AESim, a data-driven Aircraft Engine Simulator developed using transformer-based conditional generative adversarial networks. AESim generates samples of aircraft engine sensor measurements over full flights, conditioned on a given flight mission profile representing the flight conditions. It constitutes an essential tool in aircraft engine digital twins, capable of simulating their performance for different flight missions. It allows for comparison of the behavior of different engines under the same operational conditions, simulation of various scenarios for a given engine, facilitating applications like engine behavior analysis, performance limit identification, and optimization of maintenance schedules within a global Prognostics and Health Management strategy. It also allows missing flight data imputation and addresses confidentiality concerns by generating synthetic flight datasets that can be shared for public research purposes or data challenges.

1 Introduction

Amidst current technological breakthroughs, numerous industries progressively adopt digital twins to build digital representations of their real-world systems. Digital twins are detailed replications of the physical world within interconnected digital models. They assist industries in refining decision-making, enhancing health monitoring, optimizing process design, and strengthening quality control, among other areas [Thelen *et al.*, 2022; Errandonea *et al.*, 2020].

The analysis and simulation of aircraft engine behavior have garnered significant attention in the aeronautical industry [Wang *et al.*, 2017; Kim *et al.*, 2020], primarily due to its performance, maintenance, safety, and sustainability implications. In the context of aircraft engines, sensors equipped within the engines capture so-called Continuous Engine Operational Data (CEOD) [Forest *et al.*, 2018], which are multivariate time series (MTS) over entire flights. These measurements are vital for Prognostics and Health Management (PHM) applications [Coussirou *et al.*, 2022; Lacaille and Langhendries, 2022; Forest *et al.*, 2020]. However, missing data may occur due to sensor or data transfer failures or delays, leading to incomplete or biased information. Furthermore, it is challenging for aerospace companies to collaborate with academic partners from university laboratories to exploit this data in research projects, as it remains the property of airlines and is subject to contractual obligations, preventing them from being exchanged. Academic research requires opening the data for method validation and reproducibility. At Safran Aircraft Engines, we demonstrated the effectiveness of leveraging CEOD collected from the engines after each flight to build a representative model of the engine using conditional generative models. Subject to flight conditions and control settings, this digital twin tool makes it possible to simulate the engine's behavior if it had carried out the simulated flight mission. It makes it possible to provide CEOD of realistic simulated flights as performed by actual engines.

In this demonstration, we present AESim, an Aircraft Engine Simulator capable of reproducing the behavior of a real aircraft engine by replicating the complex engine dynamics. It addresses the challenges of missing data and industrial confidentiality constraints by generating realistic, simulated engine data for hypothetical missions. This capability enables the comparison of various engine behaviors under identical conditions. It provides valuable datasets for in-house and academic research, sidestepping confidentiality concerns and allowing for the development of robust PHM methods without relying on sensitive real-world data. AESim is based on a novel framework implementing our new architecture for Multivariate Time Series Conditional Generative Adversarial Nets (MTS-CGAN) [Madane *et al.*, 2023; Madane and Lacaille, 2023; Mirza and Osindero, 2014] and extending it to simulate the behavior of aircraft engines.

2 Case Study

The simulator is trained on continuous engine data recorded by aircraft engines belonging to the same fleet. Onboard sensors collect multiple measurements at different frequencies. We process them at a uniform frequency of 1 Hz. Throughout this demo, we present the generation of three parameters: Low-pressure rotor speed (N1), temperature at the entry of the combustion chamber (T) and Exhaust Gas Temperature (EGT), based on five external condition parameters, as shown in Figure 1, representing the simulated flight mission profile which are the ambient temperature, altitude, Mach number, Throttle Lever Angle and a boolean variable (not represented on the figure) indicating whether the engine is running or not.

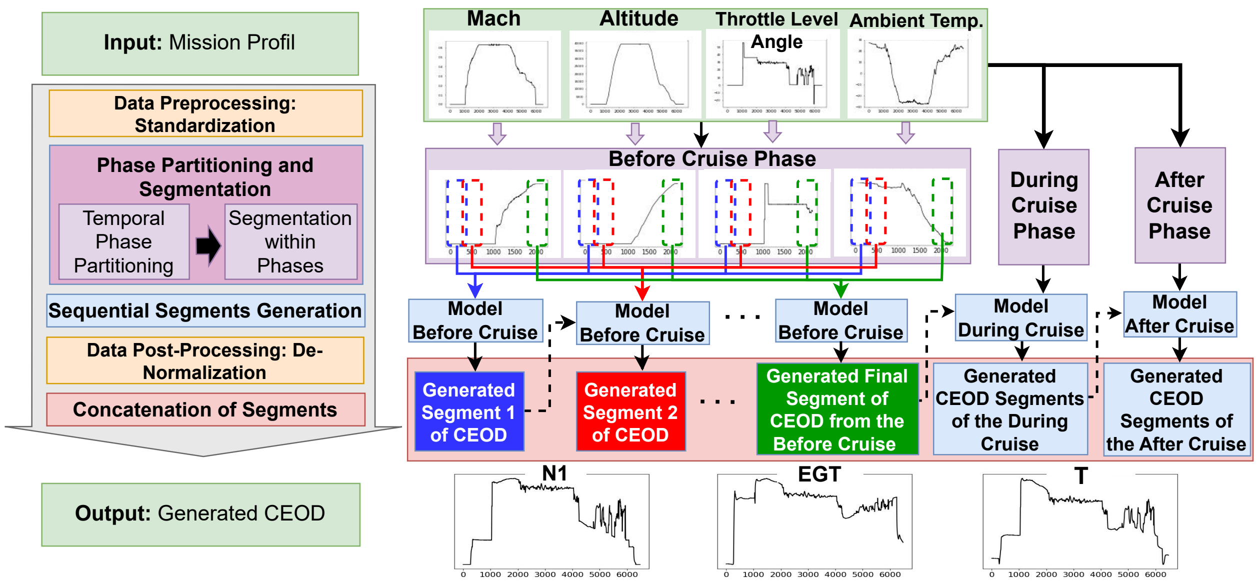


Figure 1: AESim, our proposed data-driven Aircraft Engine Simulator framework. CEOD: Continuous Engine Operational Data. N1: Low-pressure rotor speed. T: temperature before combustion chamber. EGT: Exhaust Gas Temperature.

3 System Framework

3.1 Methodology

We present an overview of our simulator framework in Figure 1, where a mission profile is used as input and the generated CEOD parameters represent the output. The workflow involves several steps:

Standardization of the input. The raw multivariate time series data are normalized by subtracting the mean and dividing by the standard deviation of each feature to ensure equal contribution to the analysis.

Temporal Phase Partitioning. Continuous Engine Data can be extremely long, have variable lengths, and have different phases, resulting in different engine operational states. Multivariate time series are divided into distinct phases reflective of the multiple temporal dynamics within each flight (before, during, after Cruise). This partitioning facilitates the isolation of periods that exhibit homogeneous characteristics, enabling more targeted modeling in subsequent steps.

Segmentation within Phases. Each identified phase is further divided into fixed-duration segments, with each segment spanning 300 seconds and with a 20-timesteps overlap between consecutive segments. This segmentation strategy facilitates processing time series data with variable lengths, including extremely long flights, by eliminating restrictions on the input shape. By dividing the data into manageable segments, the approach ensures that the simulator can accommodate sequences of any length without pre-specifying a fixed input dimension. This flexibility is crucial since flight data have diverse temporal scales, and their length cannot be predetermined. The selection of segment length was informed by expert recommendations, reflecting a careful consideration of the engine's response to local physical phenomena, which typically is, at most, a five-minute duration. Additionally, this decision was guided by a balance between the retention of sufficient information to enable effective data generation and the computational feasibility, particularly due to the quadratic complexity of self-attention mechanisms with respect to the sequence length, making it computationally expensive for long sequences. This approach ensures that each segment is optimally sized to capture relevant dynamics without imposing undue computational demands, thereby maintaining the integrity of the generation process while accommodating the practical constraints of data processing.

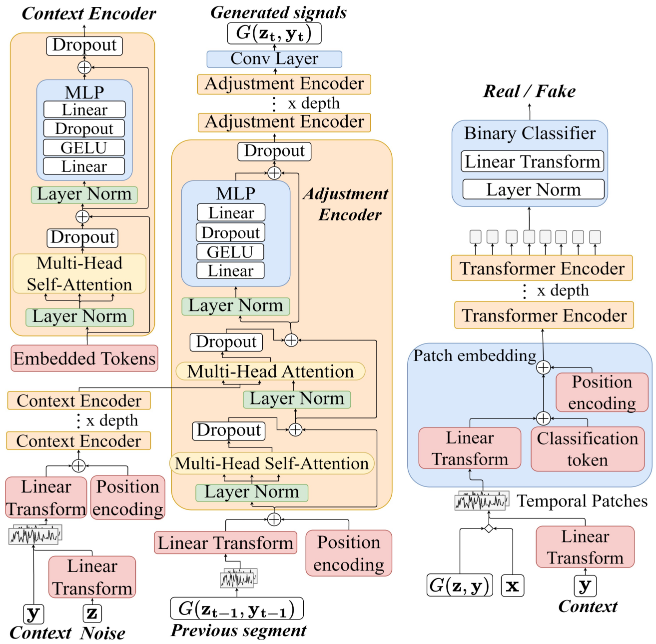
Phase-wise and Segment-wise Data Generation. Data generation is performed using phase-specific models for each phase and corresponding segments. These models are designed to generate multivariate time series that mimic the real engine's statistical characteristics and temporal behavior as a response to the mission profile events—more details about the architecture of the generative models are in the following section. The conditional aspect of the generation is a critical component of our methodology. Each window was not generated separately; instead, the generation of a given window considered the previously generated window. This approach ensured continuity and coherence in the generated data, preserving the temporal dependencies.

De-normalization of the Generated Data. Re-applying the original mean and standard deviation values to the generated data, effectively returning it to its original scale.

Concatenation of Sequentially Generated Segments. The final step involves concatenating all sequentially generated segments to form a complete multivariate time series where each variable represents a CEOD parameter.

3.2 MTS-CGAN Architecture

This model generates context-dependent multivariate time series data by incorporating a conditional layer into its gen-



(a) Generator (b) Discriminator

Figure 2: Architecture of the Generator and the Discriminator

eration mechanism. It employs a generator, denoted as G (refer to Figure 2a), alongside a discriminator, D (see Figure 2b) for the training. The enhanced MTS-CGAN framework introduces conditioning not only based on the given context but also leverages the segment generated immediately prior. Here, the context refers to the mission profile, represented as a multivariate time series. As detailed in Section 3.1, the generation process for any specific window considers the window produced beforehand. This approach guarantees a seamless transition between consecutive data segments, effectively maintaining the temporal correlations inherent in the original multivariate time series.

The conditional generator (G) consists of two distinct components: The first component encodes the context of the simulated flight. This requires two inputs: a noise vector of dimension d_z and the encoded context of dimension d_y. The noise vector is encoded into a latent dimension space d_y, then concatenated across the dimension known as 'channels.' This latent space's dimension is a data-dependent hyperparameter. We then apply linear transformations to the concatenated vectors to obtain a vector with a size equal to the target sequence length and d_c channels, where d_c must be tuned. Finally, we use a positional embedding vector to encode each element's position. It helps capture the sequence's order. Multiple consecutive blocks of the context encoder then process the final vector. The context encoder mirrors a conventional transformer encoder [Vaswani *et al.*, 2017] where the multi-head self-attention layer extracts contextual inter-dependencies between the generated signal and the provided context. The second component of the conditional generator refines the generation process to incorporate information from the previously generated window. It takes the previous window as input and generates positional-aware embeddings from it. These embeddings are then channeled into the Adjustment Encoder. This encoder consists of two main layers. First, a multi-head self-attention layer processes the embeddings to extract features from the previously generated window. Then, a separate multi-head attention layer is used, with the query derived from the output of the self-attention layer, while the key and value are derived from the output of the generator's first component. This combination allows the generation process to incorporate the context and ensure continuity in the generated sequence. We note that the Adjustment Encoder block is iteratively repeated N times, where N is a tunable hyperparameter that should be adjusted according to the specific requirements of the generation task. The final output is processed by a convolutional layer with a kernel size of (1,1), where the number of output channels equals the target dimension.

The conditional discriminator (D) is designed to distinguish between real and generated multivariate time series data. It processes either real or generated CEOD inputs, each accompanied by the mission profile. Initially, it concatenates the CEOD and the profile mission multivariate time series along the channel dimension, followed by a linear transformation to produce an embedding. This embedding is then segmented into multiple patches, each with its positional encoding. These segmented patches are then input into the consecutive layers of the Transformer's encoder. A binary classifier leverages the final embedding to assess the likelihood of the input being real or generated, assigning a score to indicate this distinction. The architecture of this mechanism is illustrated in Figure 2b.

We use the Least Squares GAN (LSGAN) loss [Mao *et al.*, 2017], replacing the standard GAN's [Goodfellow *et al.*, 2020] cross-entropy loss with a least squares loss for better training stability and convergence. An extra loss term is added to the generator to ensure seamless transitions in flight sequences by aligning overlapping segments of 20 data points each, preserving sequence continuity without data repetition. Both discriminator and generator parameters are optimized to minimize their respective loss functions, L_D and L_G.

$$
\begin{aligned}
L_D =& \frac{1}{2} E_{\mathbf{x},\mathbf{y}\sim p_{\text{data}}} \left[(D(x,y)-1)^2 \right] + \\
&+ \frac{1}{2} E_{\mathbf{z}\sim p_{\mathbf{z}}} \left[(D(G(z,y),y))^2 \right]
\end{aligned} \tag{1}
$$

$$
\begin{aligned}
L_G =& \frac{1}{2} E_{\mathbf{z}\sim p_{\mathbf{z}}} \left[(D(G(z_t,y_t),y_t)-1)^2 \right] + \\
&+ \left\| G_{1:20}\left(z_t, y_t\right) - G_{\text{end-19:end}}\left(z_{t-1}, y_{t-1}\right) \right\|_2
\end{aligned} \tag{2}
$$

4 Conclusion

In this demo, we introduced AESim, a novel aircraft engine simulator framework using a Transformer-based architecture to model multivariate time series in conditional generation tasks. The model learned the distribution of observed data for each context simultaneously. Self-attention mechanisms effectively captured the conditional generation aspect, maintaining the complex dynamics of the engine parameters. Although in this demonstration, we have focused on a single complex aircraft engine use case, it is interesting to note that the approach can be extended to other industrial use cases. The system is being tested internally at Safran's DataLab and there are plans to test it in other company departments.

References

[Coussirou *et al.*, 2022] Jean Coussirou, Thomas Vanaret, Jérôme Lacaille, and Safran Aircraft Engines DataLab. Anomaly detections on the oil system of a turbofan engine by a neural autoencoder. In *30th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, ESANN*, 2022.

[Errandonea *et al.*, 2020] Itxaro Errandonea, Sergio Beltrán, and Saioa Arrizabalaga. Digital twin for maintenance: A literature review. *Computers in Industry*, 123:103316, 2020.

[Forest *et al.*, 2018] Florent Forest, Jérôme Lacaille, Mustapha Lebbah, and Hanene Azzag. A generic and scalable pipeline for large-scale analytics of continuous aircraft engine data. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 1918–1924, 2018.

[Forest *et al.*, 2020] Florent Forest, Quentin Cochard, Cecile Noyer, Marc Joncour, Jérôme Lacaille, Mustapha Lebbah, and Hanene Azzag. Large-scale vibration monitoring of aircraft engines from operational data using self-organized models. In *annual conference of the PHM society*, volume 12, pages 11–11, 2020.

[Goodfellow *et al.*, 2020] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[Kim *et al.*, 2020] Sangjo Kim, Kuisoon Kim, and Changmin Son. Transient system simulation for an aircraft engine using a data-driven model. *Energy*, 196:117046, 2020.

[Lacaille and Langhendries, 2022] Jérôme Lacaille and Raphaël Langhendries. Corrosion risk estimation and cause analysis on turbofan engine. In *Annual Conference of the PHM Society*, volume 14, 2022.

[Madane and Lacaille, 2023] Abdellah Madane and Jerôme Lacaille. Simulation of the behaviour of engines in their current state of wear. In *Proceedings of the International Conference on Condition Monitoring and Asset Management*, volume 2023, pages 1–11. The British Institute of Non-Destructive Testing, 2023.

[Madane *et al.*, 2023] Abdellah Madane, Mohamed djallel Dilmi, Florent Forest, Hanane Azzag, Mustapha Lebbah, and Jerome Lacaille. Transformer-based conditional generative adversarial network for multivariate time series generation. In *International Workshop on Temporal Analytics@PAKDD 2023*, 2023.

[Mao *et al.*, 2017] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2794–2802, 2017.

[Mirza and Osindero, 2014] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[Thelen *et al.*, 2022] Adam Thelen, Xiaoge Zhang, Olga Fink, Yan Lu, Sayan Ghosh, Byeng D Youn, Michael D Todd, Sankaran Mahadevan, Chao Hu, and Zhen Hu. A comprehensive review of digital twin—part 1: modeling and twinning enabling technologies. *Structural and Multidisciplinary Optimization*, 65(12):354, 2022.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[Wang *et al.*, 2017] Zhiguang Wang, Weizhong Yan, and Tim Oates. Time series classification from scratch with deep neural networks: A strong baseline. In *2017 International joint conference on neural networks (IJCNN)*, pages 1578–1585. IEEE, 2017.