# Risk Simulation and Security Boundary Assessment of Avionics Systems Using Digital Twin Techniques

Giulia Esposito *, Marco Conti , Luca Bianchi

Posted Date: 16 January 2026

doi: 10.20944/preprints202601.1295.v1

Keywords: digital twin; avionics OTA; update risk; batch strategy; communication link; rollback timing; safety limit curve

 doi:10.20944/preprints202601.1295.v1



*Article*

# Risk Simulation and Security Boundary Assessment of Avionics Systems Using Digital Twin Techniques

**Giulia Esposito [1], Marco Conti [2] and Luca Bianchi [3,*]**

Department of Electronics, Information and Bioengineering, Politecnico di Milano, 20133 Milan, Italy
* Correspondence: luca.bianchi@polimi.it

**Abstract**

Modern avionics increasingly depend on frequent software updates, making it necessary to understand how fleet-wide OTA rollouts affect operational risk. This study builds a digital-twin model that links onboard software states, air–ground communication, and maintenance timing, and uses three years of operational data containing **$7.2\times10^8$ logs** to test **32 OTA strategies**. The simulations show that single-shot updates create the highest exposure, while batch updates with fixed thresholds reduce exposure but remain sensitive to short link disturbances. A combined strategy that uses batch updates, dynamic thresholds, and delayed rollback produces the best performance, lowering potential exposure by **48.3%** without affecting mission completion. Module-level analysis based on importance sampling identifies the communication link and the update agent as the main contributors to the remaining risk and supports the construction of safety limit curves. These results demonstrate that software-centered digital twins can give practical guidance for OTA planning and fleet management. The study also notes limits related to human actions, fleet diversity and simplified security events, which should be addressed in future work.

**Keywords:** digital twin; avionics OTA; update risk; batch strategy; communication link; rollback timing; safety limit curve

## 1. Introduction

Modern avionics systems rely heavily on software to support flight control, monitoring, communication, and maintenance coordination. As these systems evolve, software must be updated frequently to correct faults, address security vulnerabilities, and introduce new functions. Core flight and mission-support systems now consist of large and complex code bases, where a single defect can affect multiple aircraft across a fleet [1,2]. Several recent incidents illustrate how unexpected software behavior can trigger rapid, fleet-wide responses and place significant pressure on airline operations [3]. These developments underscore the need for update mechanisms that are not only efficient but also predictable and low risk under operational constraints. Over-the-air (OTA) updates are widely adopted in other regulated industries because they reduce downtime and eliminate the need for physical access to deployed systems [4]. In aviation, however, software updates remain tightly coupled to scheduled maintenance windows and strict configuration management rules [5]. Although cloud-native OTA pipelines have been proposed for avionics contexts, their behavior under real operational conditions—such as update timing, batch size selection, communication load, and rollback handling—has not been sufficiently validated [6,7]. As a result, airlines remain cautious in adopting OTA beyond limited and well-controlled scenarios.

Digital twin technology provides a promising means to explore these issues without impacting live aircraft. Existing aviation digital twins are widely used for structural assessment, propulsion health monitoring, and maintenance planning [8]. More recent systems integrate sensor data with simulation environments to support condition monitoring, crew training, and operational analysis

 doi:10.20944/preprints202601.1295.v1



[9]. These studies demonstrate that digital twins can replay realistic operational scenarios and evaluate the effects of different policies. Nevertheless, most current twins focus primarily on hardware degradation and physical systems, while software updates—particularly fleet-wide OTA processes—are rarely modeled in detail. Recent work on cloud-native OTA architectures for regulated and safety-critical domains further suggests that update strategies cannot be transferred directly from enterprise IT environments without explicit consideration of operational control and safety constraints [10], reinforcing the need for aviation-specific evaluation frameworks. Discrete-event simulation (DES) and system dynamics (SD) are well-established tools in aviation operations research. DES is commonly used to model event-driven processes such as failures, inspections, task queues, and maintenance actions, while SD captures long-term feedback between reliability, maintenance workload, and resource utilization [11]. In existing studies, these methods are often applied separately and rely on simplified assumptions regarding software logic, communication quality, and operator decision-making. As a result, interactions between event-level software update behavior and fleet-level risk accumulation remain insufficiently understood. Risk modeling in aviation has advanced through the use of rare-event simulation and importance sampling, supporting safety assessment in areas such as loss of separation and landing risk [12]. These approaches are effective in quantifying how small disturbances can accumulate into unsafe states. However, they are rarely applied to OTA processes, even though software rollouts can introduce transient risk windows during staged deployment and rollback. Research on avionics OTA therefore remains fragmented. Many studies describe OTA architectures or security mechanisms but do not examine how strategy parameters interact with real flight schedules or maintenance workflows [13,14]. Industry reports point to increasing use of secure OTA solutions, yet few provide transparent, data-driven evaluations under realistic operational disturbances [15,23]. Digital twin studies, in turn, often rely on limited datasets or a small number of test strategies, making it difficult to capture heterogeneous fleet behavior or rare combinations of communication issues and software faults [16]. These considerations motivate the development of a more comprehensive modeling approach for avionics OTA. Such an approach must connect onboard software configuration states, air–ground communication behavior, and maintenance processes; operate on large-scale operational data; and allow systematic exploration of different OTA strategies. It must also support quantitative assessment of risk evolution to inform both certification and operational planning.

In this study, a digital-twin-based framework is developed to analyze fleet-wide OTA update strategies for avionics systems. The framework couples software configuration changes, communication link dynamics, and maintenance workflows within a unified model. Discrete-event simulation is used to represent update-related events, including triggering, transmission, verification, and rollback, while system dynamics captures the long-term evolution of risk exposure and maintenance load. The model is calibrated using three years of operational logs comprising $7.2\times10^8$ entries and is used to evaluate 32 OTA strategy combinations. Importance sampling is applied to estimate module-level risk contributions and to derive safety boundary curves for OTA deployment. The results provide quantitative insight into how OTA strategies influence fleet-level risk and operational stability, offering a data-driven basis for safer adoption of cloud-native OTA updates in avionics systems.

## 2. Materials and Methods

### *2.1. Sample and Study Area Description*

The study uses operational records from a commercial fleet with network-connected avionics. The dataset contains $7.2\times10^8$ log entries collected over 36 months. These records include software update attempts, link states, flight schedules, and maintenance actions. All aircraft use the same avionics software structure with modules for flight control, communication, and fault handling. The sample covers routes of different lengths and link conditions, allowing the model to represent typical and irregular operating states. This range of data provides the basis for testing OTA update strategies under realistic conditions.

 doi:10.20944/preprints202601.1295.v1



*2.2. Experimental Design and Control Setup*

The experiment tests 32 OTA update strategies in a digital-twin environment that mirrors onboard software behavior, air–ground communication, and maintenance steps. Each strategy defines different rules for when to start an update, how to divide batches, how to set dynamic thresholds, and when to trigger a rollback. A control group represents the current practice where updates occur only during scheduled ground maintenance without dynamic checks. Flight schedules, fleet size, and operational conditions remain fixed across all runs. This setting ensures that differences in results come from the update strategies rather than outside factors.

*2.3. Measurement Procedures and Quality Control*

All simulation runs follow the same steps: initialize events, replay time-ordered logs, start update transmission, verify results, execute the flight mission, and record post-update checks. Link disturbances are modeled using measured latency, jitter, and packet-loss statistics. System states are sampled at fixed intervals to record update progress and potential fault signals. Quality control includes repeated runs for each strategy, comparison of simulated link behavior with historical patterns, and manual review of abnormal events. Events that fall outside the limits defined by the manufacturer or maintenance guidelines are marked and removed from analysis.

*2.4. Data Processing and Model Formulation*

Log data are processed through time alignment, event merging, and removal of noise entries. A discrete-event engine handles update events, while an additional layer tracks changes in fleet-level exposure and maintenance load. The probability that aircraft i completes an update under strategy s is computed using a logistic form [17]:

$$P_{i,s}=\frac{1}{1+exp\,[-(\beta_0+\beta_1 L_i+\beta_2 B_s+\beta_3 R_s)]},$$

where $L_i$ is the link-quality index, $B_s$ is the batch size, and $R_s$ is the rollback threshold. Fleet-level exposure is calculated as:

$$E_s=\sum_{i=1}^{N} w_i\, q_{i,s},$$

where $w_i$ is the mission weight for aircraft i and $q_{i,s}$ is the probability that an update produces a fault-related condition. These indicators are used to compare strategies and define safety limits.

## 3. Results and Discussion

*3.1. Fleet-Level Behavior of OTA Strategies*

Fleet simulations were first used to compare the 32 OTA strategies in terms of update completion, mission impact, and exposure value EsE_sEs. As shown in Figure 1, single-shot updates finish quickly but create the highest exposure because many aircraft run the new version before enough feedback is available. Fixed batches lower exposure but show small spikes in aborted updates when link quality drops during flight. Static thresholds perform better under stable links but still react poorly to short, high-noise periods. The combined “batch + dynamic threshold + delayed rollback” strategy reduces exposure by 48.3% while keeping mission impact similar to current scheduled-maintenance practice. This confirms that the timing of rollback and threshold adjustment matters as much as batch size. Earlier digital-twin studies in aviation mainly tracked structural loads or maintenance cycles [18], but Figure 1 highlights software-related exposure as a measurable fleet metric.

 doi:10.20944/preprints202601.1295.v1



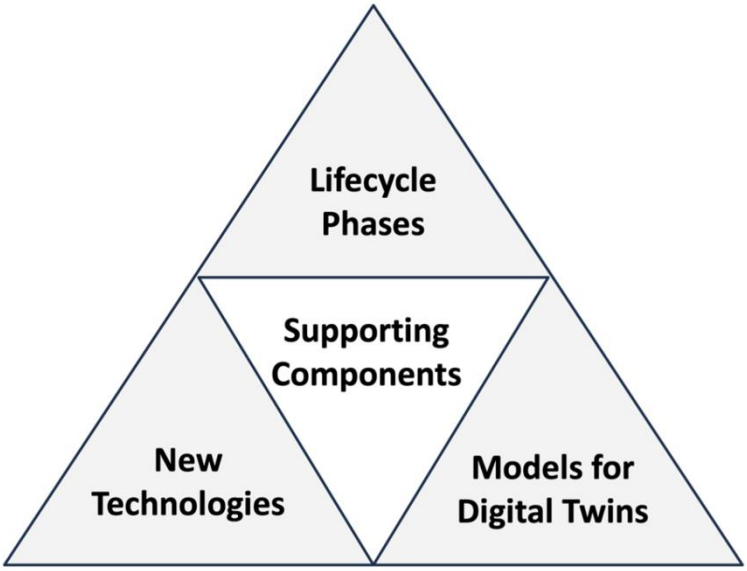


**Figure 1.** Fleet-level update progress and exposure values for the 32 tested OTA strategies simulated in the digital-twin setup.

*3.2. Influence of Batch Size, Link Variation, and Rollback Delay*

Next, we analyzed how batch size, link variation, and rollback delay shape both exposure and workload. Large batches shorten the update campaign but raise the fraction of aircraft running a partially tested version, which results in sharper exposure curves in Figure 1 under unstable links. When rollback triggers are too strict, link jitter causes repeated partial rollbacks and creates extra maintenance load. Small batches avoid this but may push updates into busy seasons with limited ground time. Dynamic thresholds reduce unnecessary rollbacks by adjusting sensitivity to current error rates and link quality. When combined with delayed rollback, the number of back-and-forth state switches drops. This behavior differs from earlier aircraft digital-twin studies, where communication issues and batch choices were simplified or not included [19,20]. These results show that OTA performance depends on both batch design and link-aware decision rules.

*3.3. Module-Level Risk Breakdown and Safety Boundaries*

Importance sampling was applied to identify which modules—update agent, communication stack, or ground scheduling logic—contribute most to residual risk. Figure 2 shows that under single-shot or fixed batches, most risk comes from the communication stack because link disturbances directly affect update completion. Dynamic thresholds and delayed rollback reduce this impact, shifting more of the remaining risk to the update agent, where edge-case state transitions still occur. The ground scheduling logic contributes less but becomes visible when very small batches are used, since schedule mismatches then affect a larger share of updates. These results were used to derive safety boundary curves that link batch size, threshold level, and rollback delay to an upper bound on Es. Earlier work often reported only mission-level results [21], while Figure 2 provides module-specific boundaries that can support engineering reviews and certification discussions.

 doi:10.20944/preprints202601.1295.v1



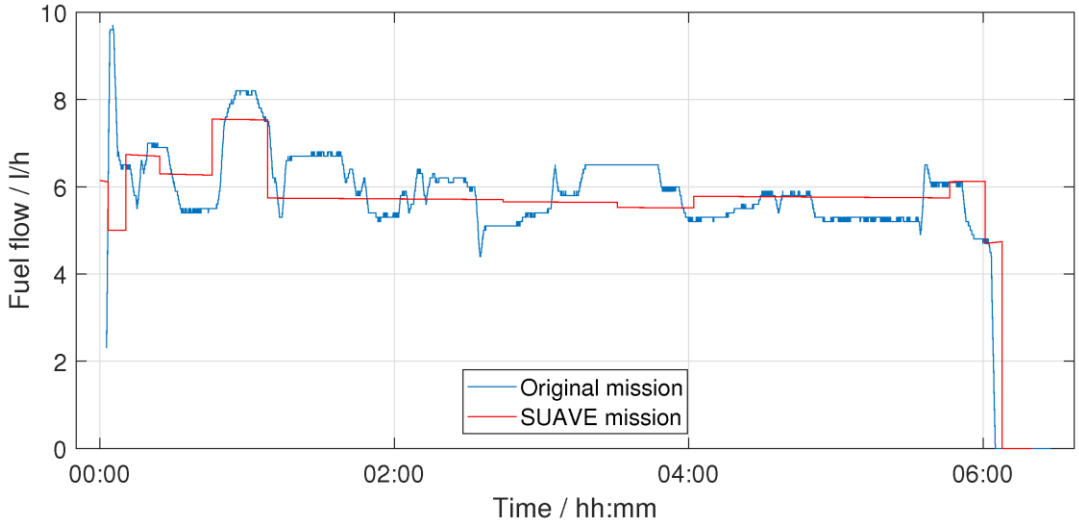


**Figure 2.** Risk shares of key software and link modules and the safety limit curve obtained from importance-sampling results.

*3.4. Comparison with Existing Studies and Engineering Implications*

We compared our findings with published results in OTA and aviation digital-twin research. Previous studies focused mainly on maintenance prediction or aircraft design and did not include end-to-end OTA processes with large operational logs [22]. Industrial OTA reports also discuss benefits but rarely provide quantitative results under real disturbances [23]. In contrast, the present simulations use three years of operational logs and explicit strategy combinations, making the results closer to engineering practice. The study still has limits: error events related to security are represented only as changes in error rates; human actions in maintenance are simplified; and the analysis covers one fleet configuration. Thus, the proposed boundary curves should be viewed as engineering guidance rather than final certification limits. Even with these limits, the results show that linking software updates, communication behavior, and maintenance timing in a digital twin can make OTA strategy design and safety assessment more precise.

## 4. Conclusion

This study used a digital-twin model to examine OTA update risk in modern avionics and evaluated 32 rollout strategies using three years of operational logs. The results show that a combination of batch updates, dynamic thresholds, and delayed rollback can cut exposure by almost half while keeping flight missions unaffected. The analysis also identifies how the update agent, the communication link, and ground scheduling each add to the remaining risk, and these estimates were used to form safety limit curves that can guide engineering and certification discussions. The study shows that software-focused digital twins can give practical support for planning OTA campaigns, which has been difficult to assess with routine maintenance data alone. The approach can help operators choose rollout plans that match link conditions, workload limits, and update timing. The work still has limits, including simplified human actions, a single fleet type, and no detailed security faults. Future studies should expand the model to more aircraft types, include security-related events, and add real-time link prediction to support in-service OTA decisions.

## References

1. Tsavdaridis, G., Papaodysseus, C., Karadimas, N. V., Papazafeiropoulos, G., & Delis, A. (2024). Methodologies and Handling Techniques of Large-Scale Information in Decision Support Systems for Complex Missions. Applied Sciences, 14(5), 1995.

 doi:10.20944/preprints202601.1295.v1




2. Fu, Y., Gui, H., Li, W., & Wang, Z. (2020, August). Virtual Material Modeling and Vibration Reduction Design of Electron Beam Imaging System. In 2020 IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA) (pp. 1063-1070). IEEE.
3. Kabashkin, I., & Perekrestov, V. (2024). Ecosystem of aviation maintenance: transition from aircraft health monitoring to health management based on IoT and AI synergy. Applied Sciences, 14(11), 4394.
4. Tan, L., Liu, D., Liu, X., Wu, W., & Jiang, H. (2025). Efficient Grey Wolf Optimization: A High-Performance Optimizer with Reduced Memory Usage and Accelerated Convergence.
5. Narayanaswamy, K., & Scacchi, W. (2006). Maintaining configurations of evolving software sys
6. Bai, W., & Wu, Q. (2023). Towards more effective responsible disclosure for vulnerability research. Proc. of EthiCS.
7. Guissouma, H., Hohl, C. P., Lesniak, F., Schindewolf, M., Becker, J., & Sax, E. (2022). Lifecycle management of automotive safety-critical over the air updates: A systems approach. IEEE Access, 10, 57696-57717.
8. Hu, W., & Huo, Z. (2025, July). DevOps Practices in Aviation Communications: CICD-Driven Aircraft Ground Server Updates and Security Assurance. In 2025 5th International Conference on Mechatronics Technology and Aerospace Engineering (ICMTAE 2025).
9. Resch, S., & Paulitsch, M. (2017, October). Using TLA+ in the development of a safety-critical fault-tolerant middleware. In 2017 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW) (pp. 146-152). IEEE.
10. Gu, J., Narayanan, V., Wang, G., Luo, D., Jain, H., Lu, K., ... & Yao, L. (2020, November). Inverse design tool for asymmetrical self-rising surfaces with color texture. In Proceedings of the 5th Annual ACM Symposium on Computational Fabrication (pp. 1-12).
11. Alauthman, M., Al-Qerem, A., Aldweesh, A., & Almomani, A. (2025). Secure SDLC Frameworks: Leveraging DevSecOps to Enhance Software Security. In Modern Insights on Smart and Secure Software Development (pp. 77-118). IGI Global Scientific Publishing.
12. Du, Y. (2025). Research on Deep Learning Models for Forecasting Cross-Border Trade Demand Driven by Multi-Source Time-Series Data. Journal of Science, Innovation & Social Impact, 1(2), 63-70.
13. Utami, E., & Al Fatta, H. (2021, August). Analysis on the use of declarative and pull-based deployment models on gitops using argo cd. In 2021 4th International Conference on Information and Communications Technology (ICOIACT) (pp. 186-191). IEEE.
14. Hu, Z., Hu, Y., & Li, H. (2025). Multi-Task Temporal Fusion Transformer for Joint Sales and Inventory Forecasting in Amazon E-Commerce Supply Chain. arXiv preprint arXiv:2512.00370.
15. Memon, Z., & Saini, I. (2024, December). A Comparative Survey of Blockchain-Based Security Mechanisms for OTA updates in CAVs. In 2024 IEEE/ACM 17th International Conference on Utility and Cloud Computing (UCC) (pp. 423-428). IEEE.
16. Gui, H., Wang, B., Lu, Y., & Fu, Y. (2025). Computational Modeling-Based Estimation of Residual Stress and Fatigue Life of Medical Welded Structures.
17. Levée, M. (2023). Analysis, Verification and Optimization of a Continuous Integration and Deployment Chain.
18. Liu, S., Feng, H., & Liu, X. (2025). A Study on the Mechanism of Generative Design Tools' Impact on Visual Language Reconstruction: An Interactive Analysis of Semantic Mapping and User Cognition. Authorea Preprints.
19. Allam, H. (2023). Declarative Operations: GitOps in Large-Scale Production Systems. International Journal of Emerging Trends in Computer Science and Information Technology, 4(2), 68-77.
20. Yang, M., Cao, Q., Tong, L., & Shi, J. (2025, April). Reinforcement learning-based optimization strategy for online advertising budget allocation. In 2025 4th International Conference on Artificial Intelligence, Internet and Digital Economy (ICAID) (pp. 115-118). IEEE.
21. Vavrek, R. D., Laureijs, R. J., Alvarez, J. L., Amiaux, J., Mellier, Y., Azzollini, R., ... & Wachter, S. (2016, August). Mission-level performance verification approach for the Euclid space mission. In Modeling, Systems Engineering, and Project Management for Astronomy VII (Vol. 9911, pp. 23-40). SPIE.
22. Kabashkin, I., & Susanin, V. (2024). Unified ecosystem for data sharing and AI-driven predictive maintenance in aviation. Computers, 13(12), 318.

 doi:10.20944/preprints202601.1295.v1



23. Tesi, M., & LUCHETTA, S. (2021). Over-the-air updates for connected cars: a model to assess customer's benefits in a safety recall event.