

Simulateur de moteurs d'avions

Abdellah Madane^{*,**}, Florent Forest^{****}, Hanane Azzag^{***}
Mustapha Lebbah^{*}, Jérôme Lacaille^{**}

*Laboratoire DAVID, Université Paris Saclay, 78035 Versailles, France
prenom.nom@uvsq.fr

**DataLab, Safran Aircraft Engines, 77550 Moissy-Cramayel, France
prenom.nom@safrangroup.com

****IMOS Lab, EPFL, Lausanne, Switzerland
florent.forest@epfl.ch

***LIPN, UMR CNRS 7030, Université Sorbonne Paris Nord, 93430 Villetaneuse, France
azzag@univ-paris13.fr

Résumé. Nous introduisons un simulateur de moteurs d'avion, développé à l'aide de réseaux antagonistes génératifs conditionnels basés sur l'architecture Transformer. Ce simulateur produit des mesures de capteurs pour des moteurs durant des vols complets, en fonction d'un profil de mission prédéfini qui reflète les conditions de vol. Intégré dans le cadre des jumeaux numériques de moteurs d'avion, cet outil est capable de simuler leurs performances pour différentes missions. Il offre la possibilité de comparer le comportement de divers moteurs dans les mêmes conditions opérationnelles et de simuler différents scénarios pour un moteur spécifique, facilitant ainsi des applications telles que l'analyse du comportement des moteurs, l'identification des limites de performance et l'optimisation des programmes de maintenance dans une stratégie globale de gestion de la santé et de la prévision. Il permet aussi la complétion de données de vol manquantes et répond aux enjeux de confidentialité en générant des jeux de données de vol synthétiques, potentiellement partageables pour des recherches publiques ou des défis de type hackathon.

1 Introduction

De nombreuses industries adoptent aujourd'hui progressivement les jumeaux numériques pour créer des représentations digitales de leurs systèmes physiques. Ces jumeaux numériques, des modèles interconnectés, sont des répliques précises du monde réel et jouent un rôle crucial dans l'amélioration de la prise de décision, la surveillance de l'état de santé, l'optimisation des processus et le renforcement du contrôle qualité (Thelen et al., 2022; Errandonea et al., 2020).

L'analyse et la simulation du comportement des moteurs d'avion captent particulièrement l'attention dans l'industrie aéronautique (Xiong et al., 2023; Kim et al., 2020) en raison de leurs répercussions sur la performance, la maintenance, la sécurité et la durabilité des appareils. Les capteurs intégrés dans les moteurs recueillent des Données Opérationnelles Continues du Moteur (CEOD) (Forest et al., 2018), des séries temporelles multivariées qui sont essentielles

pour les applications de Pronostic et Management de la Santé (PHM) (Coussirou et al., 2022; Lacaille et Langhendries, 2022; Forest et al., 2020). Toutefois, il arrive que ces données soient incomplètes ou biaisées en raison de défaillances des capteurs ou de retards dans le transfert de données.

Par ailleurs, les contraintes contractuelles limitent la collaboration entre les compagnies aérospatiales et les universités pour l'exploitation de ces données dans la recherche, car elles restent propriété des compagnies aériennes. La recherche académique nécessite un accès ouvert aux données pour permettre la validation des méthodes et assurer la reproductibilité des résultats.

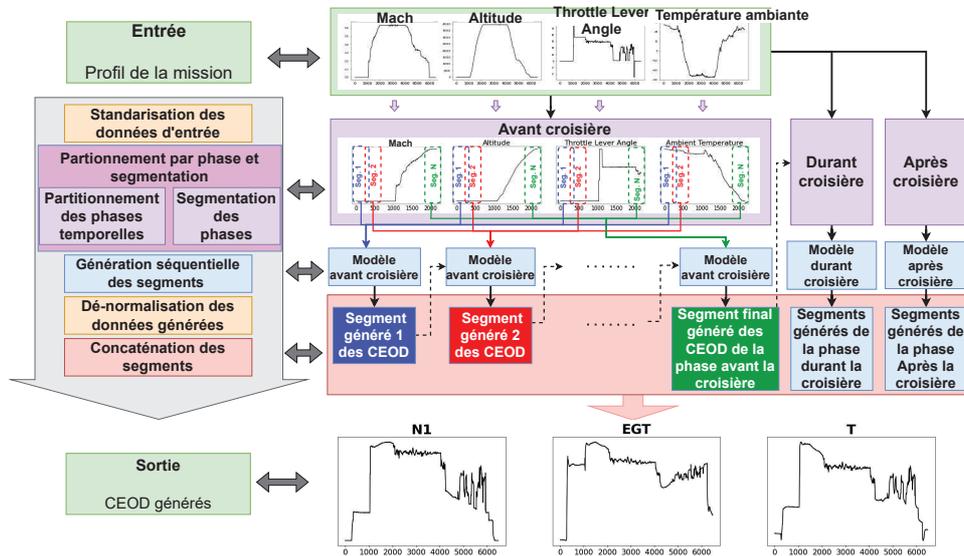
Chez Safran Aircraft Engines, nous avons prouvé que l'utilisation des CEOD recueillies après chaque vol permet de modéliser de manière précise le moteur grâce à des modèles génératifs conditionnels. Ce jumeau numérique, réglé selon les conditions de vol et les paramètres de contrôle, peut simuler le comportement du moteur comme s'il avait réalisé la mission de vol simulée, fournissant ainsi des CEOD réalistes pour des vols fictifs.

Nous présentons un simulateur de moteurs d'avions qui reproduit le comportement d'un moteur réel en simulant sa dynamique complexe. Il pallie les problèmes de données manquantes et les contraintes de confidentialité en générant des données de simulation réalistes pour des missions hypothétiques, permettant ainsi la comparaison de différents comportements de moteurs sous les mêmes conditions. Ce simulateur fournit des jeux de données pour la recherche, tant interne qu'académique, éliminant les problèmes de confidentialité et favorisant le développement de méthodes PHM robustes. Le simulateur utilise une architecture novatrice de réseaux antagonistes génératifs conditionnels pour séries temporelles multivariées (MTS-CGAN) (Madane et al., 2024, 2023; Madane et Lacaille, 2023; Mirza et Osindero, 2014), adaptée spécifiquement pour la simulation des moteurs d'avion.

Lorsqu'on parle de la simulation de performances des moteurs d'avion, les approches par modèles basés sur les données et par modèles physiques offrent des avantages distincts. Les modèles physiques, reposant sur les principes fondamentaux de la physique et de l'ingénierie, fournissent une représentation détaillée des comportements moteur sous diverses contraintes. Cependant, leur utilisation implique souvent de lourdes exigences computationnelles et dépend de la capacité à pouvoir modéliser toutes les équations physiques qui régissent les différents paramètres du moteur. En revanche, les modèles basés sur les données sont capables d'apprendre à partir de grands jeux de données la simulation des dynamiques complexes des moteurs, améliorant continuellement leur précision et leur réactivité par l'exposition à de nouvelles données, sans dépendre exclusivement des principes physiques traditionnels. Cela permet ainsi des calculs plus rapides et une meilleure adaptabilité aux nouveaux environnements opérationnels, même sans une compréhension physique approfondie. Bien que leur efficacité dépende de la qualité et de la disponibilité des données et leurs résultats soient plus difficiles à interpréter et expliquer, ces modèles gagnent en popularité, notamment pour les applications nécessitant des simulations rapides et en temps réel ainsi qu'un apprentissage adaptatif.

2 Étude de cas

Le simulateur est entraîné sur des données continues de moteurs d'avions appartenant à la même flotte. Des capteurs embarqués collectent plusieurs mesures à différentes fréquences. Nous les traitons à une fréquence uniforme de 1 Hz. Au cours de cette démonstration, nous

FIG. 1 – *Simulateur de données de moteurs d'avions*

présentons la génération de trois paramètres : la vitesse du rotor à basse pression (N1), la température à l'entrée de la chambre de combustion (T) et la température des gaz d'échappement (EGT), conditionnée sur cinq paramètres, comme illustré dans la Figure 1, représentant le profil de mission de vol simulé qui sont la température ambiante, l'altitude, la vitesse de l'avion (Mach), l'angle de la manette (TLA) et une variable booléenne (non représentée sur la figure) indiquant si le moteur est en marche ou non.

3 Fonctionnement du simulateur

3.1 Méthodologie

Nous présentons une vue globale des différents composants du simulateur dans la Figure 1, où un profil de mission est utilisé comme entrée et les paramètres CEOD générés représentent la sortie. Le fonctionnement implique plusieurs étapes :

Standardisation de l'entrée. Les données temporelles multivariées brutes sont normalisées en soustrayant la moyenne et en divisant par l'écart-type de chaque variable pour assurer une contribution égale à l'analyse.

Partitionnement des Phases Temporelles. Les données continues des moteurs peuvent être extrêmement longues, avoir des longueurs variables et avoir différentes phases, résultant en différents états opérationnels du moteur. Les séries temporelles multivariées sont divisées en phases distinctes reflétant les multiples dynamiques temporelles de chaque vol (avant, pendant,

après la Croisière). Ce partitionnement facilite l'isolation des périodes présentant des caractéristiques homogènes, permettant une modélisation plus ciblée dans les étapes ultérieures.

Segmentation à l'intérieur des Phases. Chaque phase identifiée est ensuite divisée en segments de durée fixe, chaque segment couvrant 300 secondes et avec un chevauchement de 20 intervalles de temps entre les segments consécutifs. Cette stratégie de segmentation facilite le traitement des séries temporelles de longueurs variables, y compris des vols extrêmement longs, en éliminant les restrictions sur la forme de l'entrée. En divisant les données en segments gérables, l'approche garantit que le simulateur peut accommoder des séquences de toute longueur sans pré-spécifier une dimension d'entrée fixe. Cette flexibilité est cruciale puisque les données de vol ont des échelles temporelles diverses, et leur longueur ne peut pas être prédéterminée. Le choix de la longueur des segments a été informé par des recommandations d'experts, reflétant une considération minutieuse de la réponse du moteur aux phénomènes physiques locaux, qui typiquement est, au plus, une durée de cinq minutes. De plus, cette décision était guidée par un équilibre entre la rétention d'informations suffisantes pour permettre une génération de données efficace et la faisabilité computationnelle, particulièrement en raison de la complexité quadratique des mécanismes d'auto-attention par rapport à la longueur de la séquence, rendant cela coûteux en terme de ressources pour de longues séquences. Cette approche assure que chaque segment est de taille optimale pour capturer les dynamiques pertinentes sans imposer de demandes computationnelles excessives, tout en maintenant l'intégrité du processus de génération et en accommodant les contraintes pratiques du traitement des données.

Génération de Données Phase par Phase et Segment par Segment. La génération des données est effectuée en utilisant des modèles spécifiques à chaque phase pour chaque segment correspondant. Ces modèles sont conçus pour générer des séries temporelles multivariées qui imitent les caractéristiques statistiques et le comportement temporel réels du moteur en réponse aux événements du profil de mission. Plus de détails sur l'architecture des modèles génératifs sont dans la section suivante. L'aspect conditionnel de la génération est un composant critique de notre méthodologie. Chaque fenêtre n'a pas été générée séparément; au lieu de cela, la génération d'une fenêtre donnée a pris en compte la fenêtre précédemment générée. Cette approche a assuré la continuité et la cohérence dans les données générées, préservant les dépendances temporelles.

Dé-normalisation des Données Générées. Ré-application des écarts-types et des valeurs moyennes originaux aux données générées, les ramenant effectivement à leur échelle originale.

Concaténation des Segments Générés Séquentiellement. L'étape finale implique la concaténation de tous les segments générés séquentiellement pour former une série temporelle multivariée complète où chaque variable représente un paramètre CEOD.

Ce modèle génère des données sérielles temporelles multivariées dépendantes du contexte en intégrant une couche conditionnelle dans son mécanisme de génération. Il utilise un générateur, noté G (voir Figure 2a), conjointement avec un discriminateur, D (voir Figure 2b), pour l'entraînement. Le modèle MTS-CGAN mis à jour introduit un conditionnement basé non seulement sur le contexte donné, mais aussi sur le segment généré avant. Ici, le contexte fait

référence au profil de mission, représenté comme une série temporelle multivariée. Comme détaillé dans la Section 3.1, le processus de génération pour une fenêtre spécifique prend en compte la fenêtre produite auparavant. Cette approche garantit une transition fluide entre les segments de données consécutifs, maintenant efficacement les corrélations temporelles inhérentes à la série temporelle multivariée originale.

3.2 Architecture MTS-CGAN

Le générateur conditionnel (G) contient deux composants distincts. Le premier composant encode le contexte du vol simulé. Cela nécessite deux entrées : un vecteur de bruit de dimension d_z et le contexte encodé de dimension d_y . Le vecteur de bruit est encodé dans un espace de dimension latente d_y , puis concaténé à travers la dimension connue sous le nom de 'canaux'. La dimension de cet espace latent est un hyperparamètre dépendant des données. Nous appliquons ensuite des transformations linéaires aux vecteurs concaténés pour obtenir un vecteur de taille égale à la longueur de la séquence cible et à d_c canaux, où d_c doit être ajusté. Enfin, nous utilisons un vecteur d'encodage positionnel pour coder la position de chaque élément. Cela aide à capturer l'ordre de la séquence. Plusieurs blocs consécutifs de l'encodeur de contexte traitent ensuite le vecteur final. L'encodeur de contexte reflète un encodeur de Transformer conventionnel (Vaswani et al., 2017) où la couche d'auto-attention multi-têtes extrait les interdépendances contextuelles entre le signal généré et le contexte fourni. Le second composant du générateur conditionnel affine le processus de génération pour intégrer des informations de la fenêtre précédemment générée. Il prend la fenêtre précédente comme entrée et produit des vecteurs qui encode la position de chaque élément. Ces vecteurs sont ensuite canalisés dans l'Encodeur d'Ajustement. Cet encodeur se compose de deux couches principales. D'abord, une couche d'auto-attention multi-têtes traite les vecteurs encodés pour extraire les caractéristiques de la fenêtre générée précédemment. Ensuite, une couche d'attention multi-têtes séparée est utilisée, la requête étant dérivée de la sortie de la couche d'auto-attention, tandis que la clé et la valeur proviennent de la sortie du premier composant du générateur. Cette combinaison permet au processus de génération d'incorporer le contexte et d'assurer la continuité dans la séquence générée. Nous notons que le bloc Encodeur d'Ajustement est itérativement répété N fois, où N est un hyperparamètre réglable qui devrait être ajusté selon les besoins spécifiques de la tâche de génération. La sortie finale est traitée par une couche convolutionnelle de taille de noyau (1,1), où le nombre de canaux de sortie équivaut à la dimension cible.

Le discriminateur conditionnel (D) est conçu pour distinguer entre les données réelles et générées. Il traite soit des entrées CEOD réelles, soit générées, chacune accompagnée du profil de mission. Initialement, il concatène les CEOD et le profil de mission en série temporelle multivariée le long de la dimension des canaux, suivi d'une transformation linéaire pour produire un embedding. Cet embedding est ensuite segmenté en plusieurs patches, chacun avec son codage positionnel. Ces patches segmentés sont ensuite entrés dans les couches consécutives de l'encodeur du transformateur. Un classifieur binaire utilise l'embedding final pour évaluer la probabilité que l'entrée soit réelle ou générée, attribuant un score pour indiquer cette distinction. L'architecture de ce mécanisme est illustrée dans la Figure 2b.

Nous utilisons la fonction de perte LSGAN (Least Squares GAN) (Mao et al., 2017). Un terme supplémentaire est ajouté à la fonction de perte du générateur pour assurer des transi-

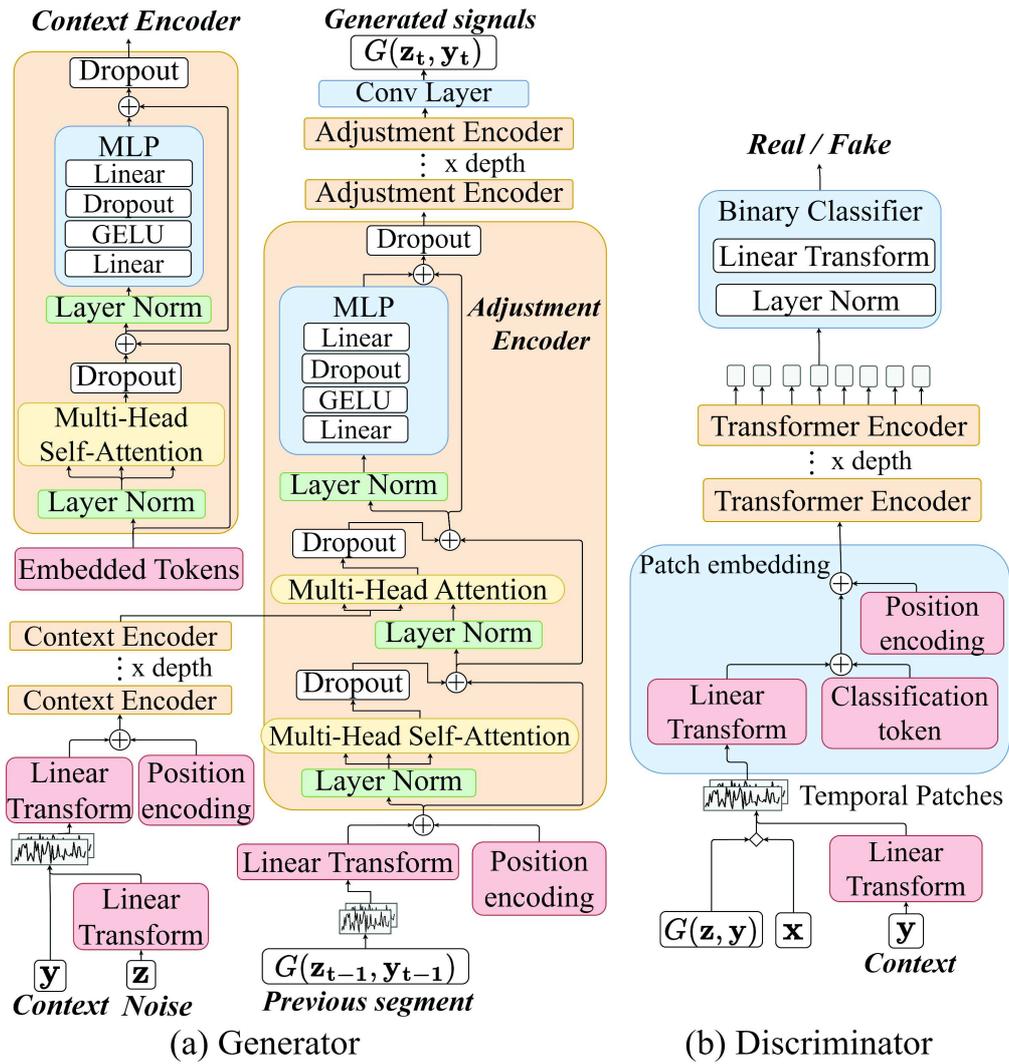


FIG. 2 – Architecture du Générateur et du Discriminateur

tions fluides dans les séquences de vol en alignant les segments chevauchants de 20 points de données chacun, préservant la continuité de la séquence sans répétition de données. Les paramètres du discriminateur et du générateur sont optimisés pour minimiser leurs fonctions de perte respectives, L_D et L_G .

4 Conclusion

Dans cette démonstration, nous avons introduit un nouveau simulateur de moteur d'avion utilisant une architecture basée sur les Transformers pour modéliser des séries temporelles multivariées pour des tâches de génération conditionnelle. Le modèle a appris la distribution des données observées pour chaque contexte simultanément. Les mécanismes d'auto-attention ont efficacement capturé l'aspect de génération conditionnelle, maintenant les dynamiques complexes des paramètres des moteurs. Bien que dans cette démonstration, nous nous soyons concentrés sur un cas d'utilisation complexe de moteur d'avion, il est intéressant de noter que l'approche peut être étendue à d'autres cas d'utilisation industriels. Le système est actuellement testé en interne au DataLab de Safran et des plans sont en place pour le tester dans d'autres départements de l'entreprise.

Références

- Coussirou, J., T. Vanaret, J. Lacaille, et S. A. E. DataLab (2022). 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*.
- Errandonea, I., S. Beltrán, et S. Arrizabalaga (2020). Digital twin for maintenance : A literature review. *Computers in Industry* 123, 103316.
- Forest, F., Q. Cochard, C. Noyer, M. Joncour, J. Lacaille, M. Lebbah, et H. Azzag (2020). Large-scale vibration monitoring of aircraft engines from operational data using self-organized models. In *annual conference of the PHM society*, Volume 12, pp. 11–11.
- Forest, F., J. Lacaille, M. Lebbah, et H. Azzag (2018). A generic and scalable pipeline for large-scale analytics of continuous aircraft engine data. In *2018 IEEE International Conference on Big Data (Big Data)*, pp. 1918–1924.
- Kim, S., K. Kim, et C. Son (2020). Transient system simulation for an aircraft engine using a data-driven model. *Energy* 196, 117046.
- Lacaille, J. et R. Langhendries (2022). Corrosion risk estimation and cause analysis on turbofan engine. In *Annual Conference of the PHM Society*, Volume 14.
- Madane, A., M. djallel Dilmi, F. Forest, H. Azzag, M. Lebbah, et J. Lacaille (2023). Transformer-based conditional generative adversarial network for multivariate time series generation. In *International Workshop on Temporal Analytics@PAKDD 2023*, doi: https://pakdd23.pakdd.org/wp-content/uploads/2023/05/pakdd23_w1_p2.pdf.
- Madane, A., F. Forest, H. Azzag, M. Lebbah, et J. Lacaille (2024). One-pass generation of multivariate time series through conditional multivariate modeling. In *2024 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–9. IEEE.

- Madane, A. et J. Lacaille (2023). 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, pp. 1–11. The British Institute of Non-Destructive Testing, doi: 10.1784/cm2023.5d2.
- Mao, X., Q. Li, H. Xie, R. Y. Lau, Z. Wang, et S. Paul Smolley (2017). Least squares generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pp. 2794–2802.
- Mirza, M. et S. Osindero (2014). Conditional generative adversarial nets. *arXiv preprint arXiv :1411.1784*.
- Thelen, A., X. Zhang, O. Fink, Y. Lu, S. Ghosh, B. D. Youn, M. D. Todd, S. Mahadevan, C. Hu, et Z. Hu (2022). A comprehensive review of digital twin—part 1 : modeling and twinning enabling technologies. *Structural and Multidisciplinary Optimization* 65(12), 354.
- Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, et I. Polosukhin (2017). Attention is all you need. *Advances in neural information processing systems* 30.
- Xiong, J., O. Fink, J. Zhou, et Y. Ma (2023). Controlled physics-informed data generation for deep learning-based remaining useful life prediction under unseen operation conditions. *Mechanical Systems and Signal Processing* 197, 110359.

Summary

We introduce an aircraft engine simulator developed with Transformer-based conditional generative adversarial networks. It simulates engine performance across complete flights based on specific mission profiles, aiding in performance analysis and maintenance optimization within digital twin frameworks.